UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
COORDENAÇÃO DE CIÊNCIA DA COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

EURISTENEDE VANUEL FRANCISCO DAS NEVES SANTOS

FRAMEWORK DE MINERAÇÃO DE DADOS AGROPECUÁRIOS

TRABALHO DE CONCLUSÃO DE CURSO

SANTA HELENA
2021

EURISTENEDE VANUEL FRANCISCO DAS NEVES SANTOS

FRAMEWORK DE MINERAÇÃO DE DADOS AGROPECUÁRIOS

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Ciência da Computação da Universidade Tecnológica Federal do Paraná, Campus Santa Helena, como requisito parcial à obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Me. Anderson Brilhador.
Coorientadora: Profa. Dra. Alessandra Matte.

SANTA HELENA

2021

EURISTENEDE VANUEL FRANCISCO DAS NEVES SANTOS

FRAMEWORK DE MINERAÇÃO DE DADOS AGROPECUÁRIOS

Trabalho de Conclusão de Curso de Graduação em Ciência da Computação, apresentado como requisito para obtenção do título de Bacharel em Ciência da Computação da Universidade Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 23 de agosto de 2021

Anderson Brilhador – Orientador
Mestre em Informática
Universidade Tecnológica Federal do Paraná

Alessandra Matte – Co-orientadora
Doutora em Desenvolvimento Rural
Universidade Tecnológica Federal do Paraná

Arlete Teresinha Beuren
Doutora em Informática
Universidade Tecnológica Federal do Paraná

Lilian Yukari Yamamoto
Doutora em Agronomia
Universidade Tecnológica Federal do Paraná

SANTA HELENA

2021

Dedico este trabalho aos meus familiares que sempre me apoiaram, a mim mesmo por ter superado as barreiras durante todo o processo de aprendizado, a minha noiva que sempre me motivou, aos meus professores que fizeram tudo isso acontecer e a todos os meus amigos que me ajudaram a vencer cada obstáculo.

AGRADECIMENTOS

Certamente estes parágrafos não irá atender a todas as pessoas que fizeram parte dessa conquista na minha vida. Portanto, desde já peço desculpas àquelas que não estão presentes entre essas palavras, mas podem ter a certeza que sou grato a cada um de vocês, que contribuíram direto ou indiretamente com esses resultados.

Quero agradecer primeiramente a Deus por ter proporcionado saúde, sabedoria e persistência para vencer todas as dificuldades.

A minha família e noiva que sempre me apoiaram e deram força mesmo nos momentos difíceis.

A Universidade Tecnológica Federal do Paraná Campus Santa Helena e todos os seus colaboradores que mantem as engrenagens funcionando.

Ao professor orientador Anderson Brilhador e a professora coorientadora Alessandra Matte, que apresentaram a ideia e me orientaram em todo o desenvolvimento do trabalho.

A todos os professores que ensinou e avaliou cada disciplina de acordo com seus critérios, alguns levarei saudades e admiração pelo profissionalismo e amizade, outros somente pela amizade e pelo o aprendizado de tipo de profissional que não quero ser.

Aos meus colegas e amigos de sala de aula que levarei para sempre em meu coração, talvez não nus reencontramos mais pessoalmente, mas sou grato por cada elogio e apoio.

A família do Vantuir, do Eudes Germano, família Vidal e do Mauro Sergio que foram os que sempre me receberam e apoiaram em todo o tempo que residi em Santa Helena.

E por todos que não foi citado nesse texto, mas que contribuíram com essa realização na minha vida acadêmica, profissional e pessoal.

Que diremos, pois, diante dessas coisas? Se Deus é por nós quem será contra nós? (Romanos 8:31).

RESUMO

Este trabalho apresenta o desenvolvimento de um *framework* de mineração de dados agropecuários. Utilizando o processo de extração de conhecimento em base de dados, denominado *Knowledge Discovery in Databases* KDD e seguindo as suas cinco etapas: seleção dos dados, pré-processamento, transformação, mineração, validação e visualização de resultados. A codificação do *framework* foi por meio da linguagem de programação Python, fazendo o uso de bibliotecas de aprendizagem de máquina do *scikit learn* e pandas, de bibliotecas para plotagem de gráficos do matplotlib e o desenho das interfaces por meio do IDE Qt Designer. Foi utilizado os algoritmos K-médias e DBScan para realizar agrupamentos de dados, e os algoritmos de árvores de decisão e *Naive bayes* para classificação. Tabelas do último censo agropecuário realizado em 2017, disponível no Sistema IBGE de Recuperação Automática – SIDRA, foram utilizadas para realizar um estudo de caso apresentado na seção 4 deste trabalho, com objetivo de testar a viabilidade do *framework*. Por meio do estudo de caso o uso do framework apresentou ser viável para utilização por profissionais ou pesquisadores da área. O algoritmo de agrupamento K-médias apresentou resultados igual a 0.78, 0.74 e 0.64 para um K igual a 2, 3 e 4 respectivamente, já o DBScan não apresentou resultados satisfatórios. O desempenho do classificador baseado em árvore de decisão conseguiu alcançar a marca de 100% sobre as métricas de precisão, *recall* e *f1-score*, enquanto, o classificador *Naive bayes* alcançou a marca de 80, 96 e 85% para precisão, *recall* e *f1-score*, respectivamente.

Palavras-chave: mineração de dados; agrupamento de dados; *framework*; classificação.

ABSTRACT

This work presents the development of a framework for data mining of agricultural data. Using the process of knowledge extraction in databases, called Knowledge Discovery in KDD Databases and following its five steps: data selection, pre-processing, transformation, mining, validation, and visualization of results. The framework was developed using the Python programming language, using machine learning libraries from scikit-learn and pandas, graph plotting libraries from matplotlib and interface design through the Qt Designer IDE. The K-means and DBScan algorithms were used to perform data grouping, and the decision tree and Naive Bayes algorithms for classification. Tables from the last agricultural census conducted in 2017, available in the IBGE System for Automatic Recovery - SIDRA, were used to perform a case study presented in section 4 of this work to test the feasibility of the framework. Through the case study, the use of the framework proved to be viable for use by professionals or researchers in the area. The K-means clustering algorithm is equal to 0.78, 0.74, and 0.64 for a K equal to 2, 3, and 4, respectively. The performance of the decision tree-based classifier achieved 100% on the metrics of precision, recall, and f1-score, while the Naive Bayes classifier achieved 80, 96, and 85% for precision, recall, and f1-score, respectively.

Keywords: data mining; data clustering; framework; classification.

LISTA DE FIGURAS

LISTA DE TABELAS E QUADROS

LISTA DE ABREVIATURAS E SIGLAS

IBGE – Instituto Brasileiro de Geografia e Estatística

SIDRA – Sistema IBGE de Recuperação Automática

IDR-PR – Instituto de Desenvolvimento Rural do Paraná

Embrapa – Empresa Brasileira de Pesquisas Agropecuárias

DERAL – Departamento de Economia Rural

KDD – *Knowledge Discovery in Databases*

IDE – *Integrated Development Environment*

SUMÁRIO

1 INTRODUÇÃO

Com o significativo avanço das tecnologias da informação e na versatilidade para armazenamento de grandes quantidades de informações, aumentou exponencialmente o volume de dados armazenados em diversos locais do mundo. Esses dados são gerados por diferentes setores, entre os quais: agropecuário, industrial, comercial, saúde, educação, etc. Caldas *et al*. (2006) aponta que diariamente um enorme volume de dados é gerado e armazenado nas organizações, e são esses dados que fornecem suporte às decisões a serem tomadas.

O Instituto Brasileiro de Geografia e Estatística (IBGE) conduziu a realização do Censo Agropecuário Brasileiro, e foi identificado mais de 5 milhões de estabelecimentos rurais no Brasil, os quais ocupam profissionalmente mais de 15 milhões de pessoas no meio rural (IBGE, 2019). O levantamento dessas informações ocorreu no período compreendido de 2017 a 2018, com resultados divulgados em 2019, por meio do Sistema IBGE de Recuperação Automática (SIDRA), compilados em um vasto banco de dados. A plataforma SIDRA reúne dados do último censo, assim como de edições anteriores realizadas em 1995 e 2006, oferecendo informações para todo o território brasileiro, agrupados em diferentes estratos geográficos, incluindo classificação por município. Trata-se de um amplo banco de dados, com múltiplas variáveis, em que as mudanças no setor agropecuário podem ser encontradas, insuficientemente compreendidas e analisadas.

A plataforma SIDRA é uma importante fonte de informação, passível da aplicação de técnicas de mineração de dados e análises estatísticas, uma vez que se trata de uma base pública com alta disponibilidade de dados por diferentes delimitações geopolíticas. Porém, esses dados armazenados são pouco explorados, considerando o grande volume de informações e a dificuldade de extrair e analisar muitas variáveis em conjunto de forma manual.

Estudos na literatura evidenciam a importância da análise dos dados secundários para avanços na ciência e na tomada de decisão nos setores público e privado. A exemplo, Luz (2017) afirma em seu trabalho que é essencial uma empresa obter informações estratégicas, relativas ao contexto de tomada de decisão, pois tais informações possibilitam um planejamento rápido relacionado às mudanças das condições do negócio, na atual conjuntura de um mercado globalizado. Caldas *et al*. (2006) afirmam que competitividade é uma palavra que se torna cada vez mais presente nas organizações, de modo que na era do conhecimento a atividade de adquirir, tratar, interpretar e utilizar a informação de forma eficaz é o que promove o diferencial estratégico nas empresas.

Diferentes órgãos públicos, como: Instituto de Desenvolvimento Rural do Paraná (IDR-Paraná), Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA), Departamento de Economia Rural (DERAL), prefeituras, e diferentes organizações relacionadas ao meio rural e o setor produtivo, faz o uso de dados do IBGE para compreender as características dos produtores rurais, dos estabelecimentos agropecuários e das atividades produtivas, oferecendo um vasto diagnóstico sobre diferentes contextos rurais, o que facilita na tomada de decisões mais assertivas sobre determinadas tendências e demandas tanto regionais quanto nacionais, trazendo possibilidades de negócios aos seus produtores.

Devido a importância dos dados gerados pelo IBGE e as dificuldades que se tem ao tentar encontrar relações entre os dados, sem o uso de uma ferramenta adequada, esse trabalho propõe o desenvolvimento de um *framework* de mineração de dados, com intuito de facilitar e impulsionar análises sobre os dados da plataforma SIDRA por diferentes pesquisadores e agentes de desenvolvimento de órgãos públicos e privados, permitindo tomadas de decisões mais assertivas, reduzindo o tempo e o custo das análises.

1.1 OBJETIVOS

Diante da contextualização apresentada, os objetivos norteadores deste estudo estão organizados em geral e específicos.

1.1.1 Geral

Construir um *framework* baseado em mineração de dados para análises dos dados do Censo Agropecuário Brasileiro.

1.1.2 Específicos

1) Implementar um módulo de captura de dados agropecuários a partir da plataforma SIDRA;
2) Implementar um módulo de pré-processamento de dados para padronizar, organizar ou transformar os dados capturados;
3) Implementar um módulo de mineração de dados como agrupamento e classificação;
4) Implementar um módulo de análise e visualização de dados como medidas de similaridade e gráficos de representação de agrupamentos;

5) Realizar um estudo de caso com o *framework* desenvolvido.

1.2 CONTRIBUIÇÕES DO TRABALHO

O desenvolvimento deste trabalho será uma contribuição para pesquisadores e demais atores envolvidos com o desenvolvimento do setor agropecuário brasileiro, possibilitando um rápido e eficiente processo de extração de informações do censo agropecuário do IBGE.

1.3 JUSTIFICATIVA

No portal da Confederação da Agricultura e Pecuária do Brasil (CNA, 2017), o superintendente técnico Bruno Lucchi ressalta que o Censo Agropecuário é a principal ferramenta que baliza e define as estratégias para adoção de programas essenciais ao crescimento do setor, fortalecendo a agricultura e a pecuária do país. A análise de grandes volumes de dados utilizando métodos tradicionais como o desenvolvimento de cálculos manuais a partir de compilado de planilhas, torna-se inviável e pode incorrer no erro, uma vez que a grande quantidade de dados pode causar equívocos (DONI, 2004).

Diante desses dados e dos problemas relatados, torna-se crucial o desenvolvimento de uma ferramenta que facilite esse processo. Com isso, este trabalho propõe o desenvolvimento de um *framework* de mineração de dados que irá consumir tabelas do último censo agropecuário do IBGE. O *framework* utilizará técnicas de extração de conhecimentos em bases de dados com objetivo de encontrar padrões, que venha facilitar a tomada de decisão de gestores públicos e privados ligados ao meio rural.

1.4 DELIMITAÇÕES DO TRABALHO

Araújo e Pereira (2011) explicam em seu trabalho que o processo de mineração de dados utiliza ferramentas da inteligência artificial e técnicas de extração de conhecimentos em bases de dados para encontrar padrões e novas relações passíveis de interpretação humana.

Dessa forma o *framework* desenvolvido neste trabalho irá encontrar padrões e gerar gráficos, ficando a cargo do pesquisador a interpretação dessas novas relações encontradas e decidir se tal relação possui algum valor, para que assim possam se tornar uma informação útil. Braga (2019), define informação como dados que já foram processados sobre um determinado

assunto, enquanto o conhecimento refere-se a informações úteis obtidas por meio de um conhecimento de mundo da pessoa detentora do conhecimento.

As fontes dos dados utilizadas para o estudo de caso do *framework* desenvolvido neste trabalho são tabelas do último Censo Agropecuário 2017, disponíveis na plataforma SIDRA. As técnicas de mineração de dados que estão disponíveis no *framework* desenvolvido neste trabalho são as de classificação e de agrupamento de dados.

2 REVISÃO BIBLIOGRÁFICA

O conteúdo deste capítulo é baseado na literatura especializada, dando sustentação para o projeto desenvolvido neste trabalho. O mesmo está estruturado de forma a situar os avanços e contribuições sobre pesquisas em mineração de dados e a sua aplicação para análise de bancos de dados.

2.1 MINERAÇÃO DE DADOS

Esta subseção está dividida em outras subseções, pautando a mineração de dados em um contexto geral, como: contextualização, conceituações, etapas e técnicas necessárias para o desenvolvimento de um *framework* de mineração de dados.

2.1.1 Contextualização Histórica da Mineração de Dados

Li (2016) publicou um boletim no *Knowledge Discovery Nuggets* (KDnuggets)[1], apresentando uma *timeline* histórica da evolução da mineração de dados. Nesta *timeline* é demonstrado que no ano de 1763, foi publicado o artigo de Thomas Bayes sobre um teorema que relaciona a probabilidade atual com a anterior, sendo reconhecido como teorema de Bayes. Este teorema tornou-se fundamental para a mineração de dados e outras aplicações, permitindo o entendimento das realidades complexas baseadas em probabilidades estimadas. No ano de 1805, Karl Friedrich Gauss utilizou o método de regressão para determinar as órbitas dos corpos em torno do sol. A análise desse método estima a relação entre as variáveis, sendo uma das principais ferramentas utilizadas no processo de mineração de dados. Turing (1936), apresentou a ideia de uma máquina universal, capaz de realizar grandes quantidades de operações matemáticas como os nossos dispositivos eletrônicos atuais. Alguns anos depois, McCulloch e Pitts (1943) publicaram artigo em que propõem um conceito de redes neurais, descrevendo a ideia de um neurônio capaz de receber entradas, processá-las e gerar saídas. Em 1965 o doutor Lawrence Jerome Fogel fundou uma empresa com foco em resolver problemas do mundo real aplicando programação evolutiva. Em 1970 surgem os sofisticados sistemas de gerenciamento de banco de dados, permitindo a consulta e armazenagem de grandes quantidades de dados em banco de dados. Holland (1975) escreve o livro *Adaptation in Natural and Artificial Systems*

[1] https://www.kdnuggets.com/

abordando a construção de algoritmos genéticos, dando início ao campo de estudo denominado *Data Mining*. Em 1980 a empresa HNC Software registra a marca *DataBase Mining Workstation*, uma ferramenta com o intuito de construir modelos baseado em redes neurais. Shapiro (1989) formalizou o termo *Knowledge Discovery in Databases* (KDD) uma ideia de extração de conhecimento em base de dados. Em 1990 o termo mineração de dados surgiu com empresas fazendo a análise de dados para reconhecer tendências de mercado e aumentar sua quantidade de clientes. Boser, Guyon e Vapnik (1992) propõem uma melhoria na *Support Vector Machine* (SVM), permitindo a criação de classificadores não lineares. Somente em 2001 que o termo Ciência de Dados é introduzido por Willian S. Cleveland como uma disciplina independente, Cleveland (2001). Em pouco tempo, Michael Lewis publica o livro Moneyball, que apresenta uma abordagem estatística baseada em dados. O Oakland Athletic, clube de futebol americano, utilizou essa abordagem para medir a qualidade de seus jogadores, montando um time de sucesso. Em 2015 DJ Patil se torna o primeiro cientista chefe de dados da Casa Branca (Li, 2016).

Nesta mesma publicação, Li (2016) afirma que a partir de 2015 os conceitos de mineração de dados têm sido amplamente utilizados nas mais diversas áreas de trabalho, como na mineração de transações de cartão de crédito, movimentação do mercado de ações, segurança nacional, sequenciamento de genomas e testes clínicos. Essas são somente algumas áreas que utilizam o benefício da extração de conhecimento em bases de dados. Atualmente, as empresas devem se atentar a uma adoção de ferramentas eficientes de análise de dados, pois assim permitirá uma visualização de resultados sistematizados, adoção de decisões e escolhas mais assertivas. Particularmente, a mineração de dados pode ser aliada no processo de compreender as dinâmicas de consumo de seus clientes, recorrendo a métodos de extração de conhecimentos em suas bases de dados ou outras bases públicas.

2.1.2 Conceitos de Mineração de Dados

Na literatura é possível encontrar diferentes conceitos para mineração de dados (tradução do inglês *Data Mining*). Conforme Félix (2002), mineração de dados é a integração de um conjunto de áreas, que tem como propósito a identificação de um conhecimento obtido a partir das bases de dados que contribuem na tomada de decisões.

Por outro lado, Hand *et al*. (2007) apresenta outro conceito baseado em um modelo estatístico, em que consiste na análise de grandes conjuntos de dados, com objetivo de encontrar

alguma relação não esperada entre esses dados, facilitando a tomada de decisão pelo proprietário dos dados.

Cabena *et al*. (1998), aborda o tema como uma área interdisciplinar que utiliza um conjunto de técnicas de análise de dados, tais como: aprendizagem de máquina, reconhecimento de padrões, banco de dados, estatísticas e visualizações de dados.

No trabalho de Costa *et al*. (2013) mineração de dados pode ser definida como uma etapa de um processo mais amplo, conhecido como descoberta de conhecimento em bases de dados, denominado *Knowledge Discovery in Databases* (KDD). Em que KDD possui mais duas grandes etapas: o pré-processamento de dados e o pós-processamento obtido da mineração dos dados.

Segundo Dantas *et al*. (2008) mineração de dados é um processo de pesquisa em grandes quantidades de dados para extração de conhecimento, utilizando técnicas de Inteligência Computacional para procurar relações de similaridade ou discordância entre os dados, com objetivo de encontrar padrões, irregularidades, regras e assim transformar dados em informações relevantes.

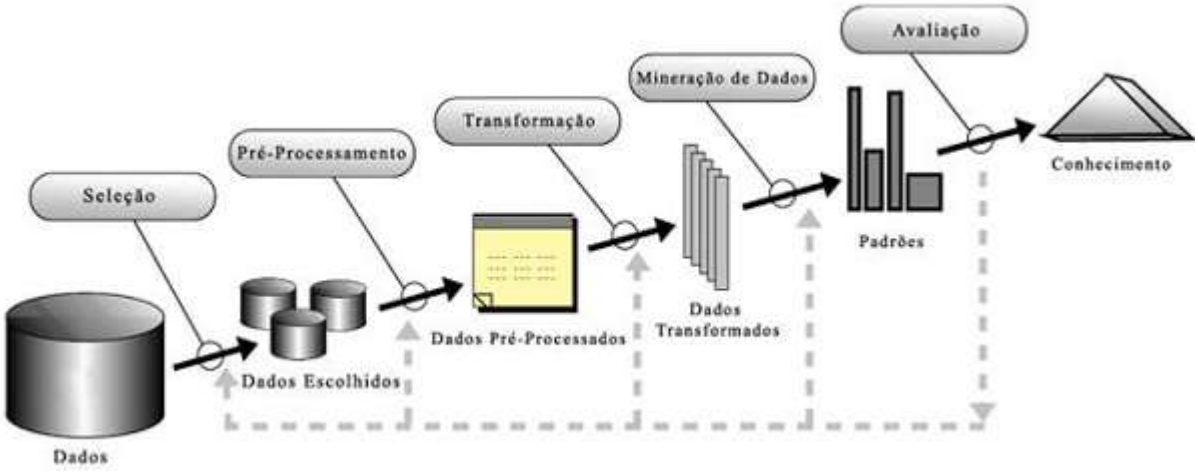
Baseado na contextualização histórica e nos conceitos da literatura, pode-se dizer que mineração de dados é a aplicação de técnicas e algoritmos que fazem um processamento dos dados retirados de outras bases, na qual analistas podem extrair conhecimentos a partir da classificação obtida no processo.

2.1.3 Processo de Extração de Conhecimentos em Banco de Dados

Na subseção anterior, pode-se observar que mineração de dados é um processo que utiliza de técnicas e algoritmos para extrair um determinado conhecimento de uma base de dados, como definido no trabalho de (COSTA *et al*. 2013). Esse processo é denominado KDD constituindo de várias etapas que se corresponde a não trivial e interativo em prol de uma identificação de padrões compreensíveis, válidos, novos e potencialmente úteis a partir de grandes conjuntos de dados (FAYYAD, 1996).

FAYYAD (1996) explica que a extração de conhecimento em base de dados KDD é estruturada em cinco etapas: seleção dos dados, pré-processamento, transformação, mineração, validação e visualização de resultados. Essas etapas são ilustradas na Figura 1, que sistematizam os processos envolvendo a mineração de dados.

Figura 1 - Etapas do processo KDD.

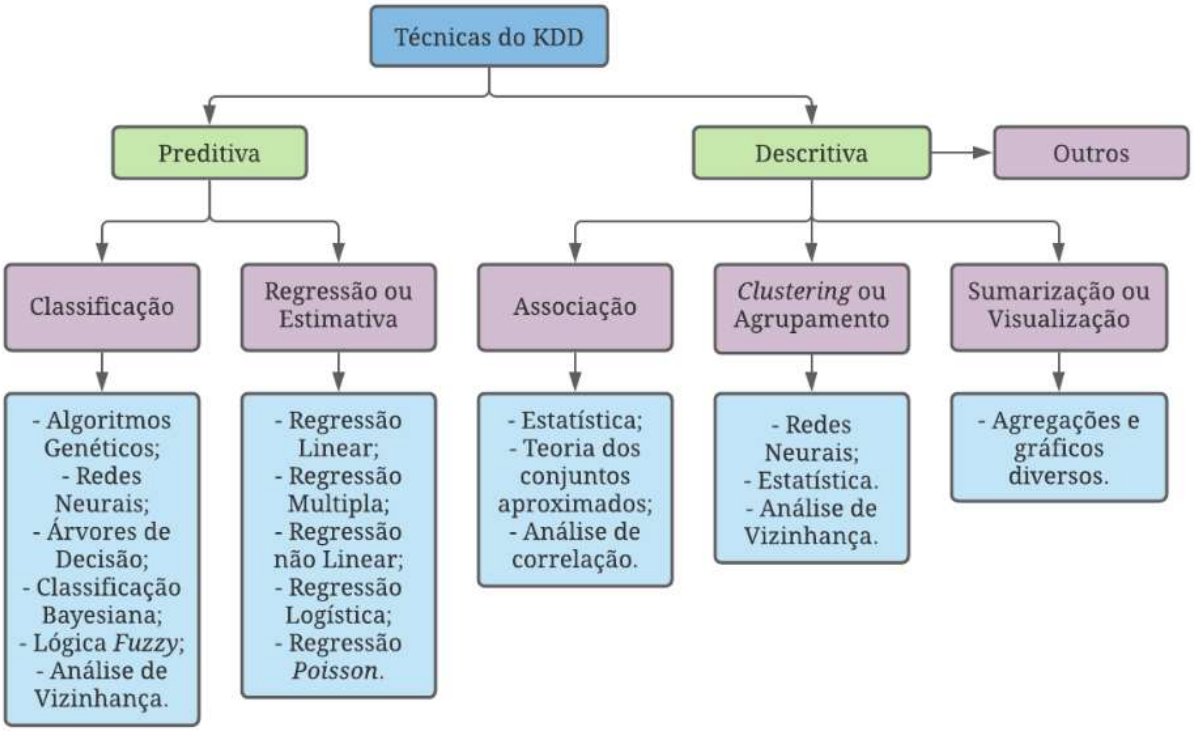


Fonte: Fayyad *et al*. (1996).

2.1.4 Técnicas do KDD

O processo do KDD funciona com base em técnicas de extração, processamento e mineração de dados, tanto de forma descritiva quanto de forma preditiva (HAN *et al*., 2011). E segundo Relich e Muszynski (2014) as técnicas mais comuns do KDD são: associação, classificação, regressão, agrupamento e visualização de dados. A Figura 2 as tarefas do KDD e as técnicas de mineração de dados.

Figura 2 - Algumas tarefas do KDD e suas técnicas de mineração de dados.



Fonte: Adaptado de Ferreira *et al*. (2018).

Segundo Monteiro (2018), uma análise preditiva consiste em avaliar um conjunto de dados, detectar padrões e selecionar alguma informação importante que possibilita uma organização a transformar tendências em resultados quantitativos. Morais (2010) define análise descritiva ou estatística descritiva como um conjunto de técnicas analíticas, utilizadas para resumir um conjunto de dados estudados, que são organizados geralmente, por meio de números, tabelas e gráficos.

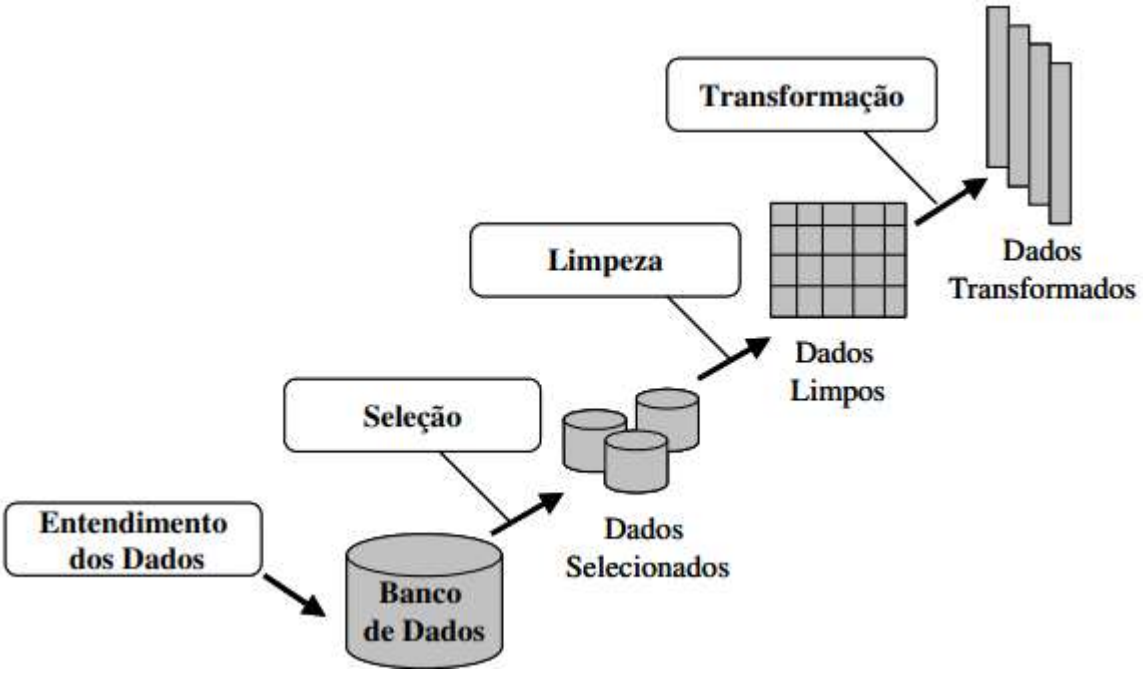
Dentro das análises preditivas e descritivas, existem uma diversidade de algoritmos que auxiliam no processo de extração de conhecimento. Na figura 2 é apresentado alguns algoritmos das principais técnicas do KDD, dentre os algoritmos pode destacar os mais utilizados na literatura, como: redes neurais, árvores de decisão, análise de vizinhança (*K-Means*), classificações bayesianas, regressão linear e outros. Nas seções seguintes serão apresentados com maior detalhamento as técnicas e algoritmos que serão utilizadas neste trabalho.

2.2 PRÉ-PROCESSAMENTO

Schmitt *et al*. (2005) explica que o pré-processamento tem por objetivo melhorar a qualidade dos dados para a etapa de mineração, pois é comum encontrar registros nas bases de dados que estão incompletos, inconsistentes, discrepantes e entre outros problemas. Nesta etapa é feito a análise inicial dos dados, como: normalização, discretização de dados contínuos, transformações, concatenações e etc.

Na Figura 3 é apresentado as fases do pré-processamento de dados baseado na metodologia do KDD, onde as etapas são: entendimento, seleção, limpeza e transformação dos dados. Logo abaixo segue a explicação de cada etapa de acordo com as conceituações de Neves *et al*. (2003).

Figura 3 - Fases do pré-processamento de dados.



Fonte: Neves *et al*. (2003).

A etapa de entendimento dos dados consiste em uma análise dos dados que vão ser trabalhados, realizando o entendimento de que se trata as tabelas envolvidas, levando em consideração o significado, formato, relevância, tamanho e o tipo dos dados. Realizar a identificação dos atributos chaves que possam ser utilizados como referência no processo de mineração de dados.

No processo de seleção dos dados, é feita a escolha das tabelas, instâncias e atributos que possam ter relação com os objetivos do trabalho. É também nesta etapa onde é realizado a

concatenação de tabelas se assim for necessário, para alcançar os objetivos esperados do trabalho.

Na etapa de limpeza dos dados é realizada a padronização, tratamentos de campos nulos, eliminação de dados que não condiz com a realidade estudada e ou dados duplicados. Esta etapa garante a qualidade dos dados a serem utilizados no processo de mineração de dados.

A fase de transformação de dados consiste em realizar conversões de valores simbólicos para valores numéricos, normalização dos dados, discretização e composição dos atributos. Essas transformações fazem com que os dados sejam apresentados de forma apropriada ao processo de mineração de dados.

É de suma importância utilizar essas técnicas de pré-processamento de dados no *framework* que será desenvolvido neste trabalho, uma vez que os dados trabalhados são provenientes da base SIDRA, algumas tabelas existem campos nulos e também um cabeçalho com diversas descrições que não tem utilidade no processo de mineração de dados, existem dados em tabelas que podem ser concatenados com outras tabelas e entre outras transformações que se julga necessária.

2.2.1 Normalização dos Dados

A normalização é geralmente aplicada na etapa de pré-processamento e consiste em modificar dados de um conjunto, para que todos fiquem em uma mesma escala, sem causar perdas ou distorções, minimizando problemas oriundos de dispersões distintas entre os dados (Blanca, 2020).

Pino (2014) afirma que a normalização faz algumas transformações no conjunto de dados, com o objetivo de produzir uma escala que reduz a variância e a assimetria, aproximando a variável da distribuição normal.

2.2.1.1 Técnica Z-Score

O método *Z-Score* descreve a relação de um valor com a média do grupo de valores e seu desvio padrão, resultando em uma pontuação (XIE, 2019). A fórmula se corresponde por meio da subtração do valor a ser normalizado pela média do conjunto de valores, dividido pelo desvio padrão do conjunto. Vale ressaltar que a pontuação pode ser positiva ou negativa, quando for positiva significa que a pontuação está acima da média, quando negativa a pontuação está abaixo da média. Assim, a equação Z-Score é representada por

$$v' = \frac{v - \mu}{\sigma} \quad (1),$$

sendo que μ é a média, σ o desvio padrão e *v* o valor a ser normalizado e *v´* o novo valor normalizado.

O método *Z-Score* pode ser aplicado a uma faixa de valores de dados agropecuários, a exemplo: produtores que possuem uma quantidade de terras entre 10 a 40 mil hectares, então é possível normalizar a quantidade de terra de um produtor que possui 18 mil hectares, suponhamos que a média seja 26 mil hectares e o desvio padrão de 8 mil hectares, podemos aplicar a Equação 1 *Z-Score* da seguinte forma: ($X = (18000 - 26000)/8000 = -1$).

McLeod (2019), descreve a importância de padronizar dados brutos em uma distribuição normal, convertendo os mesmos em pontuações Z, pois isso permite que pesquisadores possam calcular a probabilidade de uma pontuação ocorrer dentro de uma distribuição normal. Também permite realizar comparações entre duas pontuações que estão em diferentes amostras, podendo ter diferentes médias e desvios padrão.

2.2.1.2 Técnica Min-Max

Min-Max é um outro método utilizado para normalização de dados, em que, dado um conjunto de valores, o método utiliza o menor e o maior valor do conjunto para criar uma faixa de valores, assim é possível transformar qualquer valor que esteja dentro dessa faixa em um novo valor entre 0 e 1 (LOPES, 2019). Caso tenha resultados negativos a faixa de valor pode ficar entre -1 e 1, por meio da equação

$$Xchanged = \frac{X - Xmin}{Xmax - Xmin} \quad (2),$$

na qua lque X é o valor a ser normalizado, *Xmin* e *Xmax* é o menor e o maior valor do conjunto de dados e, *Xchanged* é o valor normalizado resultante do cálculo.

O método Min-Max pode ser aplicado ao mesmo exemplo de dados agropecuários descrito na subseção 2.2.1.1, aplicando a equação ($x = (18000 - 10000)/(40000 - 10000) = 0{,}2666..$).

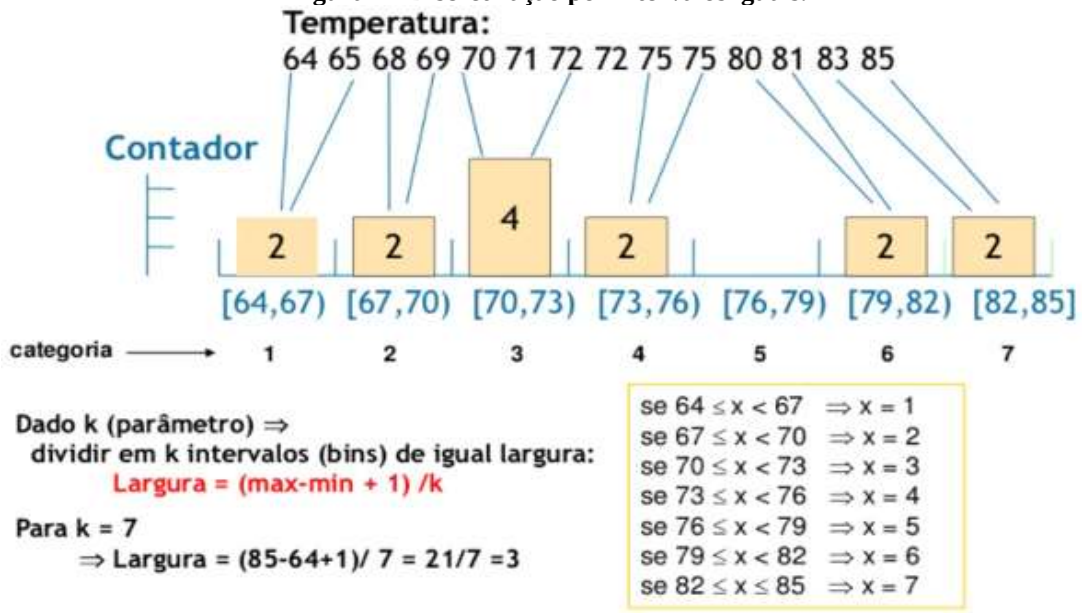
2.2.2 Discretização dos Dados Contínuos

A discretização é o processo de transformar uma variável contínua em outra discreta (YANG, 2003). Yoneyama (2003) afirma que o uso de variáveis discretas pode tornar o processo de aprendizagem mais simples e eficiente, diminuindo a necessidade de um poder de processamento e armazenamento maior.

Existem dois métodos de discretização que são considerados simples e eficientes, os de intervalos iguais e os de frequências iguais. O primeiro método gera K intervalos com tamanhos iguais, onde K é um número inteiro maior que zero definido pelo usuário (uniforme sklearning). Por outro lado, o método de frequências iguais desenvolve K intervalos de forma que cada intervalo contenha aproximadamente o mesmo número de ocorrência e valores (quantil) (YONEYAMA 2003).

Na Figura 4 pode-se observar um exemplo de dados de temperatura discretizados por intervalos iguais, em que, dado um *K* igual a 7 é aplicado a fórmula da largura, obtendo um intervalo de tamanho 3. Então os dados são agrupados entre o menor e o maior valor de cada intervalo.

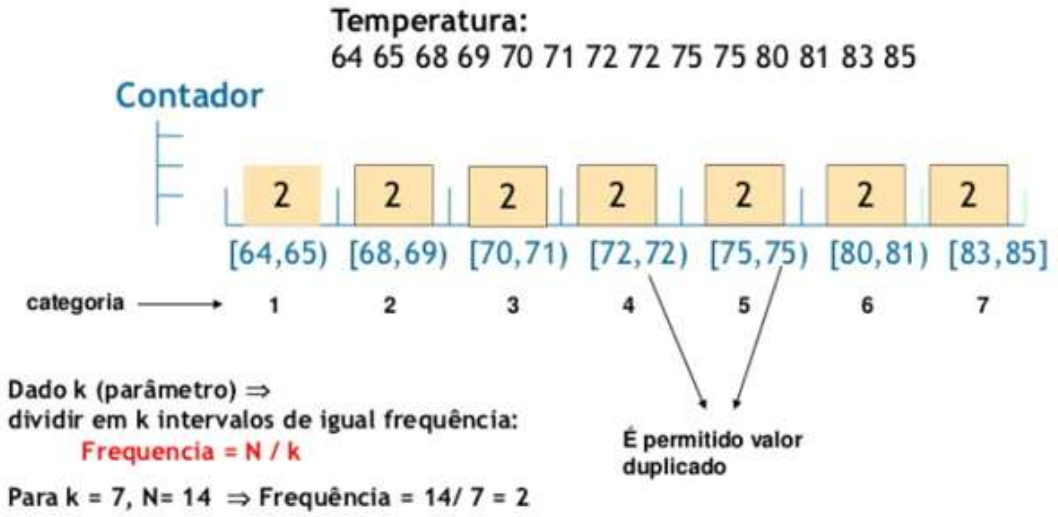
Figura 4 - Discretização por intervalos iguais.



Fonte: Castillo (2011).

A Figura 5 é um exemplo de discretização por frequência igual, a partir de dados de temperaturas. Dado um *K* igual a 7, *N* igual a 14, sendo que *N* é a quantidade de dados da temperatura, a frequência é calculada a partir da fórmula *N/K*. No exemplo da figura a frequência é igual a 2, então os dados são agrupados a cada 2 elementos.

Figura 5 - Discretização por frequências iguais.



Fonte: Castillo (2011).

2.2.3 Transformações gerais nos dados

O processo de transformação de dados consiste em realizar modificações no conjunto de dados estudados, com o objetivo de obter um modelo de classificação com maior assertividade. Essas modificações podem ser a remoção de campos, células nulas ou valores que não condiz com o conjunto de dados estudados. Outra técnica utilizada, além da remoção do campo ou da célula, é o preenchimento de campos nulos com a média ou mediana do conjunto de dados analisados (GOMES, 2019).

Outra técnica que pode ser utilizada é o *replace*, onde os dados podem ser alterados para uma outra dimensão, por exemplo, um conjunto de dados que possui letras maiúsculas e minúsculas, para melhorar o processamento faz-se necessário realizar uma padronização, então pode ser usado o método *replace* para modificar todos os dados para que fique em um só formato, maiúsculo ou minúsculo. Essa técnica ainda pode ser utilizada para padronizar números com casas decimais, valores de moedas e etc (GOMES, 2019).

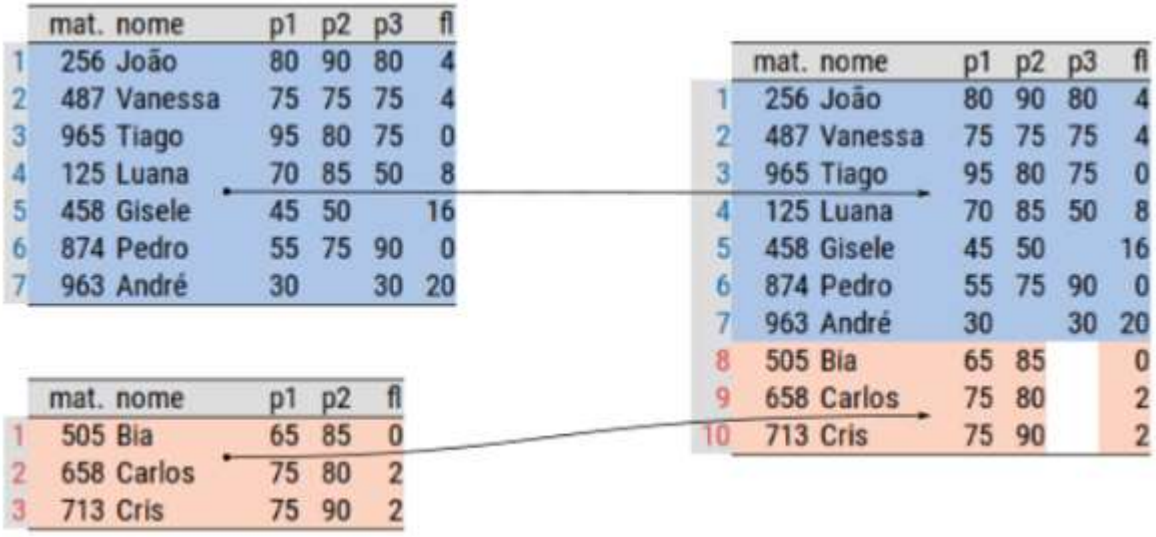
Pode ser utilizado uma técnica de agrupamento de dados não numéricos e calcular a quantidade de ocorrência desse dado e assim popular uma outra tabela com os dados numéricos dos agrupamentos (GOMES, 2019). Obter dados numéricos pode facilitar o processo de classificação e até mesmo melhorar o percentual de acerto do modelo.

2.2.4 Transformações em tabelas e colunas

Quando se trabalha com bases de dados que possui uma grande quantidade de tabelas, pode acontecer que as informações fiquem desconexas, ou seja, em espaços totalmente diferentes. Com isso, para alcançar um conjunto de dados coerente e que possa ser usado no processo de agrupamento e classificação é necessário que várias modificações sejam realizadas no conjunto de tabelas, a fim de se obter um conjunto de dados ideal para o processo de mineração de dados (GOMES, 2019).

Uma das transformações que podem ser realizadas é a concatenação de tabelas e colunas. Isso pode ser útil quando queremos facilitar o processo de agrupamento dos dados, pois uma tabela pode conter conjuntos de dados espalhados em outras tabelas. Realizar concatenações entre as tabelas fazendo com que os dados fiquem mais ou menos agrupados, podem facilitar e melhorar o desempenho do processo de agrupamento e classificação dos dados (GOMES, 2019). A Figura 6 exemplifica uma concatenação de tabelas, onde duas tabelas possuem dados relativamente parecidos.

Figura 6 - Concatenação de tabelas.

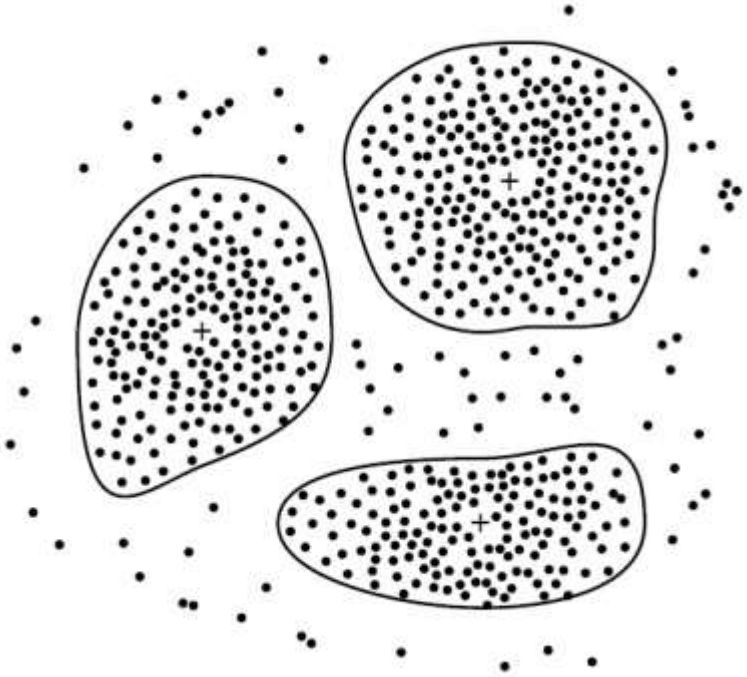


Fonte: Zeviani (2019).

2.3 AGRUPAMENTO

Segundo Santos (2009), técnicas de agrupamento ou *clustering* tem por objetivo identificar grupos de dados que possuem uma certa similaridade entre si. A Figura 7 exemplifica a existência de uma quantidade x de dados, os quais possuem certa similaridade em determinadas regiões, então pode-se agrupar e rotular esses dados em grupos separados por regiões.

Figura 7 - Agrupamento de dados.



Fonte: Camilo *et al*. (2009).

Quando se faz uma análise de dados, é necessário que os dados estejam de forma que possam ser extraídas certas conclusões, ou seja, devem estar separados em classes ou grupos. Porém, quando a quantidade de dados tem uma amplitude maior, é difícil classificá-los sem o apoio de uma ferramenta. Desse modo, os algoritmos de agrupamento formam grupos considerados naturais de acordo com alguma métrica, para que possam ser processados posteriormente como objetos correspondentes a uma mesma categoria (SANTOS *et al*. 2009).

Na maioria das tarefas realizadas com técnicas de agrupamento, os atributos são numéricos, devido a aplicação de algoritmos que calculam a distância entre os registros para agrupar os mesmos (SANTOS *et al*. 2009). Porém extensões que consideram atributos numéricos e não numéricos de forma separados podem ser implementados.

Aplicar técnicas de agrupamento de dados antes do processo de classificação é um passo muito importante, pois o processo de classificação exige que os dados sejam previamente categorizados. Santos *et al*. (2009) fala sobre a diferença entre as técnicas de agrupamento e as técnicas de classificação que necessita de grupos predefinidos, onde as de agrupamento usa métricas definidas, para formar grupos onde os dados são semelhantes entre si e diferentes de outros grupos de dados.

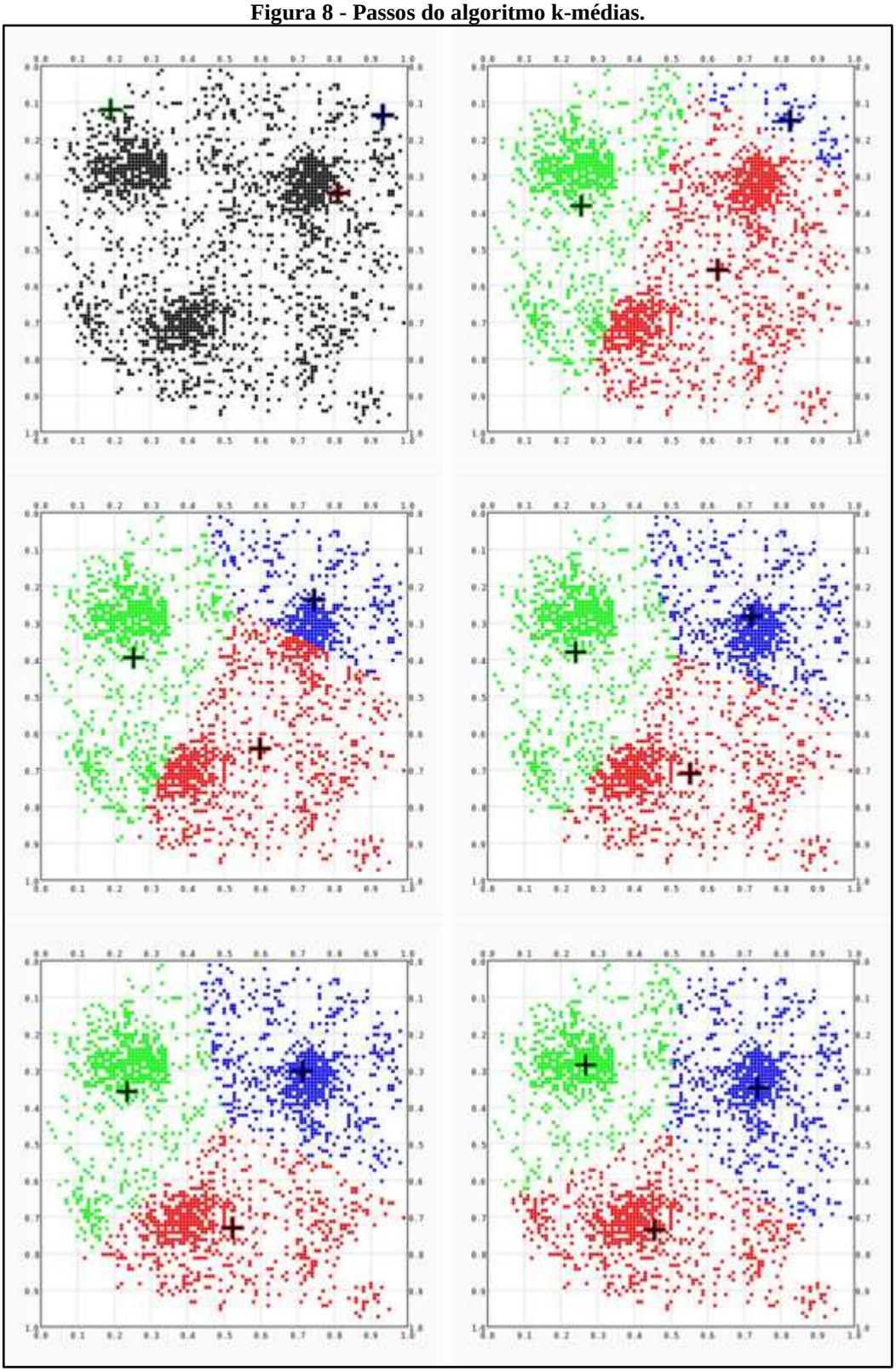
Segundo a literatura, os algoritmos mais utilizados no processo de agrupamento de dados, são os de análise de vizinhança, especificamente o K-Médias, que serve de base para inúmeros outros algoritmos.

2.3.1 Algoritmo K-Médias

O algoritmo K-Médias utiliza um valor K que corresponde ao número de grupos a ser formado, uma métrica de cálculo da distância entre dois registros e condições para finalizar as interações entre si. O algoritmo cria uma quantidade K de pontos com valores inicialmente randômicos, faz a primeira interação marcando cada registro como pertencente ao ponto mais próximo e depois recalcula os pontos dos grupos de acordo com os registros pertencentes (ou mais próximos) a esses, deve ser definida uma métrica de qualidade dos agrupamentos, essa métrica é usada como condição de parada das interações (SANTOS *et al*. 2009).

Na Figura 8 é ilustrado seis passos do algoritmo K-Médias com um K = 3, ou seja, vai ser definido aleatoriamente 3 pontos dentro de um conjunto artificial de dados, e dois atributos numéricos com valores variando entre 0 e 1. Como K é igual a 3, os pontos serão divididos em 3 grupos, inicialmente com bastante ruído. A partir da primeira interação, são marcados três pontos (centróides) aleatórios (indicado por pequenas cruzes) com cores diferentes para distinguir os grupos. Pode-se observar que os centróides vão mudando de posição, por meio de uma função de cálculo de distância, esses centróides tentam encontrar um local onde possuem mais concentração de pontos, ou seja, onde os pontos possuem menor distância entre os conjuntos de dados. Até que seja possível separar os dados por grupos.

Figura 8 - Passos do algoritmo k-médias.



Fonte: Santos *et al*. (2009).

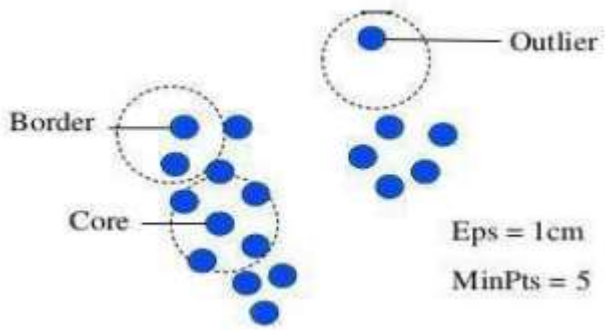
Existem outros algoritmos que utilizam conceitos semelhantes ao K-Médias, um dos mais conhecidos é o Fuzzy C-Médias (BEZDEK, 1981). O mesmo utiliza os conceitos de lógica

nebulosa para calcular a pertinência de um dado a um grupo como sendo um valor contínuo entre 0 e 1, enquanto o K-Médias considera uma pertinência booleana, onde o dado pertence a um grupo e somente àquele grupo (SANTOS *et al*. 2009).

2.3.2 DBSCAN Clusterização Espacial Baseado em Densidade de Aplicações com Ruído

DBSCAN é um algoritmo de clusterização que se baseia em densidade, sendo utilizado para descobrir clusters com formas arbitrárias (MARIA DAS GRAÇAS *et al*. 2010). Segundo Sousa (2016) para iniciar qualquer processo utilizando o DBSCAN, é necessário definir dois parâmetros. O primeiro é o Eps, que define o raio máximo da vizinhança, ou seja, a distância máxima que um ponto pode ter um do outro, para ser considerado vizinho. O segundo parâmetro que deve ser definido é o MinPts, é a quantidade mínima de dados que deve ter dentro do raio para que seja considerado um cluster. Esse processo de clusterização é melhor ilustrado na Figura 9.

Figura 9 - Clusterização com DBSCAN.



Fonte: Sousa (2016).

Sousa (2016), explica o processo do algoritmo DBSCAN, onde o algoritmo faz o agrupamento dos dados por meio de clusters como os outros algoritmos que utilizam as técnicas de clusterização. A diferença entre o DBSCAN é a forma em que é montado os clusters. A técnica utilizada é baseada nos parâmetros Eps e MinPts, em que escolhido um determinado dado é aplicado o parâmetro Eps que define a distância máxima entre o dado escolhido e a borda da circunferência que é gerada em torno do dado. Todo dado que esteja entre o dado central e a borda da circunferência é considerado um dado vizinho. O segundo passo é o parâmetro MinPts que define a quantidade mínima de dados que deve conter dentro do raio do

Eps. Se as duas condições forem atendidas, um novo cluster é criado com os dados agrupados dentro da circunferência.

Pode-se observar na Figura 9, que o parâmetro da distância máxima entre dois pontos para que seja considerado vizinho, é de 1 centímetro, e a quantidade mínima de pontos que deve estar contido dentro do raio formado a partir do ponto escolhido é igual a cinco. Se P for um ponto central e as duas condições do algoritmo forem atendidas, um novo cluster é formado. Se P for um ponto de borda, então não haverá pontos alcançáveis, sendo assim, visita-se o próximo ponto e o processo é repetido, até que não haja mais pontos dentro do raio. Se o raio de P não encontrar nenhum ponto que atenda às condições, P é caracterizado como um *outlier*.

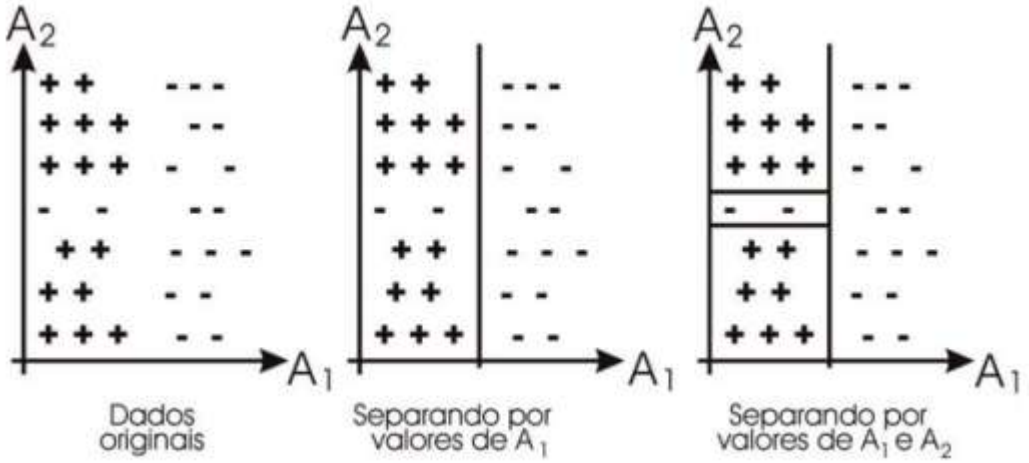
2.4 CLASSIFICAÇÃO

Para Sandra Amo (2004) descreve a classificação como um processo de identificar conjuntos, modelos ou funções que possam descrever e separar classes ou conceitos, com o propósito de utilizar o modelo para predizer classes de objetos que ainda não foram classificados. Santos *et al*. (2009) define classificação como a descoberta de uma função preditiva capaz de classificar um dado em uma de várias classes discretas que são preditivas ou conhecidas. Baseado na definição de Santos *et al*. (2009), a partir de um conjunto de classes conhecidas, ou um modelo de classes preditivas, é possível classificar um outro dado desconhecido como pertencente a uma das classes do conjunto de predição.

Cada classe possui um conjunto de dados que corresponde a um padrão de valores dos atributos previsores, podendo ser considerados como a descrição da classe, (SCHMITT, 2013). Um conjunto de classes definido como C e, a cada classe C*i*, corresponde a uma descrição D*i* das propriedades selecionadas. Dessa forma, utilizando essas descrições é possível construir um classificador que descreve um exemplo E dentro de um conjunto T de exemplos, na qual E pertence à classe C*i*, quando E satisfaz D*i* (CARVALHO, 2005).

Baseado nos conceitos acima, o principal objetivo do desenvolvimento de um classificador é a descoberta de alguma relação entre os atributos previsores e as classes. Na Figura 10 é ilustrado um exemplo de classificador, que tem como objetivo identificar a relação entre os atributos previsores A1 e A2 e os valores da classe “+” e “-”. A construção do classificador é baseada no particionamento recursivo do espaço de dados, dividindo em áreas e subáreas, a fim de obter a separação das classes.

Figura 10 - Exemplo de classificação.

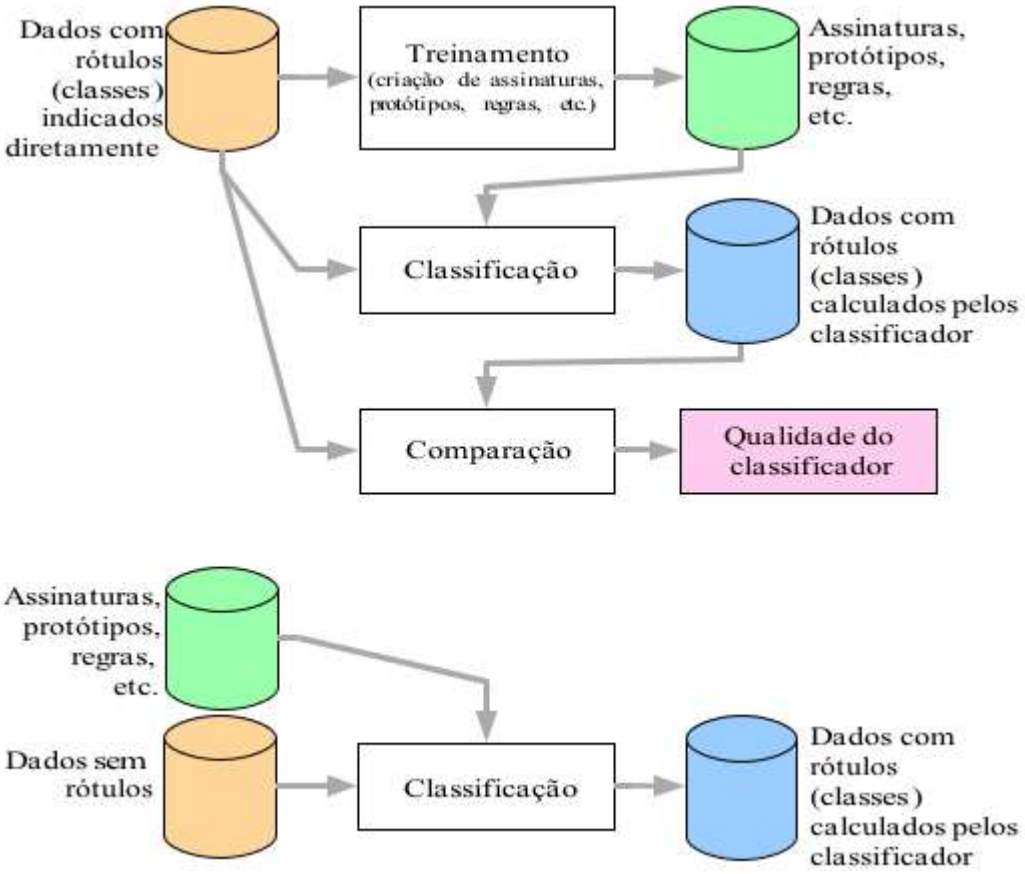


Fonte: Schmitt (2013).

Schimitt (2013) explica um conceito importante da etapa de classificação, onde os dados são divididos em dois conjuntos, um de treinamento e outro de testes. Inicialmente o conjunto de testes é disponibilizado e analisado, e a partir desse conjunto um modelo de classificação é construído, sendo utilizado para classificar outros conjuntos de dados que não fazem parte dos dados utilizados no treinamento. Vale ressaltar que cabe ao analista de dados verificar se o modelo gerado é um bom modelo de classificação. O analista vai definir uma porcentagem de aceitação, se o modelo atingir essa porcentagem, pode se dizer que, para o problema em que está sendo aplicado, o modelo classificador atende as necessidades.

De acordo com Santos *et al*. (2009), os autores afirmam em seu trabalho que uma das técnicas mais utilizadas para o desenvolvimento de modelos a partir de dados, são as que envolvem o uso de funções de classificação para classificar dados em categorias discretas, e que o ponto central dessas técnicas é justamente a criação da função. Na Figura 11, é ilustrado o processo geral de classificação.

Figura 11 - Processo de classificação supervisionada de dados.



Fonte: Adaptado de Santos *et al*. (2009).

Após rotular os dados e definir suas respectivas classes, é necessário desenvolver uma função de classificação que saiba diferenciar ou associar os valores dos atributos a uma determinada classe em descritores, que podem ser regras, protótipos, assinaturas e etc. daquela classe.

Os algoritmos de classificação e os conjuntos de descritores podem ser utilizados de duas formas, onde na primeira, demonstrado na parte superior da Figura 11, os mesmos dados utilizados para o treinamento do classificador, são utilizados na classificação, isso para mensurar a qualidade do classificador. Já na segunda forma, ilustrado na parte de baixo da Figura 11, os dados que são usados na classificação são diferentes dos dados que foram utilizados no treinamento do classificador.

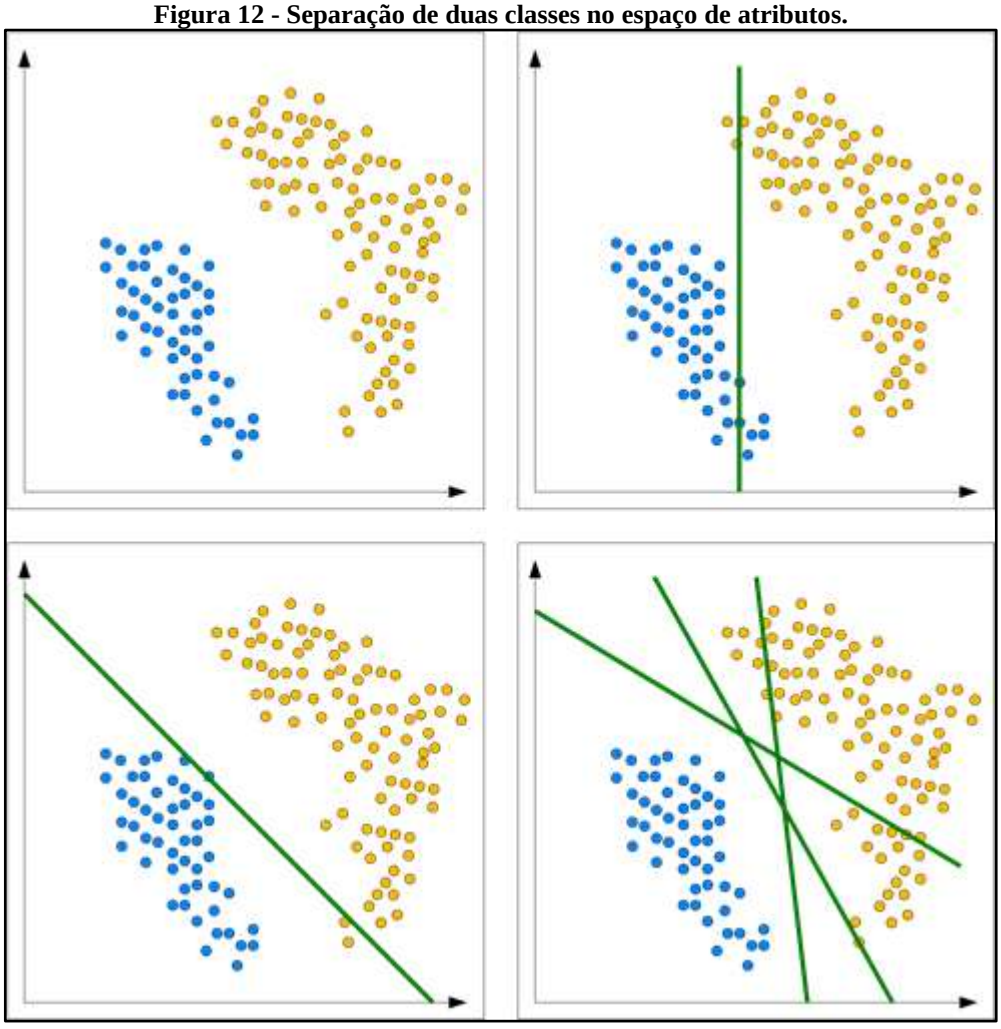
A primeira forma geralmente é utilizada no processo de construção do classificador, onde a técnica de validação cruzada é aplicada. Essa técnica consiste em dividir a base de treinamento em N partes, onde N-1 são dados rotulados utilizados no classificador, e o restante dos dados são utilizados na classificação. Esse processo é repetido N vezes, até que todos os

dados da base de treinamento sejam utilizados como dados para testes e dados para classificação (MAGALHÃES *et al*., 2012).

A segunda forma é a mais utilizada segundo Santos *et al* (2009), onde possui duas bases de dados, uma mantendo os registros rotulados, para gerar o modelo de classificação, e a outra base mantém os registros que se pretende classificar. Nesse caso não é necessário aplicar a técnica de validação cruzada.

Na Figura 12, é ilustrada uma forma simplificada do resultado da aplicação de uma função classificadora que cria partições ortogonais aos eixos dos atributos, com o objetivo de encontrar a melhor divisão das classes.

Figura 12 - Separação de duas classes no espaço de atributos.



Fonte: Santos *et al*. (2009).

Existem vários algoritmos que fazem o processo de classificação de dados. Em consulta a literatura, identificou-se alguns algoritmos comumente utilizados, como os de árvores de decisão e os baseados no teorema de Bayes.

2.4.1 Árvores de Decisão

Garcia (2003), conceitua árvores de decisão como uma representação simples de conhecimento e um meio eficiente de construir classificadores capazes de predizer classes com base em valores de atributos advindos de um conjunto de dados. O autor ainda explica que o algoritmo utiliza a técnica de dividir para conquistar, ou seja, um problema complexo é dividido em partes menores, tornando o problema mais simples de chegar a uma solução. Recursivamente os subproblemas são resolvidos com a mesma estratégia (MONARD *et al*. 2003).

Uma árvore de decisão é construída de vários nós e arestas. Os nós se classificam em três tipos: nó raiz que define a cabeça ou ponto inicial da árvore, o nó pai que são os nós onde dão seguimentos aos demais nós na árvore e os nós folhas que são os últimos nós, ou seja, não existe nenhuma ligação com os nós folhas. Cada nó folha da árvore corresponde a uma classe do conjunto de treinamento, ou seja, cada nó folha corresponde a uma classe. Cada percurso na árvore da raiz até as folhas, corresponde a uma regra de classificação (LEMOS *et al*, 2005).

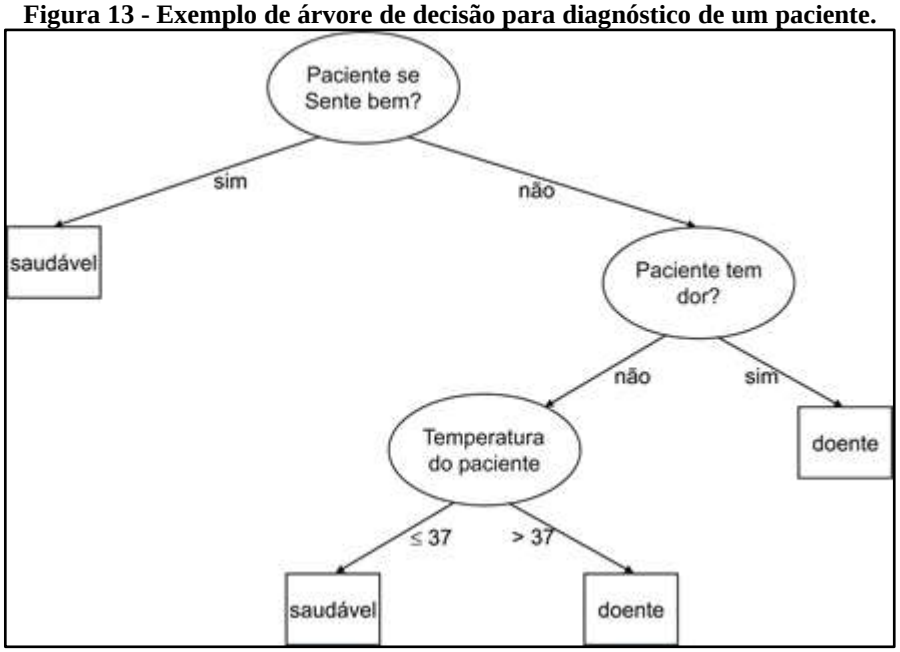
Monard *et al*. (2003) descreve que uma árvore de decisão é uma estrutura de dados definida recursivamente como:

- Um nó folha que corresponde a uma classe ou;
- Um nó de decisão que contém um teste sobre algum atributo;
- Para cada resultado dos testes existe uma aresta para uma subárvore;
- Cada subárvore tem a mesma estrutura que a árvore.

Podemos analisar as definições de Monard *et al*. (2003) com as de Garcia (2003), onde a ideia do algoritmo segue a mesma, dividir para conquistar, quebras o problema em subproblemas menores até encontrar uma solução viável.

A Figura 13 apresenta a execução de uma árvore de decisão que tem o objetivo de realizar o diagnóstico de um paciente. A árvore tem duas classes ou possibilidades, a primeira é o paciente estar doente e a segunda é estar saudável. Na figura cada elipse é um teste em um atributo do conjunto de dados do paciente, cada aresta é o resultado do teste no atributo, cada retângulo representa uma classe, ou seja, o nó folha ou o diagnóstico do paciente.

Figura 13 - Exemplo de árvore de decisão para diagnóstico de um paciente.



Fonte: Monard *et al*. (2003).

2.4.2 *Naive Bayes*

Outro algoritmo comumente utilizado na etapa de classificação é o *Naive bayes*, por ser considerado um algoritmo simples e que entrega resultados promissores na etapa de classificação de dados. O conceito do algoritmo é baseado na teoria das probabilidades, Rocha *et al*. (2008) afirma que algoritmos que se baseiam neste conceito é uma das principais fontes técnicas no processo de classificação em base de dados.

O algoritmo *Naive bayes* é baseado na teoria de Bayes, onde testa a probabilidade de um evento ocorrer, dado que um outro evento já ocorreu. Normalmente representado por P(X|Y), ilustrado na equação a seguir:

$$P(X|Y) = \frac{P(X) * P(Y|X)}{P(Y)}, \tag{3}$$

na qual, *P(X)* é a probabilidade de um evento *X* acontecer, *P(Y)* é a probabilidade de um evento *Y* acontecer e *P(X|Y)* é a probabilidde do evento *X* acontecer dado que o evento *Y* já aconteceu.

Rocha *et al*. (2008), apresenta um exemplo de classificação utilizando o algoritmo *Naive bayes*, baseado em um problema de sair ou não para jogar ao ar livre. Ele leva em consideração as questões climáticas como variáveis decisórias. O Quadro 1 apresenta as regras a serem utilizadas no algoritmo.

Quadro 1 – Exemplos de treino do problema de jogar ao ar livre.

Tempo	Temperatura (ºF)	Umidade	Vento	Jogar
Limpo	Quente	Alta	Fraco	Não
Limpo	Quente	Alta	Forte	Não
Nublado	Quente	Alta	Fraco	Sim
Chuvoso	Temperada	Alta	Fraco	Sim
Chuvoso	Fria	Normal	Fraco	Sim
Chuvoso	Fria	Normal	Forte	Não
Nublado	Fria	Normal	Forte	Sim
Limpo	Temperada	Alta	Fraco	Não
Limpo	Fria	Normal	Fraco	Sim
Chuvoso	Temperada	Normal	Fraco	Sim
Limpo	Temperada	Normal	Forte	Sim
Nublado	Temperada	Alta	Forte	Sim
Nublado	Quente	Normal	Fraco	Sim
Chuvoso	Temperada	Alta	Forte	Não

Fonte: Adaptado de Sombra *et al*. (2018).

O algoritmo *Naive bayes* é alimentado com base nos dados apresentados no Quadro 1. Pode-se observar que no exemplo de treino tem-se 4 atributos: tempo, temperatura, umidade e vento, e 2 classes: jogar igual a sim e jogar igual a não. A partir da entrada dos dados, é feito uma contagem de quantos atributos pares existem no conjunto de dados informado. A contagem finalizada é apresentada na Tabela 1.

Tabela 1 - Contagem da ocorrência de atributos pares como exemplo de cada uma das classes.

Tempo			Temperatura			Umidade			Vento		
	Sim	Não		Sim	Não		Sim	Não		Sim	Não
Limpo	2	3	Quente	2	2	Alta	3	4	Fraco	6	2
Nublado	4	0	Temperada	4	2	Normal	6	1	Forte	3	3
Chuvoso	3	2	Fria	3	1						

Fonte: Adaptado de Sombra *et al*. (2018).

A partir da frequência de cada atributo calculado pelo algoritmo, é gerada as frequências relativas com base na entrada de dados, dividindo as frequências na Tabela 1 pelo número de exemplos da respectiva classe. A Tabela 2 apresenta as respectivas frequências do problema.

Tabela 2 - Frequências relativas à entrada de dados.

Tempo			**Temperatura**			**Umidade**			**Vento**		
	Sim	Não		Sim	Não		Sim	Não		Sim	Não
Limpo	2/9	3/5	Quente	2/9	2/5	Alta	3/9	4/5	Fraco	6/9	2/5
Nublado	4/9	0/5	Temperada	4/9	2/5	Normal	6/9	1/5	Forte	3/9	3/5
Chuvoso	3/9	2/5	Fria	3/9	1/5						

Fonte: Adaptado de Sombra *et al*. (2018).

Feito essas operações, o algoritmo *Naive bayes* já está treinado e apto para classificar novas entradas de dados. Pode-se fazer um teste de classificação baseado nas seguintes entradas: “tempo” = “limpo”, “temperatura” = “quente”, “umidade” = “alta”, “vento” = “fraco”. Dado essa entrada, pode utilizar a Equação 1 para fazer os cálculos.

Para calcular a classificação do exemplo dado acima, é feito o cálculo de cada classe o valor de uma função de pertença L (*likelihood*). Segundo Sombra *et al*. (2018) o *likelihood* é obtido por meio da multiplicação das frequências relativas de cada atributo presente na entrada a ser classificada, associando a cada classe de saída presente na base de teste.

Para a entrada dada, tem-se os seguintes valores:

L(“sim”) = (2/9) * (2/9) * (3/9) * (6/9) * (9/14) = 0,642

L(“não”) = (3/5) * (2/5) * (4/5) * (2/5) * (5/14) = 0,357

Baseado nos resultados obtidos do cálculo da função de pertença L, e nos dados de entrada para classificação, pode-se dizer que as chances de ir jogar ao ar livre dada as condições de entrada são maiores que as chances de não ir jogar. Pode ainda, se achar necessário, atribuir probabilidades a cada classe, normalizando os resultados de L de forma que a soma dos dois seja 1. Assim a probabilidade (P) será:

P(“sim”) = L(“sim”) /(L(“sim”) +L(“não)) = 0,642 * 100 = (64,2%)

P(“não”) = L(“não”) /(L(“sim”) + L(“não”)) = 0,357 * 100 = (35,7%)

Utilizando os procedimentos acima, é possível realizar a classificação de dados por meio do algoritmo *Naive bayes*.

2.5 VALIDAÇÃO

Esta subseção apresenta os métodos de validação para o agrupamento e a classificação. Além disso, está dividido em outras duas subseções, uma para a validação dos algoritmos de classificação e a outra para a validação dos algoritmos de agrupamento.

2.5.1 Validação da Classificação

Após a etapa de mineração de dados, um modelo de classificação é definido baseado nos dados utilizados no treinamento do algoritmo de classificação. Para que os resultados obtidos do modelo sejam aceitos, o modelo deve ser submetido a um processo de validação, onde serão analisadas algumas métricas, como: matriz de confusão, precisão, *recall* e *F1-Score*. Essas métricas serão melhor explicadas nas subseções abaixo.

2.5.1.1 Matriz de confusão

Segundo Souza (2019), dado um modelo de classificação com um conjunto N de classes, uma matriz N x N é gerada, denominada matriz de confusão. Uma matriz de confusão apresenta todas as classes na vertical da matriz e todas as classes na horizontal da matriz, então é populada com a quantidade de atributos classificados corretamente e incorretamente (SOUZA, 2019). Se o modelo de classificação apresentar bons resultados, é possível ver uma diagonal da esquerda para a direita na matriz de confusão, essa diagonal são os atributos classificados corretamente. Caso a matriz esteja confusa e não possui uma diagonal, isso significa que o modelo de classificação não produz bons resultados.

Na Figura 14, é apresentada uma matriz de confusão que foi gerada a partir de um conjunto de dados de letras do alfabeto. Então foi gerado um modelo de classificação baseado em árvores de decisão e gerado a matriz de confusão com a quantidade de acertos e erros do modelo.

Figura 14 - Matriz de Confusão.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	<-- classified as
757	1	0	4	0	1	2	1	0	0	0	5	3	0	3	2	1	1	1	2	2	0	0	1	2	0	a = A
0	644	2	13	5	5	5	12	4	4	3	1	1	6	2	4	1	21	9	0	1	12	3	7	0	1	b = B
1	0	648	0	15	9	22	4	1	0	5	2	0	1	4	2	2	3	6	4	4	0	1	0	0	2	c = C
0	18	1	703	0	2	3	16	1	2	4	0	1	12	6	3	5	13	5	0	2	0	0	4	1	3	d = D
0	7	10	1	666	2	20	2	1	2	8	6	0	0	0	1	7	1	9	2	0	2	0	10	1	10	e = E
0	9	4	4	2	661	3	4	10	6	3	1	3	1	1	33	0	0	4	9	1	4	1	3	8	0	f = F
0	12	15	7	19	2	650	6	2	2	3	2	3	0	11	2	9	8	8	1	1	3	1	0	2	4	g = G
0	20	2	31	3	4	8	571	0	4	28	0	2	3	14	6	2	18	3	0	5	2	1	6	1	0	h = H
2	4	2	3	1	2	1	1	700	22	1	2	0	0	0	2	4	0	4	1	0	0	0	1	0	2	i = I
1	11	0	5	3	4	4	6	24	658	1	3	0	1	4	3	2	1	5	2	3	0	0	3	1	2	j = J
1	4	4	3	9	2	2	21	1	0	643	2	1	5	1	0	1	17	3	0	3	1	1	12	1	1	k = K
2	1	2	1	13	0	10	1	0	2	3	702	0	2	0	1	6	4	0	0	1	0	0	9	0	1	l = L
2	2	1	1	1	1	3	6	0	1	1	2	731	11	2	2	2	1	0	0	10	0	7	2	3	0	m = M
0	4	2	9	0	3	0	7	0	6	5	3	10	705	5	1	0	8	0	0	9	2	2	2	0	0	n = N
3	7	7	19	2	0	11	8	0	6	2	1	5	6	632	6	17	5	2	0	6	1	6	1	0	0	o = O
2	3	0	4	4	30	8	5	10	7	0	0	1	0	2	698	2	2	3	3	1	2	4	2	7	3	p = P
2	3	5	8	8	0	11	6	2	5	1	5	0	0	28	4	670	7	3	2	2	2	0	1	3	5	q = Q
1	22	1	12	4	5	8	22	1	4	15	6	1	9	8	1	2	629	2	1	0	2	1	1	0	0	r = R
1	21	3	6	11	8	10	6	7	6	2	2	1	0	4	1	0	3	638	2	0	0	1	0	2	13	s = S
0	5	2	6	2	8	0	3	3	3	3	1	1	0	1	2	5	3	3	718	1	4	0	5	14	3	t = T
3	3	9	3	0	1	4	9	1	2	2	2	7	12	7	3	1	1	3	1	733	4	1	0	1	0	u = U
1	12	2	1	1	6	0	8	0	0	0	2	3	4	5	5	1	2	0	4	2	683	10	0	12	0	v = V
1	2	1	0	0	1	3	1	0	0	0	1	13	7	6	0	1	0	1	1	5	7	696	0	5	0	w = W
1	3	2	7	13	2	1	5	1	5	10	10	0	0	2	2	1	1	7	6	0	1	0	701	2	4	x = X
1	2	3	2	2	8	2	3	4	2	0	0	0	1	1	3	7	1	2	15	3	12	3	1	705	3	y = Y
1	2	2	4	9	5	2	4	5	2	0	5	2	0	2	0	8	4	13	5	0	0	0	4	1	654	z = Z

Fonte: Autoria Própria.

Pode-se observar na Figura 14 que a matriz de confusão possui uma diagonal da esquerda para direita bem definida, com valores bem grandes, isso significa que o modelo de classificação conseguiu classificar uma boa parte dos dados analisados. Ainda que alguns dados foram classificados errôneos, o modelo de classificação apresenta ótimos resultados. Vale ressaltar que os dados que estão fora da diagonal, significa que o caractere em análise foi classificado como outro caractere.

Dado que uma matriz de confusão apresenta a quantidade de classificações corretas e incorretas, pode-se dizer que uma matriz de confusão tem por objetivo mostrar a frequência de classificação para uma das classes do modelo. Segundo Souza (2019) uma matriz de confusão apresenta as seguintes frequências:

- Verdadeiro positivo (*true positive* - TP do inglês): é quando a classe foi classificada corretamente;
- Falso positivo (*false positive* - FP do inglês): é quando a classe foi classificada incorretamente;
- Falso verdadeiro (*true negative* - TN do inglês): é quando uma classe que não estava sendo analisada, acaba sendo classificada corretamente;
- Falso negativo (*false negative* - FN do inglês): é quando uma classe que não estava sendo analisada, acaba sendo classificada incorretamente.

A seguir será apresentado alguns métodos e fórmulas que podem ser utilizados para validar o modelo de classificação a partir da matriz de confusão, segundo os conceitos de (SOUZA, 2019).

2.5.1.2 Precisão

A precisão é utilizada para definir a relação entre as classes que é corretamente classificada como verdadeiro positivo e todas as previsões de verdadeiros positivos (incluindo os falsos positivos) (SOUZA, 2019). Ou seja, daquelas classes que foram classificadas como corretas, quantas efetivamente eram corretas? pode-se obter o resultado dessa questão pela equação de precisão:

$$Precisão = \frac{Verdadeiros\ Positivos\ (TP)}{Verdadeiros\ Positivos\ (TP) + Falsos\ Positivos\ (FP)} \quad (4).$$

Na Figura 15 é possível visualizar na terceira coluna nomeada por (*precision* do inglês) a precisão obtida de cada uma das classes do conjunto de dados de letras do alfabeto.

Figura 15 - Precisão detalhada por classe.

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,959	0,001	0,967	0,959	0,963	0,962	0,984	0,940	A
	0,841	0,009	0,783	0,841	0,811	0,804	0,933	0,753	B
	0,880	0,004	0,888	0,880	0,884	0,880	0,950	0,825	C
	0,873	0,008	0,820	0,873	0,846	0,840	0,947	0,788	D
	0,867	0,007	0,840	0,867	0,853	0,847	0,949	0,803	E
	0,853	0,006	0,856	0,853	0,855	0,849	0,947	0,797	F
	0,841	0,007	0,820	0,841	0,830	0,823	0,937	0,775	G
	0,778	0,009	0,774	0,778	0,776	0,767	0,927	0,706	H
	0,927	0,004	0,900	0,927	0,913	0,910	0,971	0,893	I
	0,881	0,005	0,876	0,881	0,879	0,874	0,953	0,854	J
	0,870	0,005	0,865	0,870	0,868	0,863	0,955	0,818	K
	0,922	0,003	0,916	0,922	0,919	0,916	0,969	0,899	L
	0,923	0,003	0,926	0,923	0,925	0,922	0,968	0,911	M
	0,900	0,004	0,897	0,900	0,899	0,895	0,959	0,855	N
	0,839	0,006	0,842	0,839	0,840	0,834	0,947	0,761	O
	0,869	0,005	0,887	0,869	0,878	0,873	0,953	0,822	P
	0,856	0,005	0,885	0,856	0,870	0,865	0,944	0,804	Q
	0,830	0,006	0,834	0,830	0,832	0,825	0,928	0,727	R
	0,853	0,005	0,869	0,853	0,861	0,856	0,949	0,803	S
	0,902	0,003	0,922	0,902	0,912	0,908	0,961	0,864	T
	0,902	0,003	0,922	0,902	0,912	0,908	0,966	0,880	U
	0,894	0,003	0,918	0,894	0,906	0,902	0,961	0,842	V
	0,926	0,002	0,942	0,926	0,934	0,931	0,968	0,884	W
	0,891	0,004	0,903	0,891	0,897	0,893	0,962	0,842	X
	0,897	0,003	0,913	0,897	0,905	0,901	0,955	0,857	Y
	0,891	0,003	0,920	0,891	0,905	0,902	0,962	0,859	Z
Weighted Avg.	0,880	0,005	0,881	0,880	0,880	0,875	0,954	0,830	

Fonte: Autoria própria.

2.5.1.3 *Recall*

O *recall* ou revocação é a frequência em que o classificador classifica uma classe corretamente, ou seja, quando uma classe que é classificada como verdadeiro positivo, o quão frequente a classe é realmente classificada como verdadeiro positivo (SOUZA, 2019). A frequência pode ser obtida por meio da equação *Recall*.

$$Recall = \frac{Verdadeiros\ Positivos\ (TP)}{Verdadeiros\ Positivos\ (TP)\ +\ Falsos\ Negativos\ (FN)} \quad (5),$$

o *recall* pode ser observado na quarta coluna da Figura 15, para cada uma das classes analisadas.

2.5.1.4 *F1-Score*

O *F1-Score* combina a precisão com o *recall* com o objetivo de obter a qualidade geral do modelo de classificação (SOUZA, 2019). E pode ser obtido por meio da equação *F1-Score*:

$$F1 = \frac{2 * precisão * recall}{precisão + recall} \quad (6).$$

O *F1-Score* pode ser observado na quinta coluna nomeada por *F-Measure* da Figura 15. Indicando a qualidade do modelo de classificação para cada uma das classes.

2.5.2 Validação do Agrupamento

Aplicados os algoritmos de agrupamento nos dados em estudo, o resultado são os agrupamentos, após esse processo é importante mensurar a qualidade de cada grupo. Nas subseções é apresentado alguns métodos de validação.

2.5.2.1 Coesão

Kunz e Black (1995) define coesão como a mensuração da similaridade entre os elementos de um mesmo grupo, sendo que, quanto maior a similaridade maior é a coesão do grupo. A coesão pode ser obtida pela equação a seguir:

$$Coesão = \frac{\sum_{i>i} Sim(Pi, Pj)}{\frac{n(n-1)}{2}} \quad (7),$$

sendo que *Sim (Pi,Pj)* calcula a similaridade entre os elementos *i* e *j* que pertence a um agrupamento *P*, *n* é o número de elementos de *P*, e *Pi* e *Pj* são membros do agrupamento *P*.

2.5.2.2 Acoplamento

Segundo Kunz e Black (1995), o acoplamento faz a mensuração da similaridade média de todos os pares de elementos, onde um elemento do par pertence a um agrupamento X e o outro par pertence a outro agrupamento Y. O acoplamento pode ser obtido pela seguinte equação:

$$Acoplamento = \frac{\sum_{i>i} Sim(Ci, Cj)}{\frac{na(na-1)}{2}}$$

(8),

em que *C* é o centróide de um determinado agrupamento que está contido em *P*, *Sim*(*Ci, Cj*) é o cálculo da similaridade do elemento *i* que pertence ao agrupamento *P* e o elemento *j* que pertence a um outro agrupamento *Pi*, *Ci* é o centróide do agrupamento *P*, e *Cj* é o centróide do agrupamento *Pi* e *na* é o número de agrupamentos presente em *P*.

2.5.2.2 Coeficiente de Silhouette

Alves (2018) explica o coeficiente de Silhouette em seu trabalho, como um método que identifica o quão bem um ponto se encaixa em um *cluster*. Quando o coeficiente está próximo de 1 positivo, significa que os pontos estão longe dos pontos de um outro *cluster*, e quando o coeficiente está próximo de 0, indica que os pontos estão muito perto ou até interseccionando com um outro *cluster*.

Para Zoubi e Rawi (2008) o coeficiente de *Silhouette* é baseado na ideia de quanto um elemento é similar aos demais elementos do seu grupo, e o quão distante esse mesmo elemento está dos elementos de um outro grupo. Combinando as medidas de coesão e acoplamento por meio da equação 9.

$$S = \frac{b(i) - a(i)}{\max\big(a(i), b(i)\big)}$$

(9),

na qual, *a(i)* é a distância média entre o *i-ésimo* elemento do grupo e todos os outros elementos do mesmo grupo. O *b(i)* é a menor distância entre o *i-ésimo* elemento do grupo e qualquer outro

grupo, que não contém o elemento, e *max* é a maior distância entre $a(i)$ e $b(i)$ (ZOUBI E RAWI, 2008).

Delgado *et al*. (2013) diz que o coeficiente de *Silhouette* de um grupo, nada mais é que o cálculo da média aritmética para cada elemento que pertence ao grupo. O mesmo pode ser obtido pela Equação 9, em que o valor de *S* fica na faixa de 0 e 1.

2.6 VISUALIZAÇÃO

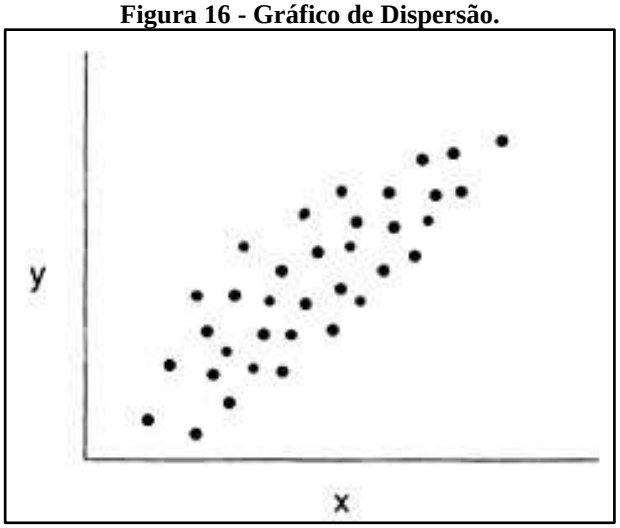
No conceito de mineração de dados, a palavra visualização de dados se refere ao processo de transformar os padrões encontrados em informação visual. Botelho (2002) afirma que a apresentação visual possibilita ao analista de dados perceber características que estão escondidas nos dados, mas que são necessárias para tarefas de exploração e análise.

Nas subseções abaixo serão apresentadas as principais técnicas de visualização dos resultados obtidos após a etapa de agrupamento e classificação dos dados.

2.6.1 Matriz ou Gráfico de Dispersão

Gráfico de dispersão também conhecido como gráfico de correlação ou gráfico XY, é uma forma de representar graficamente uma possível relação entre duas variáveis, apresentando os pares de dados numéricos e a relação entre si (FORLOGIC, 2016). Os dados são exibidos em forma de uma coleção de pontos, cada atributo representa seu valor em um plano de eixo X e Y. Ilustrado na Figura 16 e Figura 7 processo de agrupamento de dados.

Figura 16 - Gráfico de Dispersão.



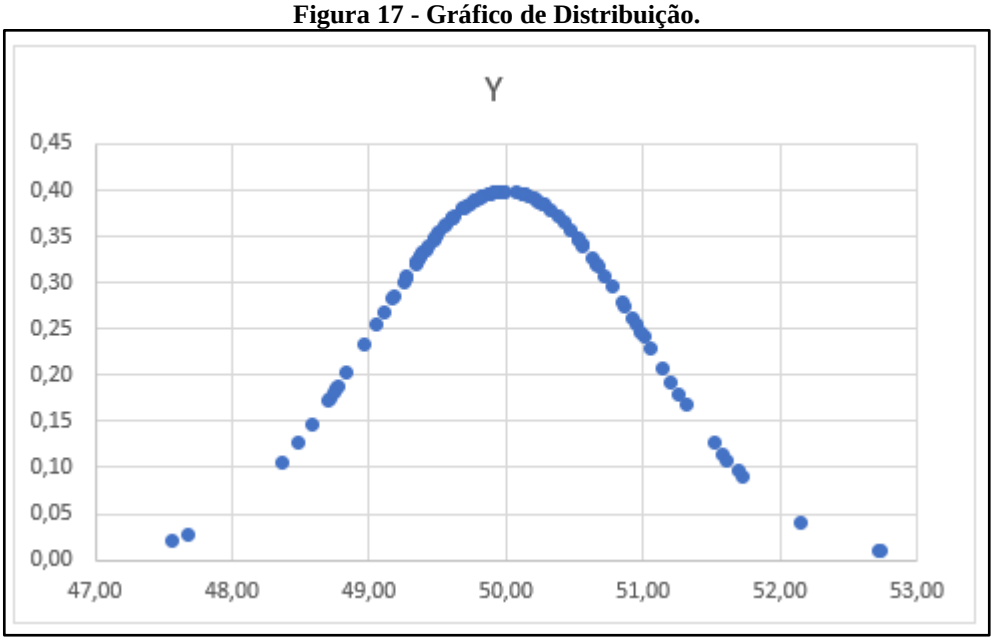
Fonte: Guilherme (2008).

O gráfico de dispersão ilustrado na figura 16, apresenta um conjunto de dados dispersos em um plano de eixo X e Y. Pode-se observar uma relação entre x e y, quanto maior a variável x maior é a variável y. Este gráfico pode ser utilizado para analisar as relações entre os atributos de uma determinada classe e pode apresentar a distribuição dos dados de um agrupamento.

2.6.2 Gráfico de Distribuição

Um gráfico de distribuição apresenta uma coleção de dados distribuídos em um eixo X e Y destacando o ponto da variável com maior valor em Y. Assim os demais pontos são distribuídos na esquerda e direita do maior valor de Y, formando duas caldas (ZIBETTI, 2019), conforme ilustrado na Figura 17.

Figura 17 - Gráfico de Distribuição.



Fonte: Excel e Accees (2020).

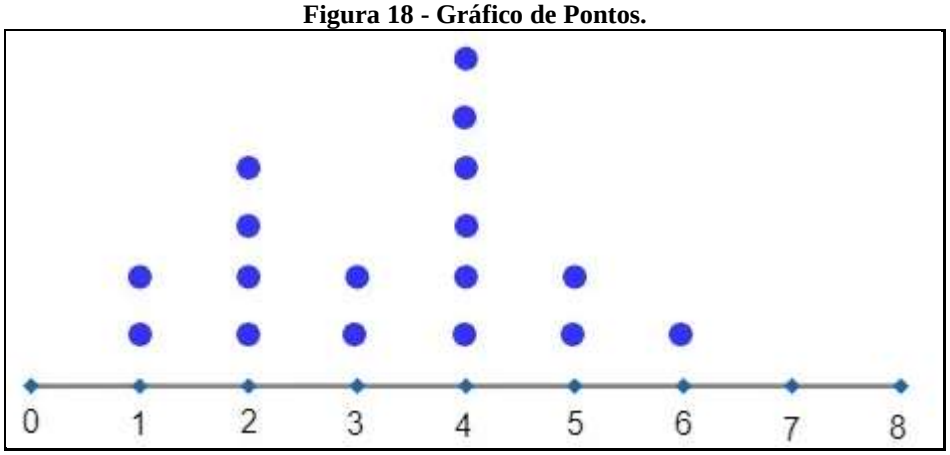
Esse gráfico é útil quando se quer observar a distribuição dos dados em relação à média e desvios padrão, geralmente utilizado na técnica *Z-Score*.

2.6.3 Gráfico de pontos

Um gráfico de pontos é uma maneira de representar informações por meio de uma reta numérica (LUDERITZ, 2021). Para apresentar informações por meio deste gráfico, coloca-se

um ponto em cima do número de cada conjunto de dados, caso o dado apareça mais que uma vez, coloca os demais pontos acima dos anteriores. Ilustrado na Figura 18.

Figura 18 - Gráfico de Pontos.



Fonte: Luiz (2021)

O gráfico de pontos, pode ser utilizado para contar a quantidade de ocorrências de um determinado dado em um conjunto, facilitando ao analista a realizar cálculos estatísticos.

2.6.4 Matriz de Confusão

A matriz de confusão foi apresentada na subseção 2.5.1.1 como um método de validação, porém esse tipo de matriz também pode ser utilizado como um dos meios de visualização dos resultados do modelo de classificação.

2.7 DADOS CENSITÁRIOS

A plataforma de Sistema IBGE de Recuperação Automática - SIDRA, que compila os resultados dos Censos Agropecuário e Demográfico é uma das mais complexas operações estatísticas realizadas por um país, tornando ainda mais complexa em países que possui dimensões continentais como o Brasil, com um território de 8.515.692,27 km², composto por 27 unidades de federação e 5.565 municípios e com uma população de 190.755.799 habitantes (IBGE, 2021).

Segundo o IBGE (2021) os dados censitários são a principal forma de se obter informações sobre a situação de vida da população de um país. Não só o governo, mas a sociedade civil se beneficia das informações advindas dos dados censitários do país, para realizarem suas escolhas com base em informações atualizadas sobre a população.

O IBGE (2021) explica que os censos demográficos tem como objetivo mensurar a quantidade de habitantes de todo o território nacional, identificar suas características, apresentar o meio de vida dos brasileiros, produzindo informações imprescindíveis para a definição de políticas públicas e a tomada de decisões de investimentos da iniciativa privada ou de governo.

2.7.1 Documentação Operacional e Principais Variáveis

Segundo informações do IBGE (2021), a documentação operacional se dá por meio dos seguintes questionários:

1. Questionário básico (sem amostra) – aplicado em todas as unidades domiciliares, exceto naquelas selecionadas para a amostra. Contém perguntas sobre as características básicas do domicílio e seus moradores;
2. Questionário de amostra – aplicado em todas as unidades domiciliares selecionadas para a amostra. Além das perguntas do questionário básico, contém outras mais detalhadas a respeito do domicílio e seus moradores;
3. Folha de coleta – utilizada para o registro das unidades residenciais e não residenciais existentes no setor e para o registrado número de moradores em cada domicílio ocupado, além de servir para a seleção dos domicílios particulares nos quais se aplicou o questionário de amostra;
4. Folha de domicílio coletivo – utilizada para o registro das pessoas recenseadas em cada domicílio coletivo, além de servir para a seleção das unidades nas quais se aplicou o questionário da amostra;
5. Caderneta do setor – utilizada para registro de resumo das informações coletadas, além de conter o mapa e a descrição dos limites do setor.

E as principais variáveis são:

1. Situação urbana e rural;
2. Características do domicílio;
3. Sexo, Idade, Cor ou Raça, Etnia ou Povo a que pertence e Língua falada só para indígenas, Religião ou Culto, Registro de Nascimento, Deficiência Física ou Mental, Educação, Deslocamento para estudo, Nupcialidade;
4. Características do Trabalho e do Rendimento, Deslocamento para trabalho e Fecundidade e Mortalidade;
5. Emigração internacional, Migração interna e Imigração internacional.

A pesquisa sempre é feita no segundo semestre do ano em questão, e os resultados são divulgados em aproximadamente 3 anos após a pesquisa. A disseminação é feita por meio de publicação impressa, CD-ROM, portal do IBGE, BME (Banco Multidimensional de Estatísticas) e SIDRA (IBGE, 2021).

2.8 ESTADO DA ARTE

No desenvolvimento desta seção, foi feito um levantamento de trabalhos que faz o uso de técnicas de mineração de dados em cima de dados censitários, buscando na literatura quais as ferramentas mais utilizadas no processo de mineração.

Esta seção está dividida em duas subseções, na primeira é apresentado trabalhos que utilizam técnicas de mineração de dados e que a fonte dos dados é o Sistema IBGE de Recuperação Automática-SIDRA. A segunda seção trata sobre trabalhos correlatos que aplicam técnicas de mineração de dados e utilizam bases de dados censitárias como fonte de dados.

Para encontrar os trabalhos, utilizou-se dos recursos do Google Acadêmico. A pesquisa foi realizada com base nos seguintes termos: “Mineração de Dados” ou “*Data Mining*” +” censo” ou” IBGE” +” Agropecuária”. O objetivo de realizar a consulta com esses termos, foi encontrar trabalhos que podem estar correlacionados com os objetivos deste trabalho.

2.8.1 Trabalhos Correlatos que possuem a Base SIDRA como Fonte de Dados

Costa *et al*. (2017) e Brito *et al*. (2015), propuseram o uso de redes bayesianas e algoritmos de busca heurística como o K2, para analisar os dados extraídos do SIDRA. Com objetivo de medir a associação entre o trabalho infantil e gravidez na adolescência e a associação entre a posse de tecnologias de informação com a gravidez na adolescência, tomando por referência os municípios da Amazonia legal em relação ao demais municípios do país.

Já Silvano et. al. (2020), propuseram o uso de clusters e árvores de decisão para realizar mineração nos dados da base SIDRA. Com o objetivo de analisar a distribuição espacial dos grupos de municípios brasileiros e a similaridade entre os conjuntos de indicadores demográficos, sociais e econômicos do país.

Santos *et al*. (2016) também fizeram um trabalho com os dados agropecuários da base SIDRA, com o objetivo de identificar os estratos de área de produção familiar rural e de produção patronal. Por mais que os autores descrevem o uso de técnicas de mineração, nenhum

algoritmo de classificação em específico foi utilizado, propondo somente um sistema desktop para apresentar os dados de forma mais amigável.

Sistemas de análise socioeconômica nacionais, abertos ao público, são carentes de funcionalidades de busca ou representação de informações aos usuários (Vieira Filho, 2013). Com isso o autor propôs um *framework* para auxiliar e agilizar o desenvolvimento de sistemas de consulta de dados socioeconômicos. Desenvolvendo métodos genéricos para extração de dados de bases como o SIDRA e o sistema DevInfo, a partir de funções mineradoras e a elaboração de uma interface para desenvolvimento de softwares a partir do *framework* proposto.

2.8.2 Trabalhos Correlatos que Aplicam Técnicas de Mineração de Dados em Bases de Dados Censitárias

Silva Filho *et al*. (2013) em seu trabalho apresentam o uso de uma técnica de mineração de dados utilizando a regra de associação *Apriori*, por ser uma das técnicas mais utilizadas, quando o assunto é regras de associação, segundo pesquisas do autor. Os autores aplicam o algoritmo em uma base de dados do IBGE com informações de natalidade do estado de São Paulo no ano de 2000. Com objetivo de encontrar correlações entre a idade da mãe e o tipo de gravidez.

Enquanto Maria das Graças *et al*. (2010), aplicaram técnicas de *data mining* sobre os dados do processo de produção e mecanização de cana de açúcar. Com o objetivo de encontrar características semelhantes entre os produtores de cana-de-açúcar. Os autores utilizaram métodos de clusterização para identificar grupos de produtores similares e alguns principais métodos de classificadores propostos pela literatura, como: árvores de decisão, classificadores Bayesianos, K-NN, redes neurais e *support vector machine* (SMO).

Já Fasiaben *et al*. (2003), apresentam o uso de técnicas de mineração de dados, sobre os dados de produção e rendimentos do banco de dados de Produção Anual Municipal (PAM) do IBGE, para a definição de alvos prioritários de pesquisa baseado na produção de arroz, feijão, milho, soja e trigo. Os autores propõem a utilização do algoritmo K-médias para realizar agrupamentos e o algoritmos de árvores de decisão para realizar a classificação dos grupos, aplicando os resultados sobre um mapa do Brasil, onde é possível identificar a porcentagem de produção e rendimentos nos estados brasileiros.

Árvores de decisão, especificamente o algoritmo J48 do WEKA, foi utilizado para classificar municípios com aptidão agrícola, na região do Matopiba. O algoritmo foi aplicado

sobre os dados socioeconômicos e físicos dos municípios, obtidos das bases de dados do IBGE (Lorensini *et al*. 2018).

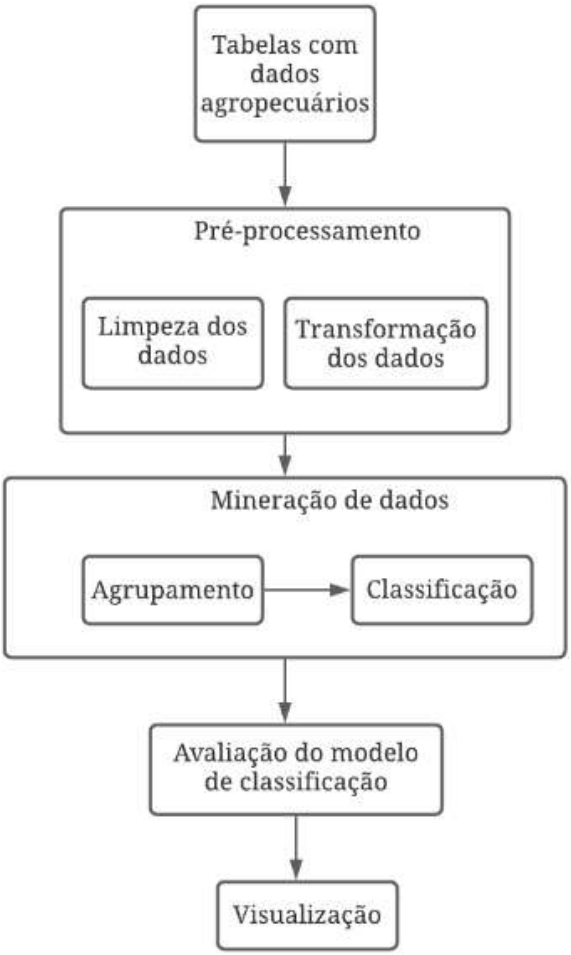
Macedo Junior (2009) com o objetivo de verificar se existe alguma associação entre os dados de produtividade do café e os dados meteorológicos, no estado de São Paulo, propôs o uso de algoritmos Apriori, Predictive Apriori e Tertius da ferramenta de mineração WEKA. Com aplicação dos algoritmos os autores conseguiram encontrar algumas associações, tais como, com o aumento da precipitação aumenta a produtividade do café.

A pesquisa por trabalhos foi realizada, no entanto não foram encontrados muitos trabalhos que fizessem uso de dados agropecuários de bases censitárias. Porém, alguns trabalhos apresentaram métodos bem vantajosos, como o uso de algoritmos de clusterização para fazer os agrupamentos dos dados trabalhados, e o uso de algoritmos de classificação como as árvores de decisão para encontrar alguma correlação entre os dados agrupados. Com isso faz-se importante o uso dessas técnicas para análise dos dados agropecuários, estudados neste trabalho.

3 MÉTODO

Neste capítulo será abordado o método para o desenvolvimento do *framework* de mineração de dados. Em um contexto geral, o fluxo de execução segue as seguintes etapas: captura dos dados agropecuários, pré-processamento, agrupamento, classificação, validação e visualização dos padrões encontrados. O processo é ilustrado na Figura 19.

Figura 19 - Fluxo da execução do *framework*.

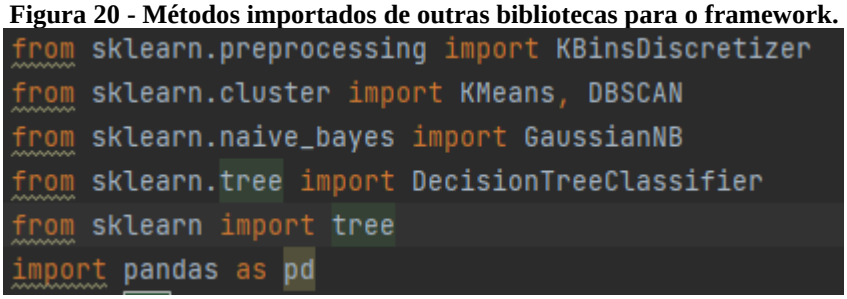


Fonte: Autoria Própria.

Para toda a codificação do *framework* foi utilizado a linguagem de programação Python, uma vez que há vasta quantidade de bibliotecas disponíveis para trabalhar com mineração de dados e a simplicidade e praticidade de escrever um código nesta linguagem, além de integrar sistemas de forma eficaz (PYTHON, 2021).

Também foram utilizadas as bibliotecas de aprendizado de máquina *scikit-learn* e pandas. As duas bibliotecas são de código aberto e voltadas para a linguagem de programação Python, possuem vários algoritmos de aprendizagem de máquina, tais como: classificação, clusterização, agrupamento, pré-processamento, além de outros algoritmos (SCIKIT-LEARN; PYDATA, 2021). A figura 20 apresenta alguns dos métodos que foram importados do *Scikit-learn*.

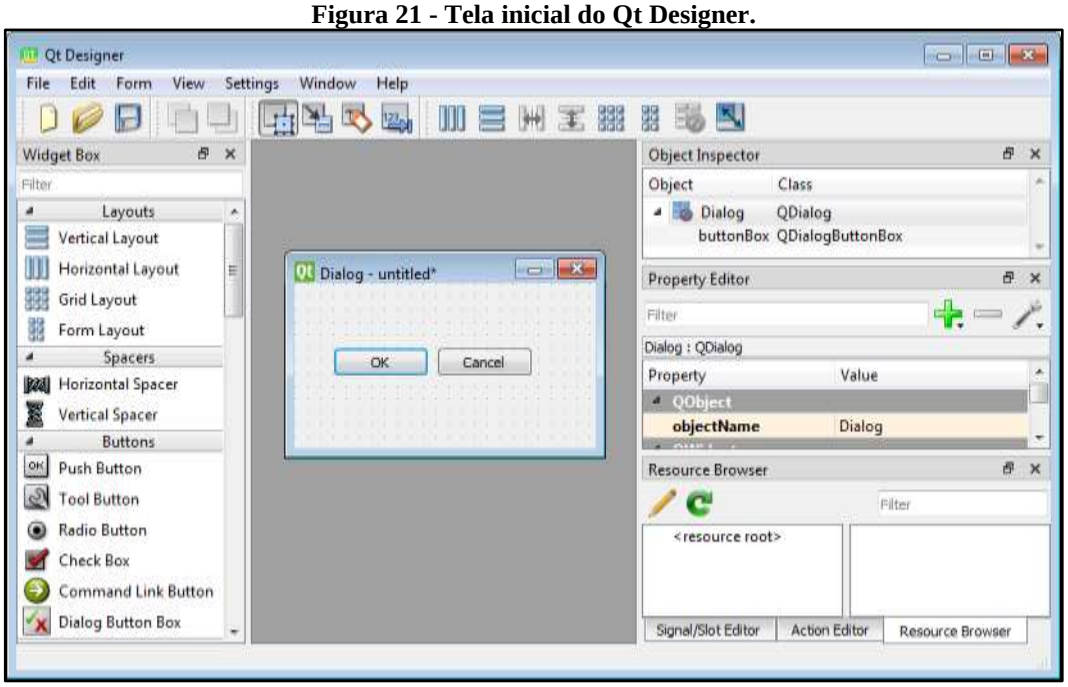
Figura 20 - Métodos importados de outras bibliotecas para o framework.



Fonte: Autoria própria.

Também foi utilizado o ambiente de desenvolvimento integrado IDE (*Integrated Development Environment*) Qt Designer para a construção das interfaces gráficas. O Qt Designer oferece uma interface simples de clicar, arrastar e soltar objetos como botões, campos de texto, caixas de combinação e demais objetos. Ao final é possível gerar um arquivo XML, figura 22, com as especificações da interface, no qual pode ser importado para dentro de um projeto escrito em Python, por meio de uma biblioteca de ligação PyQt (HERRMANN, 2021). A Figura 21 apresenta a tela inicial da ferramenta.

Figura 21 - Tela inicial do Qt Designer.



Fonte: Autoria própria.

Todas as telas do *framework* foram desenhadas no PyQt5, gerados os arquivos XML figura 22 e importado para dentro do projeto por meio da biblioteca de ligação PyQt5 uic, ilustrado na figura 23.

Figura 22 - Arquivo gerado pelo PyQt5.

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
 <class>MainWindow</class>
 <widget class="QMainWindow" name="MainWindow">
  <property name="geometry">
   <rect>
    <x>0</x>
    <y>0</y>
    <width>726</width>
    <height>560</height>
   </rect>
  </property>
  <property name="windowTitle">
   <string>MainWindow</string>
  </property>
  <widget class="QWidget" name="centralwidget">
   <widget class="QPushButton" name="pushButton_project_path">
    <property name="geometry">
     <rect>
      <x>160</x>
      <y>120</y>
      <width>89</width>
      <height>25</height>
     </rect>
    </property>
    <property name="text">
     <string>Projeto</string>
    </property>
   </widget>
```

Fonte: Autoria própria.

Figura 23 - Integrando o arquivo XML ao Python.

```
class CreateProject(QMainWindow):
    '''Cria novos projetos'''
    # Método construtor
    def __init__(self):
        super().__init__()
        self.main_path = PATH
        uic.loadUi('create_project.ui', self)
```

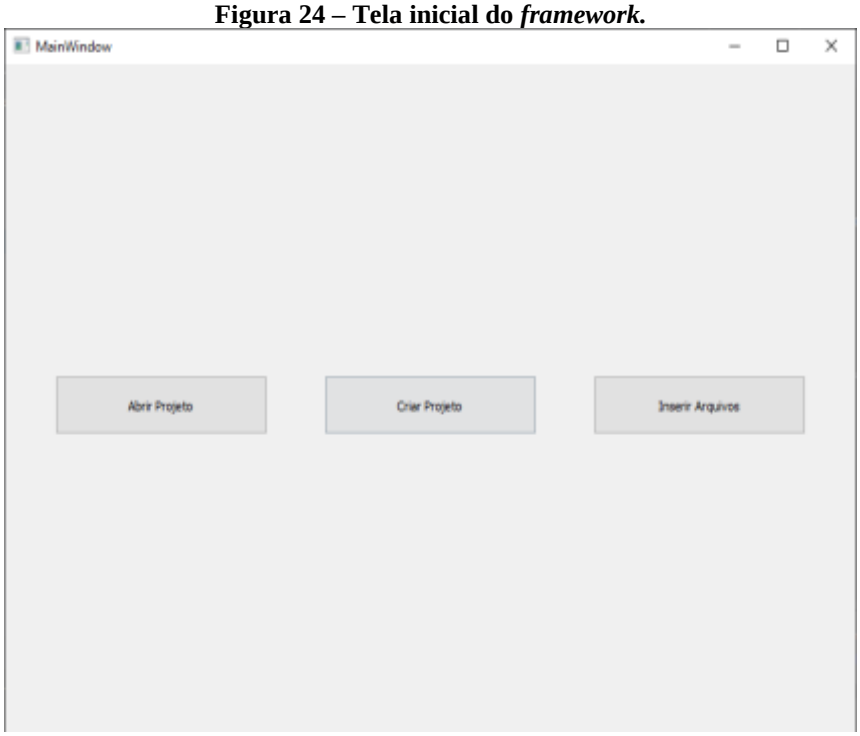
Fonte: Autoria própria.

3.1 IMPLEMENTAÇÃO DA TELA INICIAL

A tela inicial é responsável pela criação de um novo projeto, pela captura dos dados e abertura do projeto (Figura 24). Nesta tela foram implementados três botões, cada um é responsável por abrir uma outra tela de funcionalidades. O botão criar projeto abre uma janela para selecionar o local onde deve ser criado o projeto, então uma nova pasta é criada no endereço especificado com o nome que foi definido na janela figura 25.

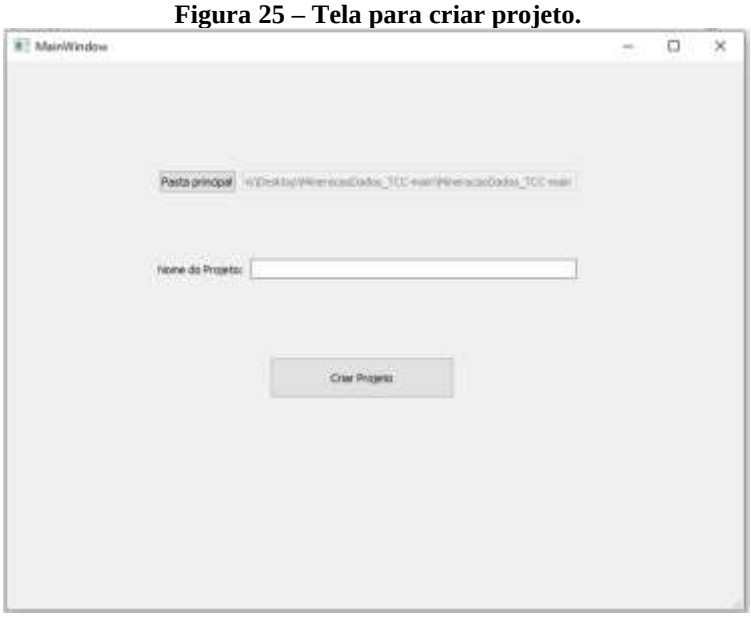
O botão abrir projeto da tela principal irá abrir uma janela do sistema operacional para selecionar a pasta do projeto, então, é aberto a tela de processamento de dados onde é possível realizar todo o pré-processamento dos dados.

Figura 24 – Tela inicial do *framework*.



Fonte: Autoria própria.

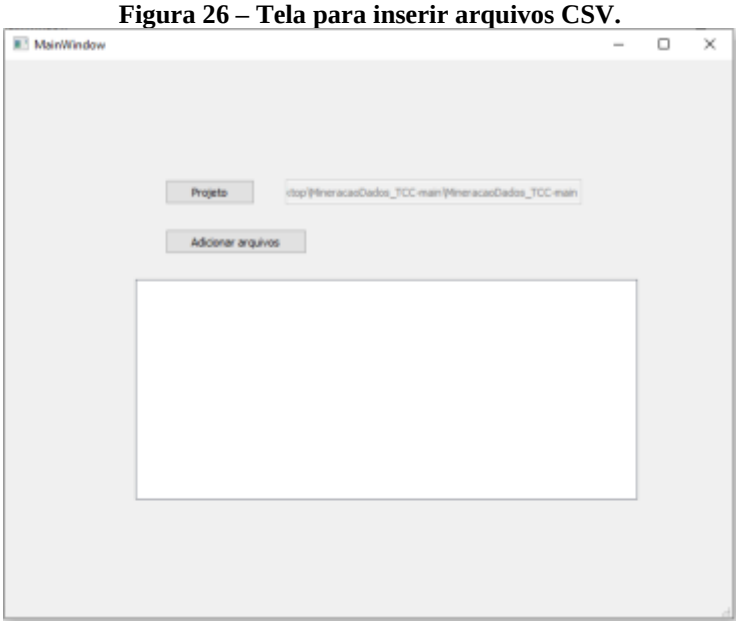
Figura 25 – Tela para criar projeto.



Fonte: Autoria própria.

Para captura dos dados, foi implementado uma tela para inserção de arquivos no projeto, a qual é renderizada a partir do clique do botão “Inserir Arquivos”, figura 26. É uma limitação do *framework* realizar somente a leitura de arquivos CSV separado por ponto e vírgula, ao realizar o download do arquivo na base do SIDRA, o usuário deve sempre optar por arquivo CSV BR. O *framework* possui uma tela de pré-processamento de dados em que é possível excluir linhas e colunas, mas para evitar problemas na leitura do arquivo, é recomendado excluir informações que não pertencem ao conjunto de dados estudado, como por exemplo, informações sobre fonte e descrições adicionais da tabela.

Figura 26 – Tela para inserir arquivos CSV.

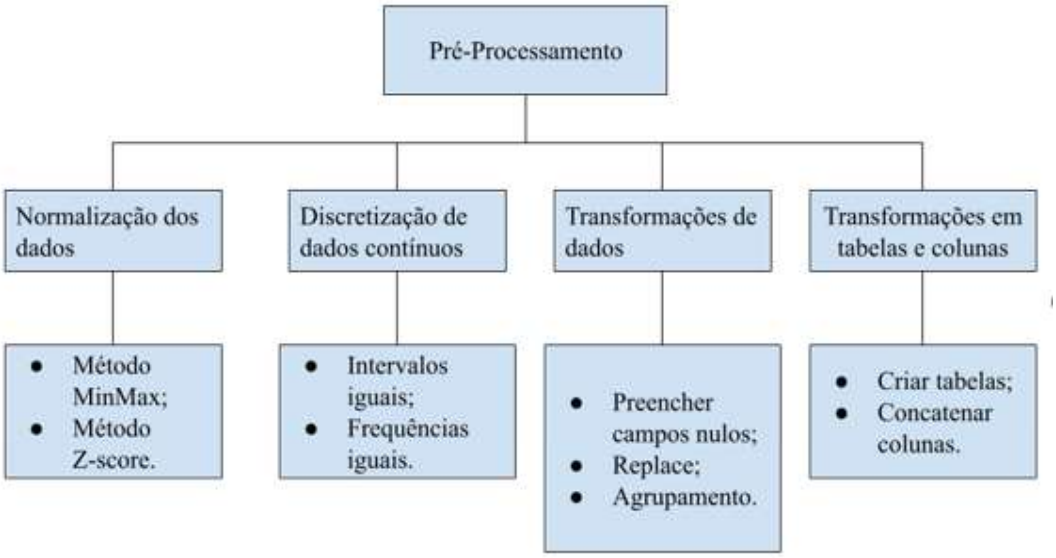


Fonte: Autoria própria.

3.2 IMPLEMENTAÇÃO DO MÓDULO DE PRÉ-PROCESSAMENTO

O módulo de pré-processamento faz a leitura dos arquivos CSV que foram adicionados na tela inicial e fica disponível para o usuário realizar as manipulações necessárias. Este módulo permite realizar normalização de dados por meio dos métodos MinMax e Z-score, discretizar os dados contínuos por intervalos e frequências iguais, realizar transformações nos dados e nas tabelas. A figura 27 ilustra as possibilidades de manipulação, vale ressaltar que não precisa seguir a ordem da imagem, o usuário pode realizar as operações que achar necessárias.

Figura 27 - Fluxo do pré-processamento.

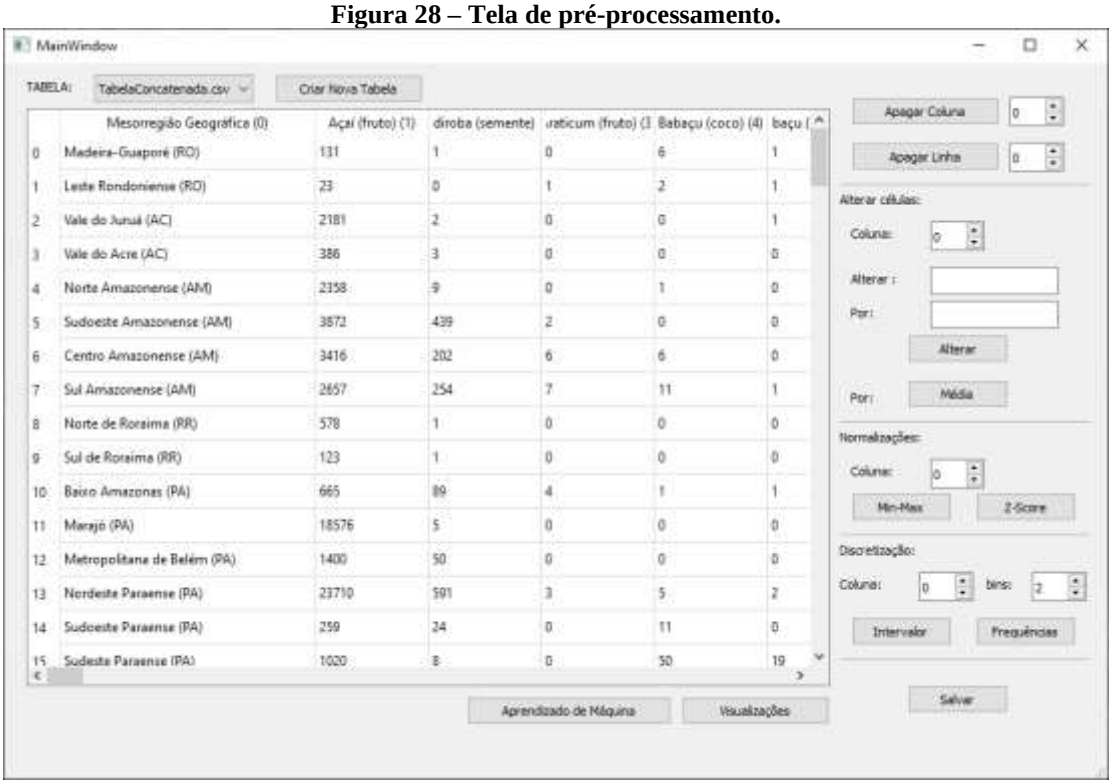


Fonte: Autoria própria.

Para o módulo de pré-processamento, foi implementado outra tela que possui as funcionalidades descritas neste trabalho, na qual se faz necessário para realizar a pré-processamento dos dados (Figura 28). Tais funcionalidades são:

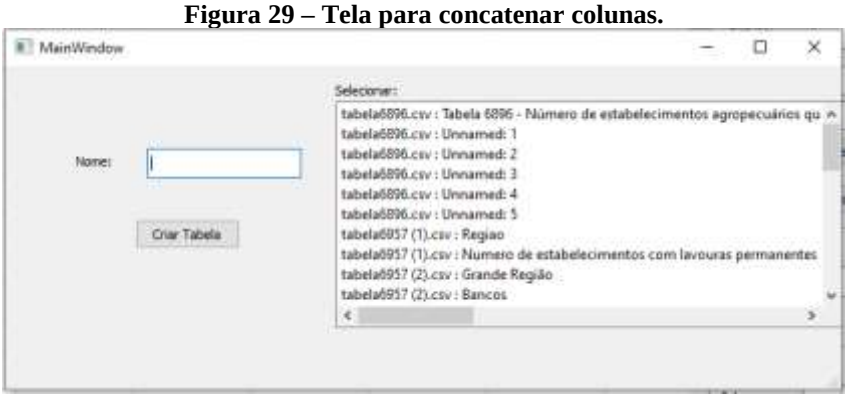
- Selecionar qual a tabela o usuário deseja realizar o pré-processamento;
- Criar uma nova tabela a partir da concatenação de colunas das tabelas inseridas no *framework* (Figura 29);
- Apresentar os dados da tabela que está sendo processada e alterar uma determinada célula da tabela;
- Especificar e deletar uma coluna ou linha da tabela;
- Realizar um *replace* em todos os dados de uma coluna por um valor especificado ou pela média do conjunto;
- Normalizar os dados pelos métodos Min-Max e Z-Score;
- Discretizar os dados por intervalos ou por frequências;
- Navegar para o módulo de mineração e visualização.

Figura 28 – Tela de pré-processamento.



Fonte: Autoria própria.

Figura 29 – Tela para concatenar colunas.



Fonte: Autoria própria.

Os métodos de normalização e discretização dos dados foi importado da biblioteca *sk-learn* e implementado no *framework*, em que é passado um conjunto de dados por parâmetro e os métodos retornam o mesmo conjunto normalizado. Ilustrado na figura 30.

Figura 30 – Método Z-score do sk-learn.

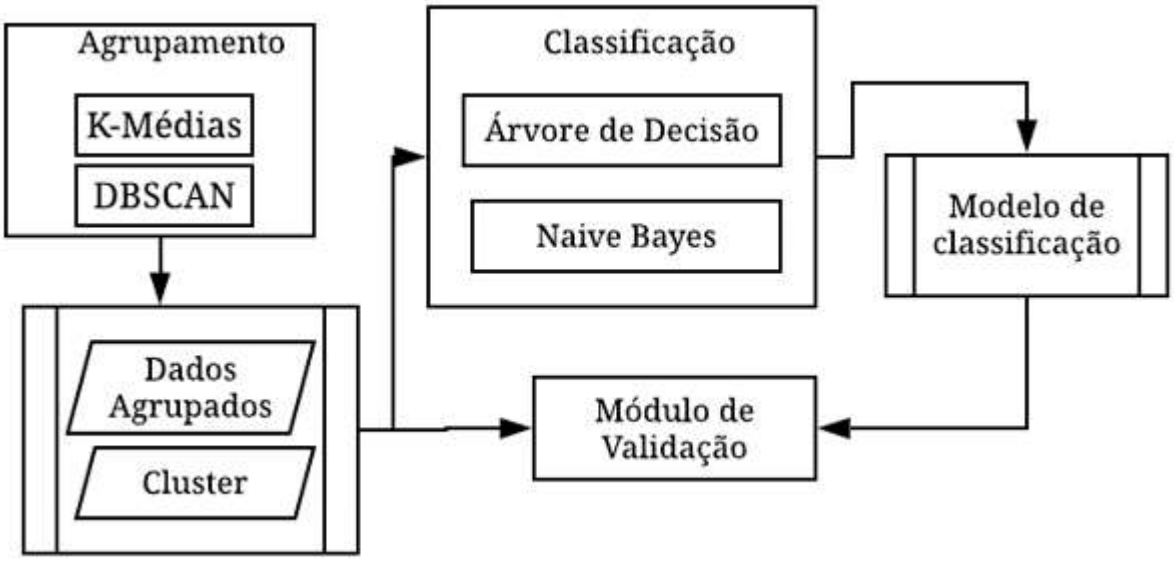
```
#Normalização Z-Score
def replaceZscore(self, column_idx):
    self.beginResetModel()
    selected_column = pd.to_numeric(self._data[self._data.columns[column_idx]], errors='coerce').values
    self._data[self._data.columns[column_idx]] = stats.zscore(selected_column)
    self.endResetModel()
```

Fonte: Autoria Própria.

3.3 IMPLEMENTAÇÃO DO MÓDULO DE MINERAÇÃO DOS DADOS

O módulo de mineração de dados tem como objetivo agrupar e/ou classificar os dados pré-processados, fazendo o uso de dois algoritmos de agrupamento: K-médias e DBScan, já para classificação de *Naive-Bayes* e Árvore de decisão. A Figura 31 apresenta o fluxo do processo deste módulo.

Figura 31 - Fluxo do processo de mineração.

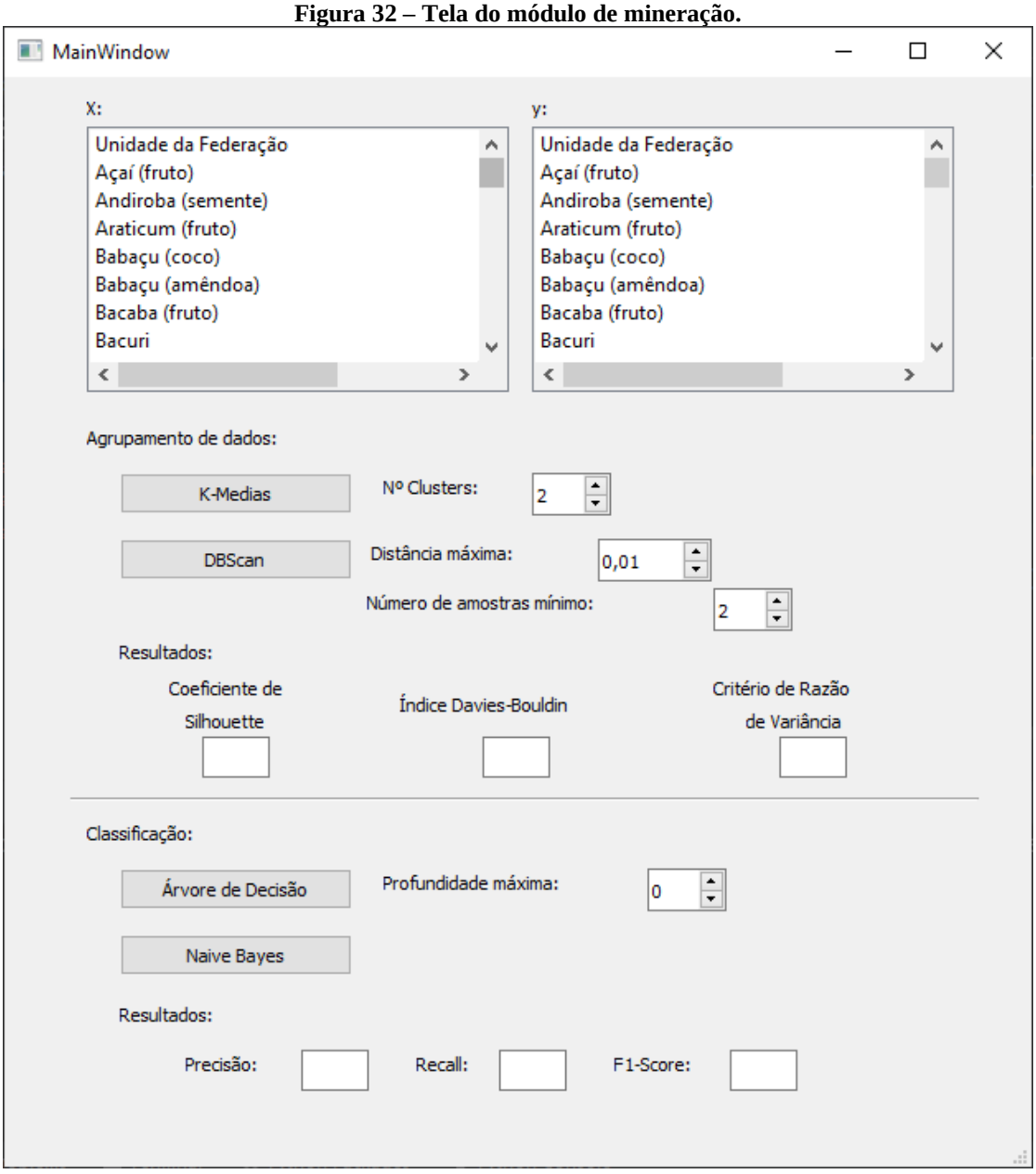


Fonte: Autoria Própria.

Foi implementado uma tela para o módulo de mineração de dados, figura 32, que possui as funcionalidades necessárias para agrupar, classificar e validar os resultados. A tela possui as seguintes funcionalidades:

- Selecionar os atributos a serem agrupados ou classificados;
- Agrupar os dados pelo algoritmo K-Médias ou DBScan passando os parâmetros necessários para realizar o agrupamento;
- Validar o agrupamento e apresentar os resultados;
- Classificar os dados pelo algoritmo *Naive bayes* ou Árvore de decisão;
- Validar a classificação e apresentar os resultados.

Figura 32 – Tela do módulo de mineração.



Fonte: Autoria Própria.

3.4 IMPLEMENTAÇÃO DO MÓDULO DE ANÁLISE E VISUALIZAÇÃO DOS DADOS

O último módulo desenvolvido é o de visualização dos resultados. Por meio da tela de visualização é plotado os gráficos apresentando os padrões encontrados pelo algoritmo de classificação e os grupos gerados pelo agrupamento. Os gráficos disponibilizados no *framework* são: gráfico de dispersão, gráfico de distribuição, gráfico de pontos e matriz de confusão. Para o desenvolvimento deste módulo, foi utilizado a biblioteca matplotlib, uma biblioteca desenvolvida para a linguagem de programação Python, que possui diversos algoritmos para a criação de visualizações estáticas, animadas e interativas (MATPLOTLIB, 2021).

Assim, para ilustrar a operacionalização da ferramenta, foram realizados dois estudos de caso, com dois grupos distintos de variáveis, a fim de verificar como a ferramenta se comporta com a disponibilidade distinta de informações. Para tanto, o primeiro grupo de dados refere-se a atividades produtivas que não são encontradas em todas as regiões do Brasil. O segundo grupo compreende informações de atividades presentes em todo o território nacional. No próximo capítulo será apresentado o estudo de caso.

4 ESTUDO DE CASO A PARTIR DE DADOS DO CENSO AGROPECUÁRIO BRASILEIRO

Desenvolvido o *framework* de mineração de dados, é necessário avaliar a sua operacionalidade e viabilidade de aplicação. Com este intuito foi elaborado um estudo de caso sobre um conjunto de dados do Censo Agropecuário 2017, composto de dados de extração vegetal, horticultura e fruticultura e/ou plantas ornamentais. A escolha dessas informações se justifica no interesse em verificar como a ferramenta se comporta com atividades que possam ser encontradas em todas as Unidades da Federação, bem como aquelas que possuem maior relação com determinadas regiões e biomas, consequentemente sendo ausente em algumas Unidades da Federação.

As informações foram extraídas da plataforma SIDRA (descrita na seção 2.7), na qual compila os resultados do Censo Agropecuário brasileiro. Cabe esclarecer que o Censo Agropecuário tem como período de referência o intervalo entre os dias 1º de outubro de 2016 a 30 de setembro de 2017, e como data de referência para coleta dos dados, o dia 30 de setembro de 2017 (IBGE, 2017). Assim, para a este estudo, foram selecionadas três tabelas com as seguintes variáveis:

a) Tabela 6950 - Número de estabelecimentos agropecuários com produtos da extração vegetal: as variáveis selecionadas compreendem o número de estabelecimentos agropecuários com produtos de extração vegetal. Sobre o total de produtores, assinalando todos os produtos de extração vegetal para as 27 Unidades da Federação.

b) Tabela 6954 - Número de estabelecimentos agropecuários com horticultura: as variáveis selecionadas são número de estabelecimentos agropecuários com horticultura, totalidade de produtos da horticultura para as 27 Unidades da Federação.

c) Tabela 6952 - Número de estabelecimentos agropecuários com produção de floricultura e/ou plantas ornamentais: as variáveis selecionadas são número de estabelecimentos agropecuários com produção de floricultura e/ou plantas ornamentais, a totalidade de produtos da floricultura para as 27 Unidades da Federação.

O primeiro passo foi selecionar os dados que serão inseridos no *framework*, neste estudo de caso definiu-se estudar as informações produtivas de horticultura, extração vegetal e floricultura das 27 Unidades da Federação. Essas informações foram extraídas diretamente na plataforma SIDRA e sumarizadas no Quadro 2.

Quadro 2 - Sistematização dos dados selecionados na plataforma Sidra.

Tabela	Extração Vegetal	Horticultura	Floricultura e/ou plantas ornamentais
Principal variável	Número de estabelecimentos agropecuários com produtos da extração vegetal.	Número de estabelecimentos agropecuários com horticultura.	Número de estabelecimentos agropecuários com produção de floricultura e/ou plantas ornamentais.
Atividades produtivas e culturas	Açaí (fruto); Andiroba (semente); Araticum (fruto); Babaçu (coco); Babaçu (amêndoa); Bacaba (fruto); Bacuri; Baru (amêndoa); Borracha (látex líquido); Borracha (látex coagulado); Buriti (coco); Buriti (palha); Butiá (fibra); Cacau (amêndoa); Cagaita (fruto); Cajarana; Camu-camu (fruto); Carnaúba (cera); Carnaúba (pó de palha); Casca de angico; Castanha-do-Brasil (castanha-do-Pará); Caucho (goma elástica); Copaíba (óleo); Cumaru (semente); Cupuaçu; Erva-mate; Ipecacuanha (raiz); Jaborandi (folha); Jambu (folha); Juçara (fruto); Lenha; Licuri (coquilho); Licuri (cera); Maçaranduba (goma não elástica); Macaúba (fruto); Mangaba (fruto); Maniçoba (goma elástica); Madeira em toras para papel; Madeira em toras outra finalidade; Murici; Murumuru (semente); Palmito; Oiticica (semente); Pequi; Piaçava (fibra); Pinhão; Pupunha (coco); Sorva (goma não elástica); Ucuuba (amêndoa); Imbú ou umbú; Outros produtos; Tucumã.	Abobrinha; Acelga; Agrião; Aipo; Alcachofra; Alcaparra; Alecrim; Alface; Alho-porró; Almeirão; Aspargo; Batata-baroa (mandioquinha); Batata-doce; Berinjela; Bertalha; Beterraba; Boldo; Brócolis; Bucha (esponja vegetal); Camomila; Cará; Caruru; Cebolinha; Cenoura; Chicória; Chuchu; Coentro; Cogumelos; Couve; Couve-flor; Erva-doce; Ervilha (vagem); Espinafre; Gengibre; Hortelã; Inhame; Jiló; Lentilha; Manjericão; Maxixe; Milho verde (espiga); Morango; Mostarda (semente); Nabiça; Nabo; Orégano; Pepino; Pimenta; Pimentão; Quiabo; Rabanete; Repolho; Rúcula; Salsa; Taioba; Tomate (estaqueado); Vagem (feijão vagem); Outros produtos; Sementes (produzidas para plantio); Mudas e outras formas de propagação (produzidas para plantio).	Flores e folhagens para corte; Gramas; Plantas ornamentais em vaso; Mudas de plantas ornamentais; Plantas, flores, folhagens medicinais; Sementes (produzidas para plantio); Mudas e outras formas de propagação (produzidas para plantio).

Fonte: Autoria própria.

O quadro acima nos permite observar as variáveis principais e as respectivas categorias indicadas na coleta dos dados pelo Censo. A extração vegetal compreende produtos colhidos ou obtidos no período de referência proveniente de espécies vegetais não plantadas (nativas), contemplando áreas com mato ralo, caatinga ou cerrado, utilizadas ou não para o pastejo de animais (IBGE, 2017). Ainda segundo o Manual, para os dados de horticultura são considerados produtos aqueles provenientes do cultivo em hortas, de verduras e legumes. Fruticultura compreende a presença de área plantada ou destinada ao plantio de flores ou áreas ocupadas

com viveiros de mudas, estufa para produção de plantas, flores ou casas de vegetação (local para experimentos em condições controladas).

Por fim, é importante destacar que a utilização do *framework* não se limita a dados do censo agropecuário brasileiro, e podem se adaptar facilmente a outros contextos, que não serão abordados neste trabalho. Além disso, o foco da ferramenta é disponibilizar informações para possibilitar análises e decisões futuras, não sendo da alçada deste trabalho análises e estudos aprofundados sobre os dados extraídos, sendo esta tarefa de responsabilidade exclusiva dos usuários da aplicação.

4.1 MÉTODO PARA OS ESTUDOS DE CASO

O processo de seleção dos dados ocorreu por meio de seleção intencional das variáveis, almejando operacionalizar a ferramenta com dados distintos e disponibilidade variada. Os dados selecionados foram submetidos à etapa de captura de dados do *framework*, este processo consiste em carregar as planilhas na aplicação. Com as planilhas carregadas é possível manipular as informações, o primeiro passo é concatenar as informações contidas nas planilhas de forma individual. Este processo é preciso para incorporar as variáveis contidas nas planilhas em suas respectivas instâncias nas unidades da federação, para aumentar o volume de dados a serem processados pelo *framework* de mineração de dados.

Em seguida, os dados passam por transformações no módulo de pré-processamento, retirando campos sem descrições ou com descrição desnecessária para o processo de mineração de dados, aplicando os métodos de remoção ou preenchimento de células vazias. Esse processo é fundamental para análise das tabelas do Censo Agropecuário, visto que há duas nomenclaturas adotadas para o caso da ausência de informação, quais sejam: uso do traço “-” para informar que não há registro de atividade para a variável selecionada, e uso de “X” para ocultar a identidade do produtor, especialmente se for apenas um responsável pela atividade na unidade geográfica selecionada. Esses valores de “-” e “X” foram substituídos pelo valor 0, devido às características das informações e limitações dos algoritmos de agrupamentos que precisam de valores numéricos para processar os dados.

Com os dados pré-processados, passa para etapa de agrupamento dos dados, para isto foi utilizado dois algoritmos de K-médias e DBSCAN, com intuito de agrupar as unidades federativas com base nas informações extraídas da plataforma SIDRA. Esse agrupamento visa encontrar informações que geralmente não são identificadas por agentes humanos, pois

algoritmos de agrupamentos são eficientes para trabalhar com um elevado número de variáveis, encontrando, frequentemente, relações entre variáveis que não são facilmente identificadas pela percepção humana.

Após o agrupamento dos dados, é possível utilizar algoritmos de classificação para possibilitar *insights* sobre o relacionamento das instâncias presentes no agrupamento. Para este processo, foi utilizado os algoritmos de árvore de decisão que devido a sua formulação permitir interpretar as regras geradas para separar os dados, seguindo esta mesma linha, utilizou o classificador *Naive bayes* que permite rankear as características das instâncias pelas importâncias na classificação dos dados, esta importância é definida pelo classificador baseado na eficácia da característica em prever as instâncias do problema.

As métricas de avaliação dos modelos são importantes para validar as informações disponibilizadas pelos classificados, valores baixos em relação às métricas (Precisão, *Recall*, *F1-Score*) indicam que as informações devem ser desconsideradas e o modelo deve ser ajustado. Por fim, as informações extraídas pelos classificadores são disponibilizadas em formato gráfico para auxiliar no processo de análise das informações pelo usuário.

4.1.1 Geração dos Agrupamentos

Os dados concatenados em única tabela foram aplicados aos algoritmos de agrupamento K-médias e DBScan. O algoritmo DBScan não obteve resultados satisfatórios para os tipos de dados em estudo, pois devido a distância entre os dados, o algoritmo não encontrou valores suficientes para realizar um bom agrupamento, em poucos casos onde as distâncias entre os dados eram parecidas, algumas variáveis foram agrupadas. Os valores de K foram definidos para gerar os clusters, porém definir os valores de K é uma tarefa difícil para ser realizada a priori. Desta forma, o algoritmo K-médias foi executado 9 vezes considerando os valores de K entre 2 e 10.

O *framework* possui três métricas de avaliação de agrupamento, o coeficiente de silhouete, o coeficiente de Davies Bouldin que é baseado em coesão e a razão da variância que é baseado em acoplamento. Para o estudo em questão considerou-se o coeficiente de silhouette, pois essa métrica indica o quão bem uma instância foi agrupada em um determinado grupo. A Tabela 3 apresenta os resultados dos agrupamentos e os 3 melhores valores de K foram 2, 3 e 4.

Tabela 3 - Resultados dos agrupamentos gerados pelo algoritmo K-médias.

Validação do Agrupamento. K-médias			
Nº de Cluster	**Silhouette**	**Davies-Boldin**	**Razão da Variância**
2	**0.78**	**0.61**	**24.22**
3	**0.74**	**0.49**	**19.52**
4	**0.64**	**0.76**	**21.48**
5	0.60	0.75	23.96
6	0.57	0.63	23.51
7	0.57	0.55	25.14
8	0.59	0.50	27.26
9	0.57	0.41	32.64
10	0.54	0.40	36.39

Fonte: Autoria Própria.

Definido os melhores valores de K, executou-se o algoritmo K-médias para cada um dos melhores valores, a cada execução uma nova tabela é criada automaticamente com as instâncias e o grupo a qual elas pertencem. Os resultados dos agrupamentos podem ser visualizados na Tabela 4.

Tabela 4- Agrupamento dos estados.

Unidades da Federação x Agrupamentos							
Estados	**K=2**	**K=3**	**K=4**	**Estados**	**K=2**	**K=3**	**K=4**
Rondônia	0	1	0	Sergipe	0	1	0
Acre	0	1	0	Bahia	1	0	1
Amazonas	1	0	2	Minas Gerais	0	1	2
Roraima	0	1	0	Espírito Santo	0	1	0
Pará	0	2	3	Rio de Janeiro	0	1	0
Amapá	0	1	0	São Paulo	0	1	0
Tocantins	0	1	0	Paraná	0	1	0
Maranhão	0	1	0	Santa Catarina	0	1	0
Piauí	0	1	2	Rio G do Sul	0	1	2
Ceará	1	0	1	Mato G do Sul	0	1	0
Rio G do Norte	0	1	0	Mato Grosso	0	1	0
Paraíba	0	1	2	Goiás	0	1	0
Pernambuco	0	1	0	Distrito Federal	0	1	0
Alagoas	0	1	0				

Fonte: Autoria Própria.

Depois que as instâncias foram agrupadas, o próximo passo é classificar os dados agrupados, com intuito de descobrir possíveis motivações para criação dos agrupamentos a partir das regras e variáveis utilizadas pelos algoritmos de classificação. Os resultados da classificação serão discutidos nas próximas subseções.

4.1.2 Classificação dos Agrupamentos com K igual a 2

Nesta seção é apresentado uma breve análise dos dados gerados pelos métodos de

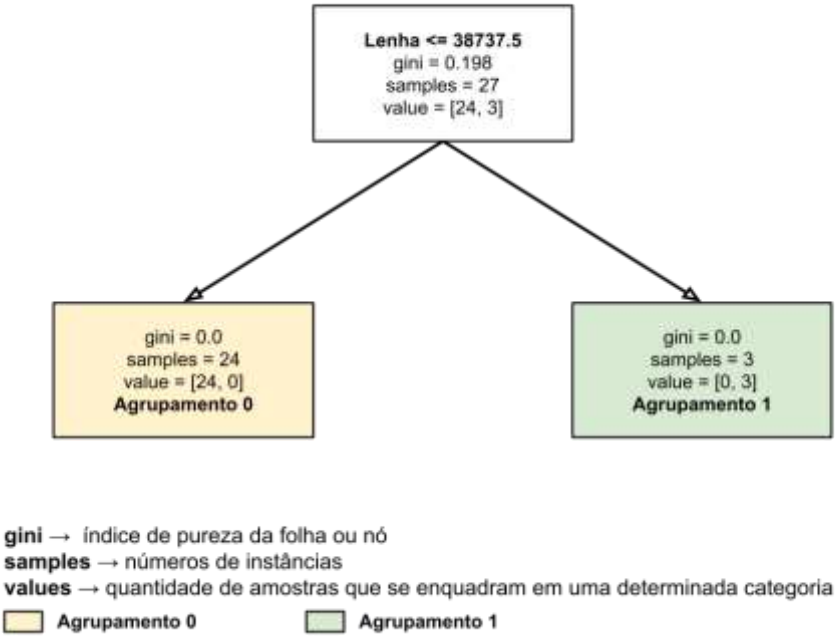
classificação sobre as instâncias agrupadas com o valor de K igual a 2, ou seja, as instâncias divididas em dois agrupamentos. O primeiro fator a ser avaliado é o desempenho desses classificadores, regras de decisão e variáveis provenientes de classificadores com baixo desempenho deve ser desconsideradas. Desta forma, foram extraídas as métricas de avaliação (descritas na seção X) dos classificadores sobres os dados agrupados.

O desempenho do classificador baseado em árvore de decisão conseguiu alcançar a marca de 100% sobre as métricas de precisão, *recall* e *f1-score*, enquanto, o classificador *Naive bayes* alcançou a marca de 80, 96 e 85% para precisão, *recall* e *f1-score*, respectivamente. Em geral, os resultados dos classificadores foram satisfatórios superando a marca de 80% em todas as métricas.

Ao observar a Figura 33 é possível visualizar a árvore de decisão gerada pelo algoritmo de classificação. A variável lenha foi utilizada para definir a regra de separação entre os dois grupamentos, considerando que valores menores ou iguais a 38737,5 pertencem ao agrupamento 0, enquanto instâncias com valores superiores pertencem ao agrupamento 1. Neste experimento, o interessante é o agrupamento 1 que reúne as unidades federativas do Amazonas, Ceará e Bahia, demonstrando serem os maiores extratores de lenha do país.

Um fator que pode ter influenciado nestes agrupamentos e definido a variável lenha como principal ponto de separação entre os grupos, é que a variável lenha é uma das poucas variáveis que possui dados para todas as instâncias, ou seja, todas as unidades da federação realizam a extração de lenha. Esse fator indica que, a qualidade das informações impacta diretamente nos resultados.

Figura 33 - Árvore de decisão com 2 agrupamentos.

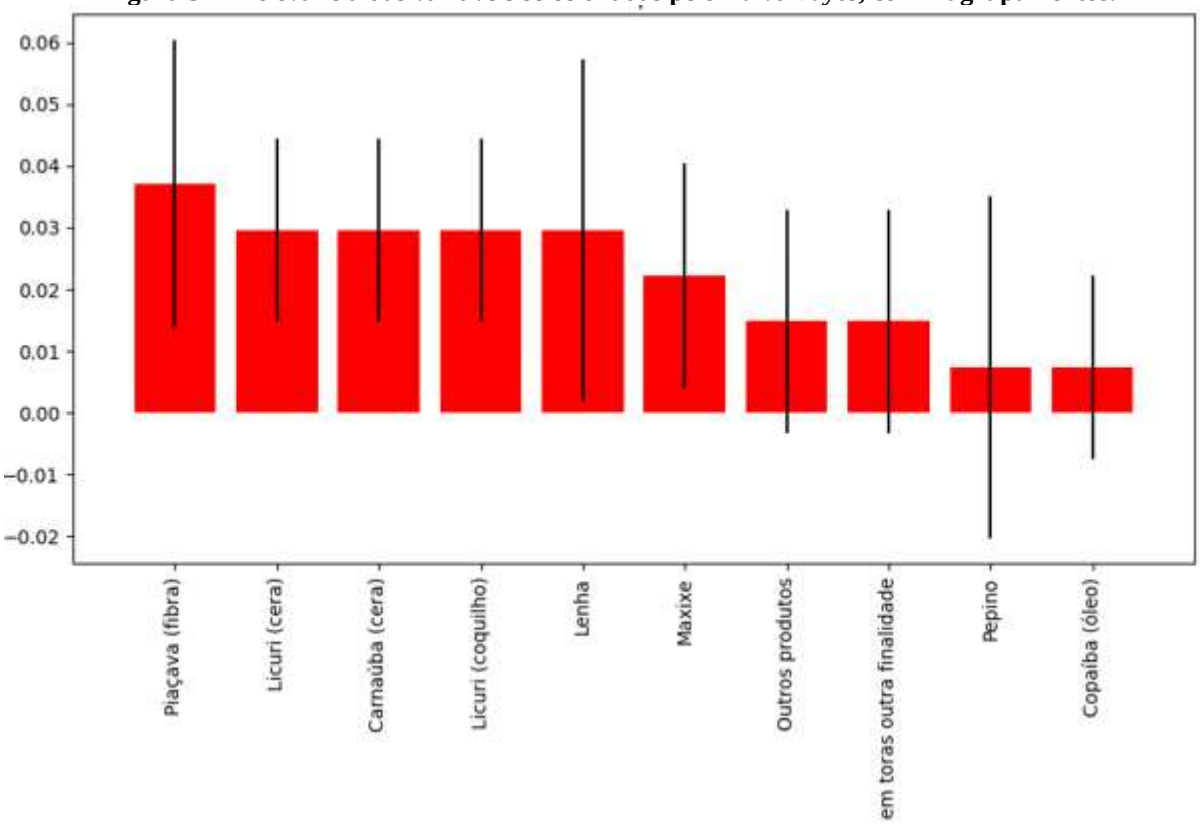


Fonte: Autoria Própria.

A Figura 34 demonstra as dez características mais importantes para classificação dos agrupamentos conforme a técnica de inspeção por permutação[2] das probabilidades extraídas do classificador *Naive bayes*. É importante ressaltar que existe a necessidade de profissional especializado para analisar os dados extraídos. Em geral, utilizando o conhecimento de senso comum, que frutos e outros extratos derivados de diferentes espécies de palmeiras foram relevantes para esses agrupamentos gerados.

[2] https://scikit-learn.org/stable/modules/generated/sklearn.inspection.permutation_importance.html

Figura 34 - Relevância das variáveis selecionadas pelo *Naive Bayes*, com 2 agrupamentos.



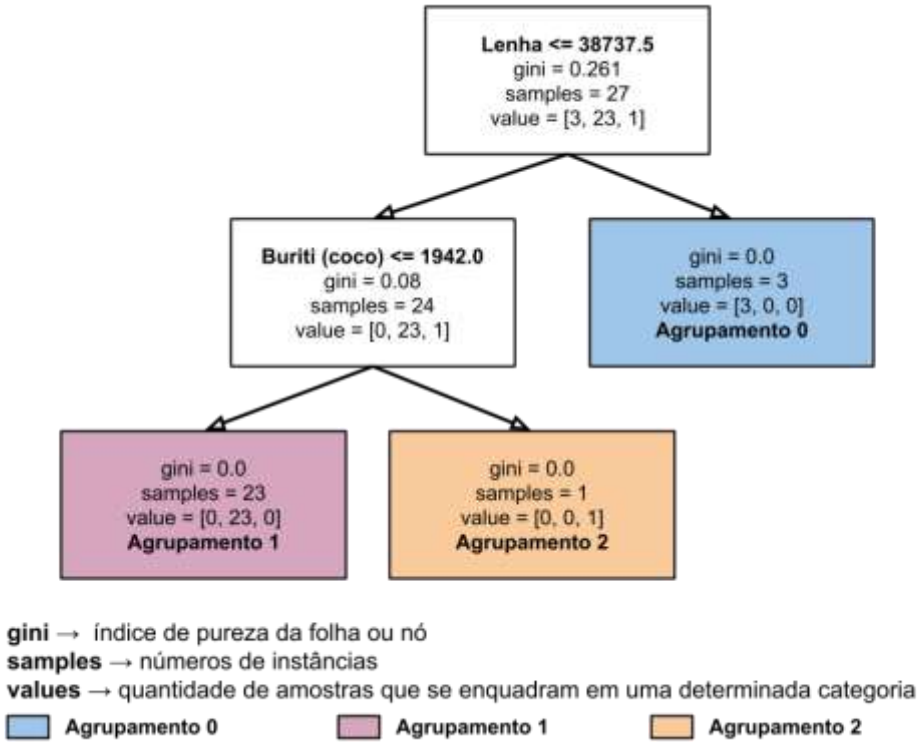
Fonte: Autoria Própria.

4.1.3 Classificação dos Agrupamentos com K igual a 3

Nesta seção é apresentado a análise dos resultados gerados pelos classificadores com valor da média igual a 3. O desempenho do classificador baseado em árvores de decisão novamente conseguiu atingir 100% sobre as métricas de precisão, *recall* e *f1-score*, enquanto o classificador *Naive bayes* atingiu 92, 99 e 94% para precisão, *recall* e *f1-score*, respectivamente. Os resultados dos classificadores com 3 agrupamentos foram satisfatórios superando os 90% em todas as métricas.

Analisando a árvore de decisão na figura 35, novamente a variável lenha foi utilizada para definir a regra de separação entre o agrupamento 0 dos outros dois agrupamentos, considerando que valores menores ou igual 38737.5 pertencem ao agrupamento 0, já a variável Buriti(coco) foi utilizada para separar o agrupamento 1 do agrupamento 2, considerando que valores menores ou igual a 1942 pertence ao agrupamento 1 enquanto instâncias com valores superiores a 38737.5 pertence ao agrupamento 2.

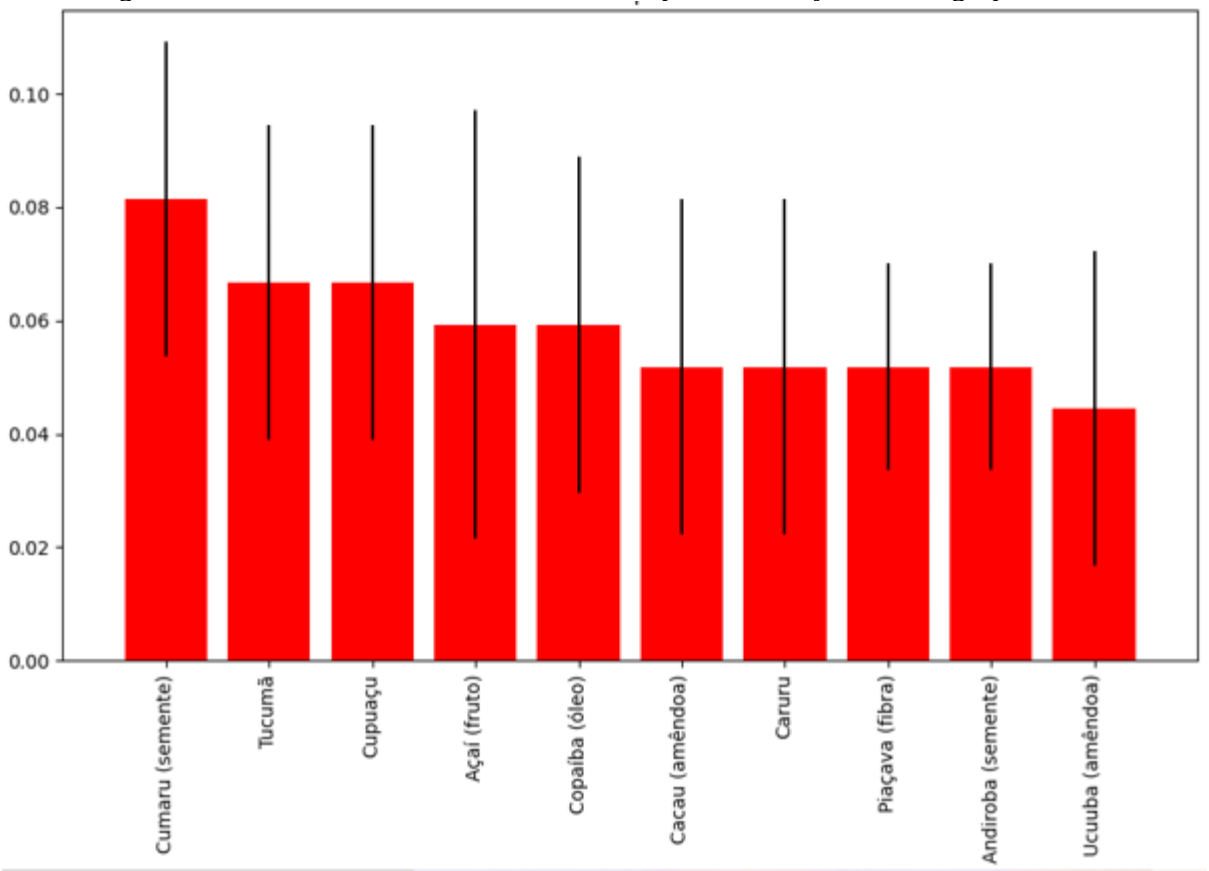
Figura 35- Árvore de decisão com 3 agrupamentos.



Fonte: Autoria Própria.

A figura 36 apresenta as dez características mais importantes para o classificador *Naive bayes*, conforme a técnica de inspeção por permutação. Como explicado na seção anterior a necessidade de um profissional da área para poder tirar conclusões assertivas sobre os resultados obtidos, a nível de classificação as 10 melhores características relevantes para os 3 agrupamentos foram as apresentadas no gráfico.

Figura 36 - Relevância das variáveis selecionadas pelo *Naive Bayes*, com 3 agrupamentos.



Fonte: Autoria Própria.

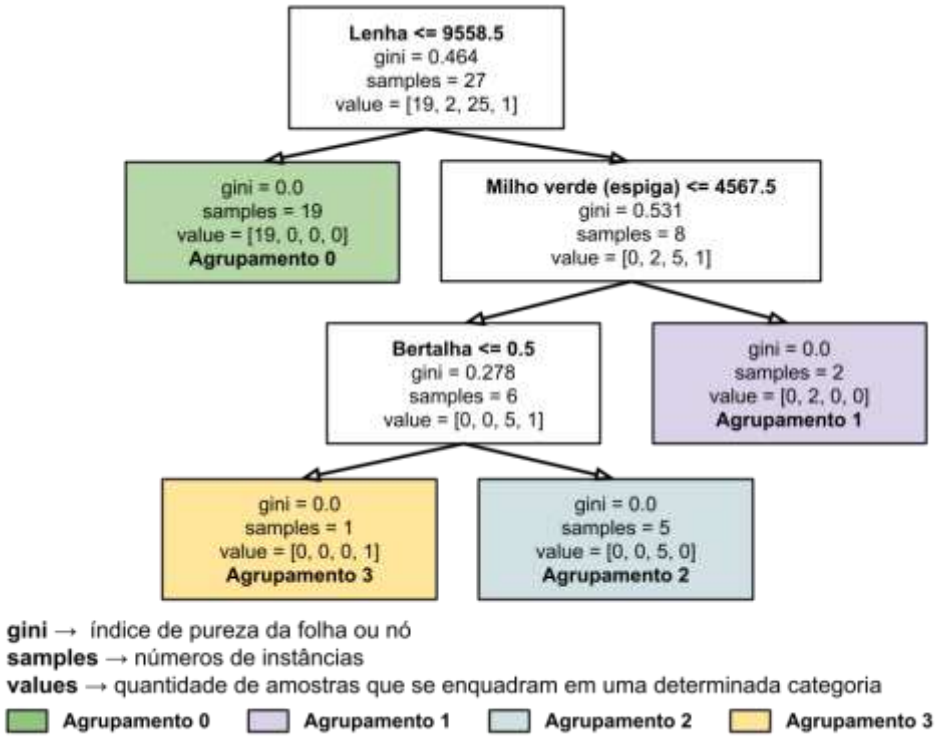
4.1.4 Classificação dos Agrupamentos com K igual a 4

Esta seção apresenta os resultados obtidos dos classificadores com uma quantidade de agrupamentos igual a 4. Analisando o desempenho de cada classificador, obteve para árvore de decisão 100% sobre as métricas de precisão, *recall* e *f1-score*. Já para o classificador *Naive bayes* obteve um resultado de 99, 95 e 97% sobre as métricas de precisão, *recall* e *f1-score* respectivamente.

Observando a árvore de decisão na figura 37, a variável lenha ainda e o fator principal para dividir o agrupamento 0 dos demais, e para todo valor menor ou igual a 9558.5 é classificado como pertencente ao grupo 0. Já a variável Milho verde (espiga) foi utilizada para separar o agrupamento 1 dos demais e para todo valor menor que 4567.5 é considerado como uma variável pertencente ao agrupamento 1. E por último a variável Bertalha foi utilizada para separar o agrupamento 2 do agrupamento 3 e com um valor de 0.5 sendo que toda variável menor que 0.5 é classificada como pertencente ao agrupamento 2 e as demais variáveis com

valores superiores a apresentada na árvore de decisão é classificada como pertencente ao agrupamento 3.

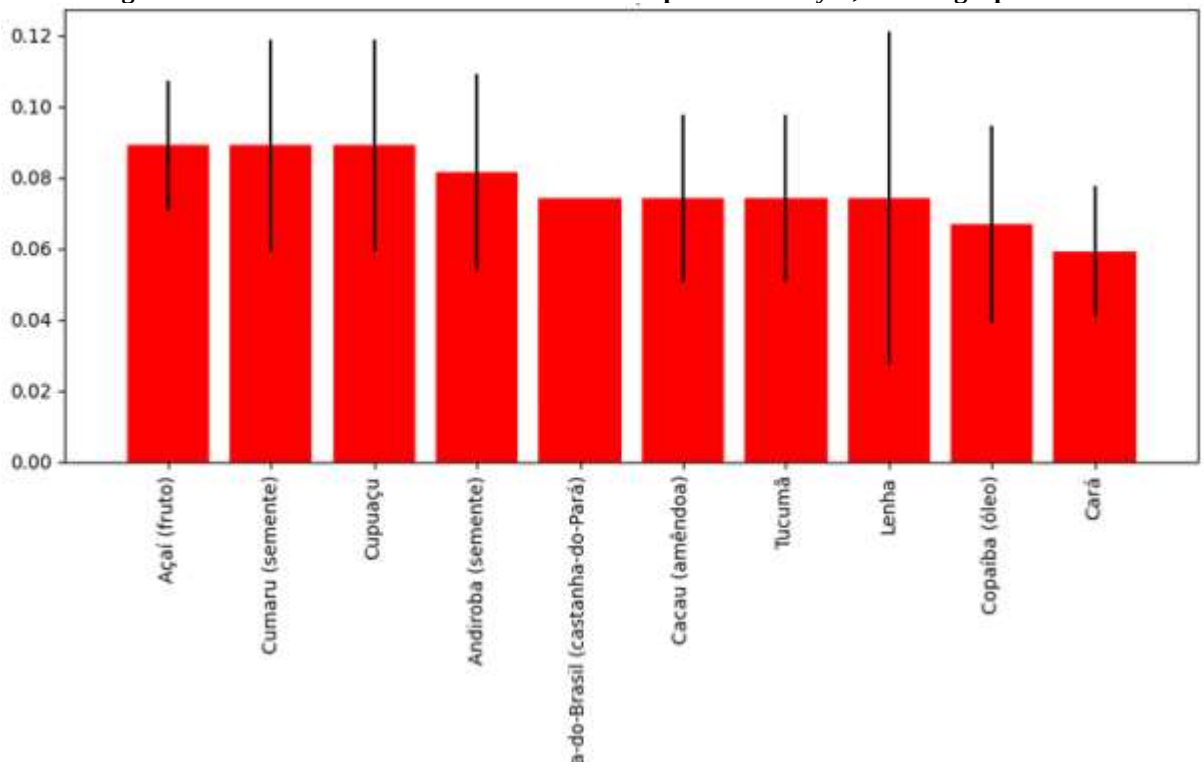
Figura 37 - Árvore de decisão com 4 agrupamentos.



Fonte: Autoria Própria.

Na Figura 38 apresenta as dez características que o classificador *Naive Bayes* considerou que é mais relevantes ou possui maiores valores na tabela de probabilidades segundo a técnica de inspeção por permutação.

Figura 38 - Relevância das variáveis selecionadas pelo *Naive Bayes*, com 4 agrupamentos.



Fonte: Autoria Própria.

Pode-se observar que o algoritmo *Naive Bayes* para K = 2 e K = 4 teve a variável lenha como uma das dez variáveis com maior relevância para a classificação, assim como os resultados da árvore de decisão para K = 2, 3 e 4 respectivamente. Visto que a variável lenha possui valores para todas as instâncias, ou seja, todas as unidades da federação realizam a extração de lenha, pode concluir que variáveis que possui uma maior quantidade de dados e com valores maiores que as demais, tende a ser melhor classificadas pelos algoritmos.

No quadro 2 é apresentado todas as variáveis que foram utilizadas no estudo de caso, porém a maioria delas não aparece nos gráficos apresentados, como é o caso das variáveis relacionadas a floricultura, isso por motivo do classificador considerar que não era tão relevantes quanto as apresentadas. Vale ressaltar que todas as variáveis foram concatenadas a uma única tabela que foi utilizada para predizer os resultados apresentados neste caso de uso.

5 CONCLUSÃO

Neste trabalho foi implementado um *framework* com foco em mineração de dados, com intuito de colaborar com as análises de dados recentemente disponibilizados pelo Instituto de Geografia e Estatística (IBGE), por meio do Censo Agropecuário 2017. Após desenvolvida a ferramenta, foi realizado estudos de caso com dados do último censo agropecuário disponível na base de dados SIDRA. A partir dos resultados obtidos no estudo de caso apresentado na seção 4, conclui-se que o *framework* desenvolvido neste trabalho é viável para o processo de extração e geração de análises nas bases de dados agropecuários.

A utilização do algoritmo DBScan foi inviável para agrupar os dados utilizados no estudo de caso, uma vez que apenas algumas variáveis foram agrupadas com o uso deste algoritmo. Portanto, é recomendado a utilização do algoritmo K-médias que apresentou resultados melhores, com o coeficiente de Silhouette igual a: 0.78 para K = 2, 0.74 para K = 3 e 0.64 para K = 4.

Para o conjunto de dados estudado, a árvore de decisão apresentou melhores resultados, atingindo 100% de precisão em todos os testes. Porém com as 118 variáveis utilizadas no estudo de caso, não é possível afirmar que em todo caso a árvore seja melhor que o *naive bayes* que obteve uma precisão de 80%, pois o conjunto estudado é pequeno para deduzir tais conclusões.

5.1 TRABALHOS FUTUROS

O *framework* desenvolvido possui alguns pontos que precisam ser ajustados para que o usuário consiga obter resultados ainda melhores do que os apresentados neste trabalho. Devido ao curto prazo para o desenvolvimento o foco foi implementar as principais funcionalidades. Ao realizar os estudos de caso, houveram algumas dificuldades para processar um conjunto de dados que possui uma quantidade maior de variáveis, pois cada ajuste é feito individualmente por colunas (variáveis), ou seja, se o conjunto de dados possui dez variáveis o usuário deve realizar ajustes individuais em cada uma das dez variáveis. Em trabalhos futuros, adicionar um segundo campo para informar o índice da coluna, com o objetivo de gerar um *range* de colunas e aplicar alterações sobre todas as colunas especificadas em uma única vez.

É uma limitação do *framework* a leitura de arquivos somente no formato CSV BR que esteja padronizado, ou seja, o usuário precisa fazer uma limpeza nos arquivos antes de submetê-

los ao *framework*, como retirar informações sobre a fonte dos arquivos e remoção do cabeçalho padrão que o sistema SIDRA inclui nos arquivos, seria importante em implementações futuras a leitura de todos os modelos de CSV e a exclusão automática de informações que não pertence ao conjunto de dados estudado.

O *framework* possui um módulo de visualização dos dados por meio dos gráficos citados neste trabalho, porém a visualização ficou limitada às figuras. Para melhor entendimento dos resultados seria importante adicionar as legendas nas figuras de forma legível. Para realizar uma melhor classificação de dados agropecuários, é viável implementar outros modelos de classificação no *framework*, que possa estar realizando agrupamento e classificação de dados categóricos, visto que as tabelas disponíveis na base do SIDRA possuem muitas informações categóricas.

Outras técnicas de mineração de dados podem ser implementadas no *framework*, como *cross-validation* que divide as instâncias em um valor k de partições, onde todas as instâncias têm participado do treinamento e do teste. Um classificador por regras de associação também pode ser implementado, podendo ser útil gerar regras de associação entre os dados agropecuários. O *framework* foi desenvolvido em um único arquivo python, para implementações futuras, é interessante aplicar algum padrão de projeto para estruturação do código fonte, assim pode facilitar a inclusão de novas funcionalidades no software, como a integração com outros sistemas como *Power BI*.

Para melhor entendimento do usuário ao utilizar o *framework*, pode ser desenvolvido um manual de uso, apresentando cada passo que é necessário para operar o sistema, de forma que se obtenha bons resultados. O código fonte do *framework* ficará disponível para download e melhoria no repositório do GitHub[3].

5.2 CONSIDERAÇÕES FINAIS

Este trabalho apresentou o desenvolvimento de um software (*framework*) com módulos de captura de dados, pré-processamento, mineração de dados, validação e visualização dos resultados, baseado no conceito de FAYYAD (1996) sobre KDD, o *framework* desenvolvido atende aos requisitos de um sistema com capacidade de realizar a extração de conhecimento em bases de dados.

Considerando os resultados obtidos por meio da mineração dos dados discutidos na

[3] https://github.com/Euristenede/FrameworkMineracaoDados_TCC

seção 4, o *framework* é viável para o uso de demais estudos de casos. Com a implementação das funcionalidades apresentadas na subseção 5.1, os resultados podem ser ainda mais satisfatórios, tornando viável o uso por profissionais e pesquisadores da área de dados agropecuários.

REFERÊNCIAS

ALVES, G. **Aprendizado não supervisionado com K-means.** Coeficiente de Silhouette. 2018. Disponível em: https://medium.com/neuronio-br/aprendizado-n%C3%A3o-supervisionado-com-k-means-f4272dee98a0#:~:text=O%20coeficiente%20de%20Silhouette%20quando,at%C3%A9%20interseccionando%20um%20outro%20cluster.. Acesso em: 15 maio 2021.

BAYES, T. An essay towards solving a problem in the doctrine of chances. 1763. **MD computing: computers in medical practice**, v. 8, n. 3, p. 157-171, 1991.

BEZDEK, J.C. **Pattern recognition with fuzzy objective function algorithms.** 1st ed. New York: Springer Science & Business Media, 1981. 267 p.

BLANCA, L. Normalização de dados: Azure Machine Learning. **Microsoft**, 2020. Disponível em: https://docs.microsoft.com/pt-br/azure/machine-learning/algorithm-module-reference/normalize-data. Acesso em: 14 de maio de 2021.

BOSER, B E.; GUYON, I.M.; VAPNIK, V.N. A training algorithm for optimal margin classifiers. In: **Proceedings of the fifth annual workshop on Computational learning theory**. 1992. p. 144-152.

BOTELHO, E. **Visualização de dados como ferramenta de classificação em sistemas de bases de dados para Data Mining.** 2002. Tese (Doutorado em Ciências Matemáticas e de Computação). Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Pauo, 2002.

BRAGA, J.K. Diferença Entre Informação E Conhecimento**. Abstartups**, 2019. Disponível em: https://abstartups.com.br/diferenca-entre-informacao-e-conhecimento/. Acesso em: 24 de ago. de 2021.

BRITO, S.R. *et al*. Gravidez na adolescência e o acesso às Tecnologias de Informação e Comunicação na Amazônia. **Mundo Amazónico**, v. 6, n. 2, 2015.

CABENA, P. *et al*. **Discovering data mining: from concept to implementation**. Prentice-Hall, Inc., 1998.

CALDAS, M.P.K.; SCANDELARI, L.; PILATTI, L.A. Data Warehouse: Uma classificação de seus Custos e Benefícios. *In*: XIII SIMPÓSIO DE ENGENHARIA DE PRODUÇÃO, 6 a 8 de novembro de 2006, Bauru, São Paulo. **Anais**[...]. Bauru: Unesp, 2006.

CAMILO, C.O.; SILVA, J.C**. Mineração de dados: Conceitos, tarefas, métodos e ferramentas**. 2009. 29 p.

CARVALHO, D.R. **Árvore de decisão/algoritmo genético para tratar o problema de pequenos disjuntos em classificação de dados**. 2005. 162p. Tese (Doutorado em Computação) - Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2005.

CASTILLO, G. Pre-processing using RapidMiner 5.0. **Brazil Documentos**, 2011. Disponível em: https://fdocumentos.tips/document/lesson-2-pre-processing-using-rapidminer-50.html. Acesso em: 15 de maio de 2021.

CLEVELAND, W.S. Data science: an action plan for expanding the technical areas of the field of statistics. **International statistical review**, v. 69, n. 1, p. 21-26, 2001.

CNA. Confederação da Agricultura e Pecuária do Brasil. **CNA Brasil**, 2021. Disponível em https://www.cnabrasil.org.br/noticias/cna-e-ibge-ressaltam-importancia-do-censo-agropecuario-2017 Acesso em: 10 de maio de 2021.

COSTA, E. *et al*. Mineração de dados educacionais: conceitos, técnicas, ferramentas e aplicações. **Jornada de Atualização em Informática na Educação**, v. 1, n. 1, p. 1-29, 2013.

COSTA, L. *et al*. Análise da Associação entre indicadores do trabalho infantil e gravidez na adolescência na Amazônia Legal brasileira. *In*: SIMPÓSIO BRASILEIRO DE SISTEMAS DE INFORMAÇÃO (SBSI), 13., 2017, Lavras. **Anais** [...]. Porto Alegre: Sociedade Brasileira de Computação, 2017. p. 124-127.

DANTAS, E.R.G. *et al*. O uso da descoberta de conhecimento em base de dados para apoiar a tomada de decisões. *In*: SIMPÓSIO DE EXCELÊNCIA EM GESTÃO E TECNOLOGIA, 5., 2008, Rio de Janeiro. **Anais** [...], Rio de Janeiro: AEDB, 2008. p. 50-60.

AMO, S. **Técnicas de mineração de dados**. In: JORNADA DE ATUALIZAÇÃO EM INFORMÁTICA. 2004. Anais... Faculdade de Computação de Uberlândia, Universidade Federal de Uberlândia, Uberlândia 2004.

DELGADO, C.C.N.; DIAS, H.D.; GUELPELI, M.V.C. Utilização de sumários humanos no modelo Cassiopeia. *In*: COMPUTER ON THE BEACH, 4., 2013, Florianópolis. **Anais** [...]. Santa Catarina, 2013. p. 258-267.

DONI, M.V. **Análise de Cluster: métodos hierárquicos e de particionamento**. Monografia (Sistemas de Informação pela Faculdade de Computação e Informática). Universidade Presbiteriana Mackenzie, São Paulo, 2004.

Excel e Accees. Site: **Tire suas dúvidas sobre excel e access**. Publicado em 15 de outubro de 2020. Disponível em http://exceleaccess.com/como-fazer-grafico-de-distribuicao-normal-no-excel/ Acesso em 09 de maio de 2021.

FASIABEN, M. *et al*. Priorização de ações de pesquisa agropecuária baseada em mineração de dados. *In*: CONGRESSO BRASILEIRO DA SOCIEDADE BRASILEIRA DE INFORMÁTICA APLICADA À AGROPECUÁRIA E À AGROINDÚSTRIA, 4., 2003, Porto Seguro. **Anais** [...]. Porto Seguro: SBI Agro, 2003.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. **AI magazine**, v. 17, n. 3, p. 37-37, 1996.

FÉLIX, M.; CARLOS, L. **Data mining: torturando a los datos hasta que confiesen**. 2002. 11 f. Universitat Oberta de Catalunya, Barcelona, 2002. Disponível em: https://www.uoc.edu/web/esp/art/uoc/molina1102/molina1102.html. Acesso em: 30 de ago. de 2021.

FERREIRA, J.C. *et al*. Knowledge Discovery in Database e Data Mining: Uma Contribuição Cibliométrica. In: XXXVIII ENCONTRO NACIONAL DE ENGENHARIA DE PRODUCAO, 38., 2018, Maceió. **Anais** [...]. Maceió: enegep, 2018.

FORLOGIC, Grupo. Diagrama de Dispersão. **Ferramentas da Qualidade** 2016. Disponível em: https://ferramentasdaqualidade.org/diagrama-de-dispersao/. Acesso em: 15 de maio de 2021.

GARCIA, S.C. **O uso de árvores de decisão na descoberta de conhecimento na área da saúde.** 2003. 88 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2003.

GOMES, P.C.T. Conheça as Etapas do Pré-Processamento de dados. **Datageeks,** 2019. Disponível em: https://www.datageeks.com.br/pre-processamento-de-dados/. Acesso em: 15 de maio de 2021.

GUILHERME, P. **Estatística**: Uma ferramenta interdisciplinar. Monografia (Trabalho de Conclusão de Curso). Faculdade Estadual de Filosofia, Licenciatura Plena em Matemática, Paranaguá, Dec. 2008. Disponível em: <https://www.researchgate.net/publication/260201209_Estatistica_Uma_ferramenta_interdisciplinar>. Acesso em 09 de maio de 2021.

HAN, J.; KAMBER M.; PEI, J. **Data Mining: concepts and techniques**. 3nd ed. Amsterdam; Boston: Elsevier: Morgan Kaufmann, 2011.

HAND, D.J. Principles of data mining. **Drug safety**, v. 30, n. 7, p. 621-622, 2007.

HERRMANN, M. Qt Designer. O que é Qt Designer?. **Fman Build System**, 2021. Disponível em: https://build-system.fman.io/qt-designer-download. Acesso em: 15 de maio de 2021.

HOLLAND, J.H. *et al*. **Adaptation in natural and artificial systems**: an introductory analysis with applications to biology, control, and artificial intelligence. 1st ed. Cambridge: Bradford Book, 1992. 232 p.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA - IBGE. **Censo Agropecuário 2017**. IBGE, 2019. Disponível em: <https://sidra.ibge.gov.br/pesquisa/censo-agropecuario/censo-agropecuario-2017>.Acesso em: 10 de maio de 2021.

INSTITUTO DE GEOGRAFIA E ESTATÍSTICA - IBGE. **Manual do Recenseador CA - 1.09**. IBGE, 2017. Disponível em: https://biblioteca.ibge.gov.br/visualizacao/instrumentos_de_coleta/doc5537.pdf. Acesso em: 10 de ago. de 2021.

KUNZ, T.; BLACK, J.P. Using automatic process clustering for design recovery and distributed debugging. **IEEE Transactions on Software Engineering**, Nova Jersey v. 21, n. 6, p. 515-527, 1995.

LEMOS, E.P.; STEINER, M.T.A.; NIEVOLA, J.C. Análise de crédito bancário por meio de redes neurais e árvores de decisão: uma aplicação simples de data mining. **Revista de Administração-RAUSP**, v. 40, n. 3, p. 225-234, 2005.

LI, Ray. História da mineração de dados. **Kdnuggets**, 2016. Disponível em: https://www.kdnuggets.com/2016/06/rayli-history-data-mining.html. Acesso em: 17 de abr. de 2021.

LOPES, Guilherme. Aquisição e pré-processamento de dados para Machine Learning. MinMax. **Mediun**, 2019. Disponível em: https://medium.com/@gslp1994/aquisi%C3%A7%C3%A3o-e-pr%C3%A9-processamento-de-dados-para-machine-learning-2fcffe0fc965. Acesso em: 15 de maio de 2021.

LORENSINI, Carolina Lobello; OLIVEIRA, SR de M.; VICTORIA, D. de C. Modelos preditivos para classificação de aptidão agrícola de municípios. In: MOSTRA DE ESTAGIÁRIOS E BOLSISTAS DA EMBRAPA INFORMÁTICA AGROPECUÁRIA, 14., 2018, Campinas. **Resumos expandidos...** Brasília, DF: Embrapa, 2018. Disponível em: http://www.alice.cnptia.embrapa.br/alice/handle/doc/1101328. Acesso em 30 de ago. de 2021.

LUDERITZ, Edward. Gráfico de Pontos. **Nelogica**, 2021. Disponível em: https://ajuda.nelogica.com.br/hc/pt-br/articles/360041299251-Gr%C3%A1fico-de-Pontos. Acesso em: 15 de maio de 2021.

LUIZ, Robson. Gráficos. **Brasil Escola**, 2021. Disponível em: https://brasilescola.uol.com.br/matematica/graficos.htm. Acesso em: 27 de agosto de 2021.

LUZ, Adson Whobbert da. **Comparação de ferramentas de data warehouse**: estudo de caso com dados do IBGE. 2017. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná.

MACEDO JUNIOR, Celso *et al*. Utilização De Regras De Associação Para Relacionar Dados Meteorológicos E Dados De Produtividade Do Café No Estado De São Paulo. *In*: SIMPÓSIO DE PESQUISA DOS CAFÉS DO BRASIL, 6., 2009. Vitória, ES. **Anais** [...]. Vitória: Embrada, 2009.

MANHÃES, Laci Mary Barbosa et al. Previsão de Estudantes com Risco de Evasão Utilizando Técnicas de Mineração de Dados. **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)**, [S.l.], out. 2012. ISSN 2316-6533. Disponível em: <http://www.br-ie.org/pub/index.php/sbie/article/view/1585>. Acesso em: 30 ago. 2021. doi:http://dx.doi.org/10.5753/cbie.sbie.2011.%p.

MARIA DAS GRAÇAS, J. M. *et al*. **Aplicação de Técnicas de Mineração de Dados para Caracterização de Grupos de Cidades Produtoras de Cana-de-Açúcar do Estado de São Paulo e Definição de Políticas Específicas**. 2010. 9 f. Faculdade de Tecnologia de Indaituba, FATEC, Indaituba, 2010.

MATPLOTLIB. Visualização com Python. **Matplotlib**, 2021. Disponível em: https://matplotlib.org/. Acesso em: 15 de maio de 2021.

MCCULLOCH, Warren S.; PITTS, Walter. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, v. 5, n. 4, p. 115-133, 1943.

MCLEOD, Saul. Pontuação Z: Definição, Cálculo e Interpretação. **Simply Psychology**, 2019. Disponível em: https://www.simplypsychology.org/z-score.html. Acesso em: 15 de maio de 2021.

MONARD, Maria Carolina; BARANAUSKAS, José Augusto. Indução de regras e árvores de decisão. **Sistemas Inteligentes-Fundamentos e Aplicações**, v. 1, p. 115-139, 2003.

MONTEIRO, Márcio Ozal de Abreu. **A análise preditiva sob o aspecto da regulação.** 2018. 49 f. Dissertação (Mestrado em Estratégia de Investimento e Internacionalização) – Instituto Superior de Gestão, Lisboa, 2018.

MORAIS, Carlos. **Descrição, análise e interpretação de informação quantitativa**. Escalas de medida, estatística descritiva e inferência estatística. 2010. 30 f. Escola Superior de Educação-Instituto Politécnico de Bragança, 2010. Disponível em: http://www.ipb.pt/~cmmm/discip/ConceitosEstatistica.pdf. Acesso em: 28 de ago. de 2021.

NEVES, Rita de Cássia David das. **Pré-processamento no processo de descoberta de conhecimento em banco de dados**. 2003. 137 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2003.

PINO, Francisco Alberto. A questão da não normalidade: Uma revisão. **Revista de economia agrícola**, v. 61, n. 2, p. 17-33, 2014.

PYDATA. Biblioteca Pandas. **Pandas**, 2021. Disponível em: https://pandas.pydata.org/. Acesso em: 15 de maio de 2021.

PYTHON. Linguagem de programação python. **Python**, 2021. Disponível em: https://www.python.org/. Acesso em: 15 de maio de 2021.

RELICH, M., MUSZYNSKI, W. The use of intelligent systems for planning and scheduling of product development projects. **Procedia Computer Science**, v. 35, p. 1586-1595, 2014.

ROCHA, Miguel; CORTEZ, Paulo; NEVES, José Maia. **Análise inteligente de dados**: algoritmos e implementação em Java. 1. ed. Lisboa: FCA editora, 2008. 204 p.

SANTOS, Ricardo Braz. **IDENTCAMP**: Algoritmo computacional para identificação da produção camponesa a partir de dados secundários. 2016. 47 f. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) – Faculdade de Computação e Engenharia Elétrica, Universidade Federal do Sul e Sudeste do Pará, Marabá, 2016.

SANTOS, Rafael et al. Conceitos de Mineração de dados na web. *In*: SIMPÓSIO BRASILEIRO DE SISTEMAS MULTIMÍDIA E WEB, 15., 2009, São Paulo. **Anais** [...]. São Paulo: USP, 2009. p. 81-124.

SCHMITT, Jeovani *et al*. **Pré-processamento para a mineração de dados**: uso da análise de componentes principais com escalonamento ótimo. 2005. 146 f. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Santa Catarina, Florianópolis, 2005.

SCHMITT, Vinícius Fernandes. **Uma análise comparativa de técnicas de aprendizagem de máquina para prever a popularidade de postagens no facebook**. 2013. 57 f. Trabalho de

Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2013.

SCIKIT-LEARN. Aprendizado de máquina em Python. **Scikit-learn**, 2021. Disponível em: https://scikit-learn.org/stable/index.html. Acesso em: 15 de maio de 2021.

SHAPIRO, Gregory Piatetsky. Knowledge Discovery in Databases (KDD). **Kdnuggets**, 1989. Disponível em: https://www.kdnuggets.com/meetings-past/kdd89/index.html. Acesso em: 17 de abr. de 2021.

SILVA FILHO, Arivaldo Pereira; DA SILVA, Samuel Bueno; HYPÓLITO, João Mauricio. **Data Mining Através Da Regra De Associação Apriori**. 2013. 4 f. Faculdade de Tecnologia do Estado de São Paulo, FATEC, Ourinhos, 2013.

SILVANO, Tiago Prudencio; CORREA, Bryan Maia; BARBOSA, Ivanildo. Análise da distribuição espacial de indicadores sociais e demográficos: uma abordagem baseada em mineração de dados. **Revista Brasileira de Cartografia**, v. 72, n. 1, p. 67-80, 2020.

SOMBRA, Tobias Ribeiro *et al*. **Reconhecimento de padrões em rede social científica**: aplicação do algoritmo Naive Bayes para classificação de papers no Mendeley. 2018. 198 f. Dissertação (Mestrado em Ciência da Informação) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2018.

SOUSA, Clycia Najara Silva. **Identificação de destinos turísticos baseada em densidade espacial de fotos**. 2016. 37 f. Trabalho de Conclusão de Curso (Tecnólogo em Redes de Computadores) – Universidade Federal do Ceará, Quixadá, 2016.

SOUZA, Emanuel G de. Entendendo o que é Matriz de Confusão com Python. **Mediun**, 2019. Disponível em: https://medium.com/data-hackers/entendendo-o-que-%C3%A9-matriz-de-confus%C3%A3o-com-python-114e683ec509. Acesso em: 15 de maio de 2021.

TURING, Alan Mathison. On computable numbers, with an application to the Entscheidungsproblem. **Proceedings of the London mathematical society**, v. 2, n. 1, p. 230-265, 1937.

VIEIRA FILHO, Júlio Lima. **Proposta de framework para sistemas de mineração de dados socioeconômicos**. 2013. 49 f. Trabalho de Conclusão de Curso (Bacharel em Ciência da Computação) – Universidade Federal do Maranhão, São Luiz, 2013.

YANG, Ying. **Discretization for naive-bayes learning**. 2003. 163 p. Tese (Doutorado em Ciência da Computação e Engenharia de Software) - Monash University, Melbourne, Austrália, 2003.

YONEYAMA, Takashi. **Discretização para Aprendizagem Bayesiana**: Aplicação no Auxílio à Validação de Dados em Proteção ao Vôo. 2003. 69 p. Dissertação (Mestrado em Engenharia Eletrônica e Computação) - Instituto Tecnológico de Aeronáutica, ITA, São José dos Campos, 2003.

ZEVIANI, Walmes Marques. Manipulação de dados no R. **Material de aula**, 03 de junho de 2019. 93 p. Disponível em: http://www.leg.ufpr.br/~walmes/ensino/dsbd-linprog/slides/02-r-tidyverse.pdf. Acesso em: 15 de maio de 2021.

ZIBETTI, Andre. Distribuição Normal (Gaussiana) Distribuição Normal. **UFSC**, 2019. Disponível em: https://www.inf.ufsc.br/~andre.zibetti/probabilidade/normal.html. Acesso em: 15 de maio de 2021.

ZOUBI, Moh'd B. Al-; RAWI, Mohammad al. An efficient approach for computing silhouette coefficients. **Journal of computer science**, v. 4, n. 3, p. 252, 2008.