



Universidade Federal do Piauí  
Centro de Ciências da Natureza  
Programa de Pós-Graduação em Ciência da Computação

# **Rotulação Automática de Grupos Através de Algoritmos Supervisionados baseados em Árvores e Estatísticos**

**Tarcísio Franco Jaime**

**Número de Ordem PPGCC: M001**

**Teresina-PI, Junho de 2018**



Tarcísio Franco Jaime

# **Rotulação Automática de Grupos Através de Algoritmos Supervisionados baseados em Árvores e Estatísticos**

**Qualificação de Mestrado** apresentada ao Programa de Pós-Graduação em Ciência da Computação da UFPI (área de concentração: Sistemas de Computação), como parte dos requisitos necessários para a obtenção do Título de Mestre em Ciência da Computação.

Universidade Federal do Piauí – UFPI

Centro de Ciências da Natureza

Programa de Pós-Graduação em Ciência da Computação

Orientador: Vinicius Ponte Machado

Teresina-PI

Junho de 2018

---

Tarcísio Franco Jaime

Rotulação Automática de Grupos Através de Algoritmos Supervisionados baseados em Árvores e Estatísticos/ Tarcísio Franco Jaime. – Teresina-PI, Junho de 2018-

58 p. : il.

Orientador: Vinicius Ponte Machado

Qualificação (Mestrado) – Universidade Federal do Piauí – UFPI

Centro de Ciências da Natureza

Programa de Pós-Graduação em Ciência da Computação, Junho de 2018.

1. Rotulação. 2. Algoritmos Supervisionados. 3. CART. 4. Naive Bayes. I. Dr. Vinicius Ponte Machado. II. Universidade Federal do Piauí. III. Rotulação Automática de Grupos Através de Algoritmos Supervisionados baseados em Árvores e Estatísticos.

CDU 02:141:005.7

---

Tarcísio Franco Jaime

## **Rotulação Automática de Grupos Através de Algoritmos Supervisionados baseados em Árvores e Estatísticos**

**Qualificação de Mestrado** apresentada ao Programa de Pós-Graduação em Ciência da Computação da UFPI (área de concentração: Sistemas de Computação), como parte dos requisitos necessários para a obtenção do Título de Mestre em Ciência da Computação.

Teresina-PI, 21 de junho de 2018:

---

**Vinicius Ponte Machado**  
Orientador

---

**Raimundo Santos Moura**

---

**Erico Meneses Leão**

Teresina-PI  
Junho de 2018



# Resumo

Com o avanço da tecnologia cada vez mais equipamentos estão se conectando nas redes gerando fluxos e processamento de dados, com isso, mais algoritmos de aprendizado de máquina estão sendo estudados para extraírem informações relevantes desses grandes volumes. Com o grande aumento desse fluxo de dados a interpretação dos mesmos podem ser prejudicada sendo o grau de dificuldade proporcional a esse crescimento. É nesse contexto que essa pesquisa atua, pois alguns algoritmos de aprendizado de máquina criam grupos de dados que possuem algumas características, e nesse trabalho foi realizado uma pesquisa científica com objetivo de identificar nesses grupos quais são os atributos mais significativos junto com os valores que mais se repetem a ponto de representar o grupo, chamando essa técnica de rotulação. Dessa forma, esta pesquisa utiliza nessa técnica algoritmos supervisionados, já implementados por um software de cálculo numérico (MATLAB), onde pretende-se rotular grupos já criados em diferentes bases de dados exibindo um resultado em porcentagem de acordo com o número de registros que são representados pelo rótulo criado.

**Palavras-chaves:** grupos. rotulação. aprendizado supervisionado.





# Abstract

**Keywords:** cluster. rotulação.



# Lista de ilustrações

Figura 1 – Hipóteses ajustadas – Função $h$ próxima da função $f$ real . . . . .	6
Figura 2 – Exemplo de Fluxograma de árvore: R1 a R5 são as folhas relacionadas de acordo com as respostas sim ou não dos nós . . . . .	7
Figura 3 – Exemplo do funcionamento do algoritmo KNN . . . . .	11
Figura 4 – Exemplos de técnicas diferentes utilizada por algoritmos para dividir em grupos . . . . .	13
Figura 5 – Ponto de Corte (R-1); 2(R) significa o valor de R=2; $2 - 1 = 1$ . . . . .	14
Figura 6 – Discretização EWD . . . . .	15
Figura 7 – Discretização EWD de acordo com a amostra da tabela 3 . . . . .	15
Figura 8 – Discretização EFD . . . . .	17
Figura 9 – Discretização EFD de acordo com a amostra da tabela 3 . . . . .	18
Figura 10 – Modelo de (LOPES; MACHADO; RABELO, 2016) . . . . .	20
Figura 11 – Modelo de Resolução Proposto . . . . .	25
Figura 12 – Exemplo da técnica de correlação aplicada aos atributos atr1, atr2 e atr3. . . . .	26
Figura 13 – Montagem da Matriz de Atributos Importantes . . . . .	27
Figura 14 – Exemplo da técnica de correlação aplicada ao atributo, atr1, sendo classe . . . . .	27
Figura 15 – Discretização de atributos utilizando EFD com $R = 3$ (Figura adaptada de (LOPES; MACHADO; RABELO, 2016)) . . . . .	29
Figura 16 – Resultado dos Algoritmos . . . . .	31
Figura 17 – Gráfico de Execuções dos algoritmos supervisionados na base de dados SEEDS. . . . .	50
Figura 18 – Gráfico de Execuções dos algoritmos supervisionados na base de dados IRIS. . . . .	51
Figura 19 – Gráfico de Execuções do algoritmo supervisionado Naive Bayes na base de dados GLASS. . . . .	52
Figura 20 – Gráfico de Execuções do algoritmo supervisionado CART na base de dados GLASS. . . . .	52
Figura 21 – Acurácia por Clusters (Os clusters estão numerados em ordem crescente em cada Base de Dados) . . . . .	53



# Lista de tabelas

Tabela 1	– Base de exemplo onde exhibe através do atributo Decisão, se existe, ou não condição de jogo perante as outras características (Aspecto, temperatura, Umidade e Vento) . . . . .	10
Tabela 2	– Tabela em ordem crescente das distâncias euclidianas encontradas do objeto a que se deseja classificar para os outros objetos da amostra, de acordo com as setas da figura 3 . . . . .	12
Tabela 3	– Amostra de dados para exemplificar a discretização EWD e EFD . . . . .	16
Tabela 4	– Base de Dados Modelo . . . . .	28
Tabela 5	– Base de Dados Modelo Discretizada . . . . .	30
Tabela 6	– Valores das faixas com R=3 da Base de Dados Modelo . . . . .	30
Tabela 7	– Resultado da rotulação com o algoritmo Naive Bayes . . . . .	36
Tabela 8	– Resultado da Correlação dos atributos pelo Naive Bayes; Legenda dos Atributos: (A)area, (B)perimetro, (C)compactness, (D)Lkernel, (E)Wkernel, (F)asymetry, (G)lkgroove . . . . .	37
Tabela 9	– Resultado de 4 (quatro) execuções do algoritmo Naive Bayes; Legenda dos Atributos: (A)area, (B)perimetro, (C)compactness, (D)Lkernel, (E)Wkernel, (F)asymetry, (G)lkgroove . . . . .	38
Tabela 10	– Resultado da aplicação do algoritmo CART . . . . .	38
Tabela 11	– Resultado de 4 (quatro) execuções do algoritmo Naive Bayes; Legenda dos Atributos: (A)area, (B)perimetro, (C)compactness, (D)Lkernel, (E)Wkernel, (F)asymetry, (G)lkgroove . . . . .	39
Tabela 12	– Resultado da aplicação do algoritmo Naive Bayes . . . . .	40
Tabela 13	– Resultado (em %) de 4 (quatro) execuções do algoritmo Naive Bayes; Legenda dos Atributos: (SL)sepalength, (SW)sepalwidth, (PL)petallength, (PW)petalwidth . . . . .	41
Tabela 14	– Resultado da aplicação do algoritmo CART . . . . .	41
Tabela 15	– Resultado de 4 (quatro) iterações do algoritmo CART; Legenda dos Atributos: (SL)sepalength,(SW)sepalwidth,(PL)petallength,(PW)petalwidth . . . . .	42
Tabela 16	– Resultado da aplicação do algoritmo Naive Bayes . . . . .	44
Tabela 17	– Resultado de 4 (quatro) execuções do algoritmo Naive Bayes. . . . .	45
Tabela 18	– Resultado da aplicação do algoritmo CART . . . . .	46
Tabela 19	– Resultado de 4 (quatro) execuções do algoritmo CART. . . . .	47
Tabela 20	– Resultado da rotulação utilizando Redes Neurais (??) referente a base de dados SEEDS. . . . .	54
Tabela 21	– Resultado da rotulação utilizando Naive Bayes referente a base de dados SEEDS. . . . .	54

Tabela 22 – Resultado da rotulação utilizando CART referente a base de dados  
SEEDS. . . . . 54

Tabela 23 – Cronograma de atividades . . . . . 55

# Lista de abreviaturas e siglas

ANN	Artificial Neural Networks
EWD	Equal Width Discretization
EFD	Equal Frequency Discretization
CART	Classification and Regression Trees
RNA	Redes Neurais Artificiais
GP	Grau de Pertinência
GS	Grau de Seleção
IGS	Incremento do Grau de Seleção
SVM	Support Vector Machine
TEDA	Typicality and Eccentricity Data Analytics





# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>4</b>
<b>2.1</b>	<b>Aprendizado de Máquina</b>	<b>4</b>
2.1.1	Aprendizado Supervisionado	5
2.1.1.1	Algoritmo Classification And Regression Trees - CART	6
2.1.1.2	Algoritmo Naive Bayes	8
2.1.1.3	Algoritmo k-Nearest Neighbor - KNN	11
2.1.2	Aprendizado Não-Supervisionado	12
<b>2.2</b>	<b>Discretização</b>	<b>14</b>
2.2.1	Equal Weight Discretization - EWD	14
2.2.2	Discretização por Frequência Iguais - EFD	16
<b>2.3</b>	<b>Trabalhos Correlatos</b>	<b>18</b>
<b>3</b>	<b>METODOLOGIA</b>	<b>23</b>
<b>3.1</b>	<b>Rotulação de Cluster</b>	<b>23</b>
<b>3.2</b>	<b>O Modelo de Resolução</b>	<b>24</b>
<b>3.3</b>	<b>Técnica de Correlação entre Atributos através de Algoritmos Super-</b>	
	<b>visionados</b>	<b>26</b>
<b>3.4</b>	<b>Exemplo</b>	<b>28</b>
3.4.1	Processo (I) - Discretização	29
3.4.2	Processo (II) - Algoritmos Supervisionados	30
3.4.3	Processo (III) - Rotulação	31
<b>4</b>	<b>RESULTADOS</b>	<b>34</b>
<b>4.1</b>	<b>Implementação</b>	<b>34</b>
<b>4.2</b>	<b>Seeds - Identificação de Tipos de Semente</b>	<b>35</b>
4.2.1	Naive Bayes	36
4.2.2	CART	37
<b>4.3</b>	<b>Iris - Identificação de Tipos de Plantas</b>	<b>39</b>
4.3.1	Naive Bayes	40
4.3.2	CART	41
<b>4.4</b>	<b>Glass - Identificação de Tipos de Vidros</b>	<b>42</b>
4.4.1	Naive Bayes	43
4.4.2	CART	44

<b>5</b>	<b>CONCLUSÕES, TRABALHOS FUTUROS E CRONOGRAMA . . .</b>	<b>48</b>
<b>5.1</b>	<b>Conclusão . . . . .</b>	<b>48</b>
<b>5.2</b>	<b>Trabalhos Futuros . . . . .</b>	<b>54</b>
<b>5.3</b>	<b>Cronograma . . . . .</b>	<b>55</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>56</b>

# 1 Introdução

Agrupamento de dados, ou clustering, é o termo usado para identificar dois ou mais objetos pertencentes ao mesmo grupo que compartilham um conceito em comum (KUMAR; ANDU; THANAMANI, 2013). Cluster é um termo bastante pesquisado no aprendizado não-supervisionado (subárea do aprendizado de máquina) e aplicado em vários contextos como segmentação de imagens, recuperação de informação e reconhecimento de objetos. Os algoritmos de agrupamento, conforme (KUMAR; ANDU; THANAMANI, 2013), são aplicados em diferentes campos: Biologia (classificação de plantas e animais), Marketing (encontrar grupos de clientes com comportamentos semelhantes), planejamento de cidades (identificação de casas de acordo com seu tipo, valor e localização geográfica), entre outros.

Com a popularização da internet e mídias sociais, cada vez mais dados são processados, transportados e produzidos. É nesse cenário, com grandes volumes de dados, que não só a formação de grupos ganha importância, mas também a compreensão dos mesmos, pois a interpretação dos grupos fornecerá informações úteis para análises desses clusters.

O grau de escalabilidade dos dados gradativamente aumenta no decorrer dos anos e, embora os estudos sobre o problema de agrupamento de dados estejam avançados, fica cada vez mais complexo entender como são formados esses clusters pela razão do número crescente de grupos criados. Quanto maiores são os números de grupos produzidos, mais difíceis são suas interpretações.

Diante desse contexto é que se extrai a temática desta proposta de mestrado - Rotulação automática de grupos através de algoritmos supervisionados baseados em árvores e estatísticos - o estudo em questão dedica-se na aplicabilidade de algoritmos supervisionados, com bases de dados distintas, a fim de definir a tupla atributo/valor de maior importância nos clusters, determinando um significado para estes clusters (rotulação).

A rotulação dita neste trabalho segue a própria definição da palavra, que serve para informar sobre algo. Então, a partir de um grupo de dados, seria possível destacar neste grupo uma informação que o represente, e uma forma seria encontrar através de técnicas uma tupla: atributo(s) e faixa(s). Onde o atributo selecionado seria o que teria maior relevância no grupo, no sentido de representar o grupo, e a faixa de valor escolhida, seria a que mais tivesse ocorrência nos valores do atributo. Poderá também haver no grupo mais de um atributo com sua respectiva faixa, representando o rótulo.

O termo rotulação, neste trabalho, segue a definição conforme (LOPES; MACHADO; RABELO, 2016):

**Definição 1** Dado um conjunto de clusters  $C = \{c_1, \dots, c_k | K \geq 1\}$ , de modo que cada cluster contém um conjunto de elementos  $c_i = \{\vec{e}_1, \dots, \vec{e}_{n(c_i)} | n(c_i) \geq 1\}$  que podem ser representados por um vetor de atributos definidos em  $\mathbb{R}^m$  e expresso por  $\vec{e}^i = (a_1, \dots, a_m)$  e ainda que com  $c_i \cap c_{i'} = \emptyset$  com  $1 \leq i, i' \leq K$  e  $i \neq i'$ ; o objetivo consiste em apresentar um conjunto de rótulos  $R = \{r_{c_1}, \dots, r_{c_k}\}$ , no qual cada rótulo específico é dado por um conjunto de pares de valores, atributo e seu respectivo intervalo,  $r_{c_i} = \{(a_1, [p_1, q_1]), \dots, (a_{m(c_i)}, [p_{m(c_i)}, q_{m(c_i)}])\}$  capaz de melhor expressar o cluster  $c_i$  associado.

- $K$  é o número de clusters;
- $c_i$  é o  $i$ -ésimo cluster;
- $n^{c_i}$  é o número de elementos do cluster  $c_i$ ;
- $\vec{e}_{j(c_i)}$  se refere ao  $j$ -ésimo elemento pertencente ao cluster  $c_i$ ;
- $m$  é a dimensão do problema;
- $r_{c_i}$  é o rótulo referente ao cluster  $c_i$ ;
- $[p_{m(c_i)}, q_{m(c_i)}]$  representa o intervalo de valores do atributo  $a_{m(c_i)}$ , onde  $p_{m(c_i)}$  é o limite inferior e  $q_{m(c_i)}$  é o limite superior;

A formação do problema desta pesquisa nasce a partir do trabalho realizado por (LOPES; MACHADO; RABELO, 2016), que se dedicou a estudar a possibilidade de realização de rotulação automática de grupos utilizando para isso dois algoritmos: i) Um para realizar a formação de grupos através de algoritmo não supervisionado (K-means); e ii) utiliza o algoritmo supervisionado (Redes Neurais Artificiais - RNA) para fazer a rotulação de grupos. Assim, partindo do estudo já realizado, este trabalho se dedica a realizar rotulação de grupos de dados a partir de outros algoritmos supervisionados não testados, e realizando um comparativo entre eles.

Nesta pesquisa foi aferida a acurácia de cada resultado através do percentual de acertos dos atributos que são representados pelos rótulos gerados, sendo essa acurácia possível em virtude das bases de dados escolhidas já serem classificadas, possibilitando o comparativo dos rótulos encontrados com a classificação da base de dados. É importante destacar que um pré-requisito para utilização de uma base para teste, é esta base conter registros já pertencente a algum grupo. Isto posto, no desenvolvimento deste trabalho não há preocupação na criação de grupos, e sim, na rotulação dos mesmos, isto é, compreender os grupos de dados já formados.

Quando se analisa grupos que já estão formados sabe-se que esses grupos existem, pois há uma correlação das características pelos quais seus dados se mantêm juntos em grupos. Acontece que, com grandes números de grupos sendo criados, isso acaba por não deixar visível qual característica se apresenta mais significativa dentro desses grupos. Tem-se na rotulação a intenção de definir algum significado para estes grupos, gerando um

tipo de rótulo,  $R = \{r_{c1}, \dots, r_{ck}\}$ , para melhor expressar o cluster  $c_i$  associado (Definição 1).

Tecnicamente a informação do rótulo aplicada no cluster pode ajudar na tomada de decisão em algum contexto. A exemplo disso, supõe-se uma situação empregada na área urbana, onde pessoas circulam na cidade e imagina-se que os dados de controle de seus celulares estão sendo capturados pelas células das torres e gravados em uma base de dados pelas operadoras. Uma vez em posse desses dados, são criados clusters podendo ser aplicado rotulação nestes grupos, e através dos rótulos pode-se personalizar alguns serviços para esses grupos já formados.

Seguindo o exemplo dos dados capturados do celular, caso o rótulo ( $r_{c_i}$ ) de um cluster ( $c_i$ ) fosse o atributo localização, e os valores desse atributo escolhido para compor o rótulo, fossem as coordenadas geográficas, o qual definiriam o tipo de localização. Logo percebe-se que os participantes desse grupo possuem característica de frequentar alguma localização em comum. A interpretação deste rótulo poderá implicar uma tomada de decisão personalizada para este grupo, objetivando otimizar um problema.

O trabalho será disposto em cinco capítulos já incluso a Introdução e Conclusão, capítulos 1 e 5 respectivamente. O Referencial Teórico abordado no capítulo 2 esclarece as tecnologias utilizadas nesta pesquisa sendo dividida em três seções. Inicialmente na seção 2.1, tem-se uma explanação sobre aprendizado de máquina e quais os aprendizados indutivos são mais relevantes para este trabalho, ademais, a explicação dos algoritmos supervisionados utilizados para fazer rotulação de dados. Já na seção 2.2 é realizado a divisão das faixas de valores de cada atributo, chamada de discretização. E logo na seção 2.3 são apresentas pesquisas já consolidadas referentes a assuntos como aprendizado de máquina, classificação, agrupamentos e rotulação de dados.

No capítulo 3 é abordado a definição do problema da pesquisa. A partir dessa definição um modelo de resolução é definido e apresentado um fluxograma exibindo os processos a serem seguidos. Logo na seção 3.3 é demonstrado o funcionamento da técnica de correlação entre atributos. E na seção 3.4 uma base de dados fictícia é utilizada para exemplificar a execução dos processos do modelo de resolução nas seguintes etapas: discretização da base de dados, a aplicação do algoritmo supervisionado e resultado da rotulação.

No capítulo 4 os resultados são separados por base de dados. Em cada seção referente a uma base de dados testada são criadas duas subseções referentes aos algoritmos utilizados. Cada algoritmo apresenta uma tabela com informações desde o número do cluster, rótulos, relevância do atributo até acurácia do rótulo no cluster. É também exibida uma tabela possuindo os valores de relevância dos atributos por cluster, posto que esses valores desta tabela servirão de apoio para entender como o quão os atributos estão bem correlacionados.

## 2 Referencial Teórico

Para se compreender a temática proposta este capítulo abordará o conteúdo base deste trabalho dividido em 3 seções: Aprendizado de Máquina, Discretização e Trabalhos Correlatos.

Essa primeira seção discorrerá sobre aprendizado de máquina e os aprendizados indutivos, embora cada aprendizado tenham suas importâncias será dado maior ênfase ao aprendizado supervisionado, pelo fato da utilização de algoritmos de aprendizado supervisionado na rotulação de grupos. Já na seção 2.2 dissertará sobre as técnicas de discretizações adotadas nesta pesquisa, onde há grande contribuição para os resultados gerados e ganhando assim uma seção própria para explanação do funcionamento dessas técnicas. Na última seção serão abordados trabalhos com mesmas características desta pesquisa adicionando conhecimento ao tema.

### 2.1 Aprendizado de Máquina

A aprendizagem de máquina, diferente das metodologias tradicionais de implementação, utiliza sua experiência anterior, para melhorar suas respostas a partir de problemas em determinadas áreas.

“Um programa de computador aprende com a experiência  $E$  em relação a alguma classe de tarefas  $T$  e medida de desempenho  $P$ , se seu desempenho em tarefas em  $T$ , conforme medido por  $P$ , melhora com a experiência  $E$ ” (MITCHELL, 1997, p. 2).

O aprendizado de máquina correspondem a algoritmos capazes de aprender automaticamente através de determinados exemplos, ou comportamentos. Esse aprendizado automático preenche algumas lacunas no desenvolvimento de programas, posto que não é possível simplesmente exigir do projetista implementar melhorias em um sistema, de forma que ele esteja robusto bastante para lidar com todas as situações (RUSSEL; NORVIG, 2013), pois seria impossível um programador antecipar todas as situações possíveis de implementação.

Utilizando a ideia acima, uma vez inserida uma foto no banco de dados e determiná-la como masculina, nesse momento, estará se fazendo uma classificação desse novo registro (nova foto). Uma vez com a base de dados classificada, pode-se utilizar algoritmos para prever um novo registro e defini-lo como masculino ou feminino. Prever uma determinada condição dependerá da base de dados como também do algoritmo utilizado para fazer essa classificação. Alguns exemplos de algoritmos são: RNA, K-Nearest Neighbor - KNN, Suport Vector Machine – SVM, etc. A escolha apropriada do algoritmo se dará

através de métricas que avaliarão seus desempenhos, e a melhor servirá de parâmetro para a escolha do algoritmo apropriado para aquele problema de classificação de dados.

Segundo (MOHRI; ROSTAMIZADEH; TALWALKAR, 2012) o aprendizado de máquina possui abordagens diferentes, são elas: aprendizado supervisionado, aprendizado não-supervisionado, aprendizado semi-supervisionado, aprendizado por reforço. Todavia, nesta pesquisa serão comentados somente as abordagens de referência específicas utilizadas neste trabalho.

### 2.1.1 Aprendizado Supervisionado

O aprendizado supervisionado é um método que através de uma base de dados classificada, será realizado uma predição de novos registros com base em vários desses exemplos já classificados, ou seja, é quando existem casos que possuem uma classificação disponível para determinados conjuntos de dados (conjunto de treinamento), mas precisa ser previsto para outras instâncias. Os responsáveis por essas predições de novos registros são algoritmos de aprendizado supervisionados projetados para determinados fins.

O termo “Supervisionado” indica uma correlação entre os dados de entrada com a saída desejada (classe), e seguindo essa afirmação considere uma base de dados de imagens de rostos, onde cada imagem possui uma saída representada por uma classe (masculino ou feminino). A tarefa seria criar um preditor capaz de acertar a cada novo registro se a imagem é masculina ou feminina. Seria difícil implementar de maneira tradicional, utilizando estruturas condicionais e laços, uma vez que são inúmeras as diferenças das faces masculinas e femininas. Embora haja uma dificuldade de distinção entre as faces, uma alternativa seria dar exemplos de rostos classificados, masculino ou feminino, e através desses exemplos aplicar o algoritmo que automaticamente faça a máquina aprender uma regra para predizer qual sexo pertence cada rosto (BARBER, 2011).

Em (RUSSEL; NORVIG, 2013), é feita apresentação formal do funcionamento da aprendizagem supervisionada, pois dado um conjunto de treinamento,  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ , onde cada  $y_j$  foi gerado por  $y = f(x)$  desconhecida, encontrar uma função  $h$  (hipótese) dentre várias possíveis, que se aproxime ao máximo da função  $f$  (real). Quanto mais próxima de  $f$  melhor o desempenho da função  $h$ , mas para medir esse desempenho é testado um conjunto diferente (dados de teste) do conjunto de treinamento, e aferida a precisão da função hipótese.

O exemplo da figura 1a mostra a função  $h$  de grau 6 onde acontece um sobreajuste (*overfitting*) no conjunto de dados de treinamento. Esse modelo exibe uma função mais complexa para atender todo o conjunto de dados do gráfico, tornando-se um modelo específico para essa amostra de dados.

Já na figura 1b o ajuste da função  $h$  se torna mais simples e mesmo o gráfico não



Figura 1 – Hipóteses ajustadas – Função  $h$  próxima da função  $f$  real

passando por todos os pontos, acabou por generalizar melhor o conjunto de treinamento, tornando, um melhor resultado da predição de novos valores.

Em análise da figura 1 é apresentado duas hipóteses que tentam se aproximar ao máximo da função verdadeira ( $f$ ), que é desconhecida. Mesmo parecendo que na figura 1a obteve-se melhor resultado, pois todos os pontos são atingidos pelo gráfico da função, este modelo acabou se ajustando muito bem na amostra de dados deixando a função  $h$  muito específica, não retratando os dados em um mundo real. Então, apesar de parecer que a figura 1a é a melhor opção por ela ser mais específica, esta, não é a melhor opção, pois quanto mais generalizado for o modelo, melhor será para predizer os valores de  $y$  para novos conjuntos de dados.

#### 2.1.1.1 Algoritmo Classification And Regression Trees - CART

O algoritmo Classification And Regression Trees - CART constrói modelos de previsão a partir de dados de treinamento onde seus resultados podem ser representados em uma árvore de decisão. A árvore de decisão é uma ferramenta que dá suporte à decisão utilizando como modelo um fluxograma semelhante a uma árvore, onde a cada nó interno é feito um teste para tomada de decisão, e tendo como resposta sim ou não (a exemplo da figura 2), permitindo uma abordagem do problema de forma estruturada e sistemática até chegar a uma conclusão lógica. “Uma árvore de decisão alcança sua decisão executando uma sequência de testes” (RUSSEL; NORVIG, 2013, p. 811)

O CART pode se tornar uma árvore de classificação ou também uma árvore de regressão, o que definirá o tipo de árvore é o valor do atributo classe, se categórico ou contínuo. Por exemplo, em um conjunto de dados de um paciente onde tenta prever se o mesmo possuirá câncer, ou não, a classe seria “Terá Câncer” ou “Não terá Câncer”, podendo esse atributo assumir duas categorias (classes), e assumindo uma árvore de classificação. Na regressão o atributo classe pode assumir um valor contínuo tornando-se uma árvore de regressão, onde poderá prever valores numéricos como: período de tempo de internação do paciente, preço de uma cirurgia, temperatura do paciente ou quantidade de água ingerida.

No caso de não ser probabilístico o grau de confiança em seu modelo de predição será



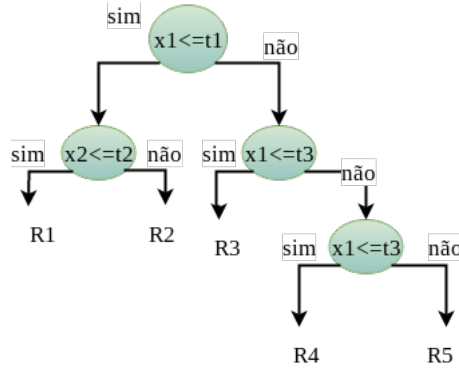


Figura 2 – Exemplo de Fluxograma de árvore: R1 a R5 são as folhas relacionadas de acordo com as respostas sim ou não dos nós

embasada em respostas semelhantes em outras circunstâncias antes analisadas, e utiliza uma técnica como partição recursiva binária, no qual cada nó pai é sempre decomposto em dois nós filhos, e cada nó filho irá ser tratado posteriormente no processo como nó pai.

De acordo com (YOHANNES; WEBB, 1999; RAIMUNDO; MATTOS; WALESKA, 2008), existem três componentes importantes na construção de uma árvore de decisão:

- Um conjunto de perguntas que servirá de base para fazer uma divisão;
- Regras de divisão para julgar o quanto é boa esta divisão;
- Regras para atribuir uma classe a cada nó;

Na divisão inicial será atribuído ao nó pai uma questão onde dependendo da resposta (sim ou não - a exemplo da figura 2) os registros irão para nó filho esquerdo ou direito, e após, será realizado o teste do ponto de divisão. O CART percorrerá todos os atributos e construirá uma árvore de decisão baseado no melhor ponto de divisão, uma vez que são testados todos os atributos como potencial divisor. Após a escolha do melhor ponto de divisão do nó faz-se a atribuição de uma classe para o nó, e após essa etapa, o novo nó filho passa a ser nó pai, refazendo os mesmos passos anteriores para a divisão desse nó.

Logo na equação 2.1 pode-se verificar como o CART faz a escolha da divisão dos nós em função da regra Gini de Impureza (BREIMAN et al., 1984). É definido o grau de pureza variando de 0 (zero) a 1 (um), portanto quando o nó tende um resultado de índice Gini aproximando-se de 1 (um), maior a impureza do nó, e o inverso maior a pureza.

$$Gini(S) = 1 - \sum [p(j/t)]^2 \quad (2.1)$$

Onde:  $p(j/t)$  é probabilidade a priori da classe  $j$  se formar no nó  $t$ , e  $S$  é um conjunto de dados que contém exemplos de  $n$  classes.

A equação 2.1 mede a impureza do conjunto  $S$ , e caso todos os dados forem da mesma classe (dados puros), o resultado da equação seria  $1 - 1 = 0$ . Já a equação 2.2 tem como finalidade a escolha da divisão do nó (conjunto  $S$ ), onde é medida a impureza da divisão do nó pai com os nós filhos, e para isso, contém a média ponderada de cada índice do subgrupo formado por essa divisão de  $S$ . Então o menor valor encontrado em  $Gini_{split}(S)$  será o escolhido para dividir o nó.

$$Gini_{split}(S) = \frac{S_l}{S} gini(S_l) + \frac{S_r}{S} gini(S_r) \quad (2.2)$$

Onde:

- $S$  conjunto de dados que contém exemplos de  $n$  classes;
- $S_l$  subconjunto esquerdo de  $S$ ;
- $S_r$  subconjunto direito de  $S$ ;

Para a escolha da variável e ponto de divisão necessário do nó, terão que ser aplicados testes em todos os atributos através das equações 2.1 e 2.2, e após o envolvimento de todos os atributos será escolhido o nó com menor valor  $Gini_{split}(S)$ .

No procedimento da divisão do nó em dois subconjuntos, o atributo poderá conter valores contínuos ou categóricos. Nas duas opções serão aplicados a equação 2.2 em todos os valores, e escolhido o melhor ponto de divisão. No caso de serem valores contínuos, simplesmente após o valor ser o escolhido para a divisão, esta divisão será: menor igual (ramo da esquerda) ou maior (ramo da direita) que o valor escolhido. Em outra situação, sendo as variáveis categóricas - por exemplo X, Y e Z, terão que ser testadas, dentre todas as possibilidades, qual melhor divisão entre elas, e como é uma divisão binária, o nó não poderá ser dividido em 3 (três) ramos X, Y e Z, e sim, em grupos de dois como: {X} e {Y,Z}, {Y} e {X,Z} ou {Z} e {X,Y}.

#### 2.1.1.2 Algoritmo Naive Bayes

O algoritmo Naive Bayes é um modelo probabilístico de aprendizado que pode ser calculado diretamente entre seus dados de treinamento. Depois de calculado, o modelo pode ser utilizado para fazer previsões de novos dados através do teorema de Bayes. “O teorema de Bayes fornece uma maneira de calcular a probabilidade de uma hipótese com base em sua probabilidade anterior, as probabilidades de observar vários dados, dadas as hipóteses, e os dados observados em si” (MITCHELL, 1997, p. 156).

Esse teorema utiliza uma teoria estatística e probabilística para previsão de acontecimento de um evento, sendo este evento relacionado a condição da probabilidade de ocorrência anteriores do mesmo, portanto, é nesse seguimento que o algoritmo Naive Bayes funciona, criando classificadores probabilísticos baseados no teorema de Bayes.

Pode-se citar como exemplo desse evento, a descoberta do câncer em uma pessoa, pois se tal doença estiver relacionada ao sexo, então, utilizando o teorema de Bayes, o sexo de uma pessoa pode ser utilizada para dar maior precisão a probabilidade de câncer, em vez de fazer uma avaliação de probabilidade sem a utilização do sexo da pessoa.

O Naive Bayes utiliza uma técnica de independência dos atributos, onde cada variável de entrada não depende de recursos de outras. Essa independência condicionada entre os atributos, os quais nem sempre ocorrem nos problemas reais, acabou deixando conhecida por Bayes ingênuo, ou Naive Bayes.

Em (RUSSEL; NORVIG, 2013) a equação 2.3 mostra a relação  $P(causa/efeito)$  onde o efeito é evidência de alguma causa desconhecida, e quer se determinar a causa.

$$P(causa|efeito) = \frac{P(efeito|causa)P(causa)}{P(efeito)} \quad (2.3)$$

Naive Bayes como classificador estatístico possui um modelo de simples construção, e ficou conhecido por ter bons resultados em relação a algoritmos mais sofisticados, mesmo trabalhando com grandes quantidades de dado, e possui uma característica de agrupar objetos de uma certa classe em razão da probabilidade do objeto pertencer a esta classe.

$$P(c/x) = \frac{P(x/c)P(c)}{P(x)} \quad (2.4)$$

$$P(c/x) = P(x_1|c) * P(x_2|c) * ... * P(x_n|c) * P(c) \quad (2.5)$$

- $P(c/x)$  probabilidade posterior da classe  $c$ , alvo dada preditor  $x$ , atributos.
- $P(c)$  é a probabilidade original da classe.
- $P(x|c)$  é a probabilidade que representa a probabilidade de preditor dada a classe.
- $P(x)$  é a probabilidade original do preditor.

A utilização do algoritmo Naive Bayes já é bem difundida e está presente em vários trabalhos, como classificação de textos, filtro de SPAM, analisador de sentimentos, entre outros (MADUREIRA, 2017; LUCCA et al., 2013; WU et al., 2008; MCCALLUM; NIGAM, 1997), entretanto, mesmo atingido boa popularidade o algoritmo se utiliza da suposição de ter preditores independentes, e isso não acontece muito na vida real, pois acaba sendo difícil ter uma amostra de dados que sejam inteiramente independentes.

A tabela 1 é uma amostra de dados, onde é possível aplicar o funcionamento do Naiva Bayes através da equação 2.5, sendo esta tabela composta por 4 (quatro) características (Aspecto, Temperatura, Umidade e Vento) e pela coluna Decisão, representando a classe.

	Aspecto	Temperatura	Umidade	Vento	Decisão
1	Sol	Quente	Alta	Fraco	Não
2	Sol	Quente	Alta	Forte	Não
3	Nublado	Quente	Alta	Fraco	Sim
4	Chuva	Agradável	Alta	Fraco	Sim
5	Chuva	Fria	Normal	Fraco	Sim
6	Chuva	Fria	Normal	Forte	Não
7	Nublado	Fria	Normal	Forte	Sim
8	Sol	Agradável	Alta	Fraco	Não
9	Sol	Fria	Normal	Fraco	Sim
10	Chuva	Agradável	Normal	Fraco	Sim
11	Sol	Agradável	Normal	Forte	Sim
12	Nublado	Agradável	Alta	Forte	Sim
13	Nublado	Quente	Alta	Fraco	Sim
14	Chuva	Agradável	Alta	Forte	Não

Tabela 1 – Base de exemplo onde exhibe através do atributo Decisão, se existe, ou não condição de jogo perante as outras características (Aspecto, temperatura, Umidade e Vento)

Então cada registro assume uma condição de jogo “Sim” ou “Não” - por exemplo, na linha 2 (dois) faz “Sol”, é “Quente”, possui umidade “Alta” e vento “Forte”, e contém no atributo classe a não possibilidade de jogo (Decisão = Não). Pode-se com as informações dessa amostra prever algumas possibilidades de jogo, dependendo de como estão dispostos os valores dessas características. Para exemplificar melhor, as seguintes condições são para saber se há possibilidade de jogo, “Sim”:  $P(\text{Jogar}=\text{sim}|\text{Aspecto}=\text{sol}, \text{Temperatura}=\text{fria}, \text{Umidade}=\text{alta} \text{ e } \text{Vento}=\text{forte})$ . Para obter a resposta será necessário a aplicação na equação 2.5.

$$\begin{aligned}
&= \frac{P(\text{Sol}|\text{Sim}) * P(\text{Fria}|\text{Sim}) * P(\text{Alta}|\text{Sim}) * P(\text{Forte}|\text{Sim}) * P(\text{Sim})}{P(\text{Sol}) * P(\text{Fria}) * P(\text{Alta}) * P(\text{Forte})} \\
&= \frac{\frac{2}{9} * \frac{3}{9} * \frac{3}{9} * \frac{3}{9} * \frac{9}{14}}{\frac{5}{14} * \frac{4}{14} * \frac{7}{14} * \frac{6}{14}} \\
&= \frac{0,0053}{0,02186} \\
&= 0,242
\end{aligned} \tag{2.6}$$

O resultado (0,242) mostra uma probabilidade aproximada de 20% de chance de acontecer o jogo, de acordo com as características apresentadas, implicando aproximadamente 80% de não acontecer, portanto, seguindo esse resultado pode-se afirmar que a previsão é de não ocorrer o jogo.

### 2.1.1.3 Algoritmo k-Nearest Neighbor - KNN

O K-Nearest Neighbor - KNN é um algoritmo de classificação simples, onde os objetos são classificados através de um conjunto de treinamento que estão próximos no espaço de características. Uma vez que seja necessário definir qual a classificação de um objeto, será averiguado quais são os exemplos mais próximos, determinado por uma distância, e assim definirá através desses elementos próximos, qual sua classificação.

Na execução do KNN algumas considerações são importantes, como a definição da métrica entre os elementos e o número que a variável K assumirá, ademais, segue alguns passos: i) o algoritmo funcionará calculando a distância entre todos os exemplos próximos do elemento a classificar; ii) será identificado os K vizinhos mais próximos; iii) e através do número de K será determinado a mesma classe do vizinho para classificação do elemento.

Na figura 3 pode-se observar comportamento do algoritmo, onde um objeto está próximo de outros objetos vizinhos, e o número de vizinhos é determinado pelo K.

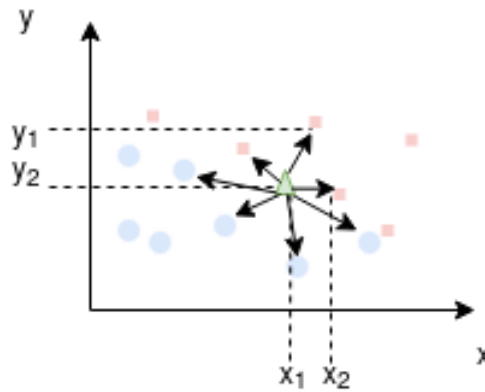


Figura 3 – Exemplo do funcionamento do algoritmo KNN

Neste exemplo da figura 3 um novo objeto precisa ser classificado, e na amostra de dados distribuída no plano cartesiano, o algoritmo KNN prevê sua classificação de acordo com objetos que estão próximos a ele. Na figura 3 existem objetos “círculos”, “quadrados”, e um novo objeto triângulo que seria o novo objeto a ser classificado.

O algoritmo KNN calcula a distância do objeto “triângulo” para com os outros objetos utilizando a distância euclidiana (LACHI; ROCHA, 2005), conforme equação 2.7 que está na forma bidimensional de acordo com o exemplo apresentado na figura 3.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2.7)$$

Onde:

- $d$  resultado da distância euclidiana entre dois objetos no plano cartesiano;
- $x_2 - x_1$  distância no eixo  $x$  entre objetos;

- $y_2 - y_1$  distância no eixo  $y$  entre objetos;

Ao calcular todas as distâncias o algoritmo irá ordenar estes resultados de forma crescente, de acordo com a tabela 2, e depois dependendo do valor de  $K$  será determinado a classe do objeto. A tabela de exemplo é formada por 3 (três) colunas, onde  $K$  significa o número de vizinhos, seguida da distância e também qual o tipo de objeto (quadrado ou círculo).






K	Distância	Classe
1	0,11	
2	0,29	
3	1,10	
4	1,40	
5	1,55	

Tabela 2 – Tabela em ordem crescente das distâncias euclidianas encontradas do objeto a que se deseja classificar para os outros objetos da amostra, de acordo com as setas da figura 3

Para definir qual a classe do objeto “triângulo” assumirá, será necessário saber qual valor de  $K$ , pois o maior número de ocorrências de uma determinada classe, será a classe eleita para o novo objeto - por exemplo, utilizando a tabela 2, com  $K = 1$  a classe deste registro é “círculo”, dado que o número de ocorrências é CÍRCULO=1 e QUADRADO=0, então, caso  $K = 1$ , o número de ocorrência do “círculo” é maior que “quadrado”. No caso de  $K = 2$ , aparece uma ocorrência de “círculo” e uma ocorrência de “quadrado”, totalizando CÍRCULO=1 e QUADRADO=1, nesse caso não fica definido qual a classe, pois cada classe possui o mesmo número de ocorrências. No caso de  $K = 3$  a contagem de ocorrências em cada classe totaliza CÍRCULO=1 e QUADRADO=2, e desta vez, a classe que possui maior número de ocorrências é a escolhida para o novo objeto, “quadrado”, e assim acontece sucessivamente para os outros valores  $K$  assumir. Cada vez que o valor de  $K$  aumenta é feita a soma de ocorrências de cada classe, sendo a classe elegida a que mais ocorrências houver.

Ao escolhe um valor par de  $K$ , poderá haver um empate no número de ocorrências das classes, essa situação fica clara no exemplo acima, tabela 2, quando  $K = 2$  (CÍRCULO=1 e QUADRADO=1) e  $K = 4$  (CÍRCULO=2 e QUADRADO=2), mas para não acontecer um valor de empate é necessário escolher sempre um valor  $K$  ímpar, e este parâmetro pode ser definido ao executar o algoritmo.

### 2.1.2 Aprendizado Não-Supervisionado

Outro cenário de aprendizado de máquina é o aprendizado não-supervisionado, onde não existe uma tentativa de se encontrar uma função que se aproxime da real, logo

porque os registros não são classificados, visto que o conjunto de treinamento não possui informação da saída sobre determinada entrada. Desta forma os algoritmos procuram algum grau de similaridade entre os registros e tentam agrupá-los de forma a ter algum sentido deles estarem juntos.

Quando o algoritmo encontra dados com mesma similaridade ele os agrupa formando *clusters*. Os números de *clusters* encontrados dependerá do funcionamento, técnica, configuração dos algoritmos e também do grau de dissimilaridade entre elementos de grupos diferentes. Segundo (BARBER, 2011) não existe uma variável classe no aprendizado não-supervisionado, então, o maior interesse, seria em uma perspectiva probabilística de distribuição  $p(x)$  de um determinado conjunto de dados  $D = \{x_n, n = 1, \dots, N\}$ . Mesmo não possuindo rótulos (classes), novos dados inseridos são submetidos aos algoritmos não-supervisionados, e esses algoritmos são capazes de encontrar padrões nos atributos em um conjunto de treinamento, conseguindo inferir sobre os dados de testes, classificando-os em algum grupo.

Cada algoritmo utilizará alguma característica para determinar grupos diferentes na base de dados, no exemplo da figura 4, fica visível a existência de agrupamentos diferentes (h1, h2, h3, h4).

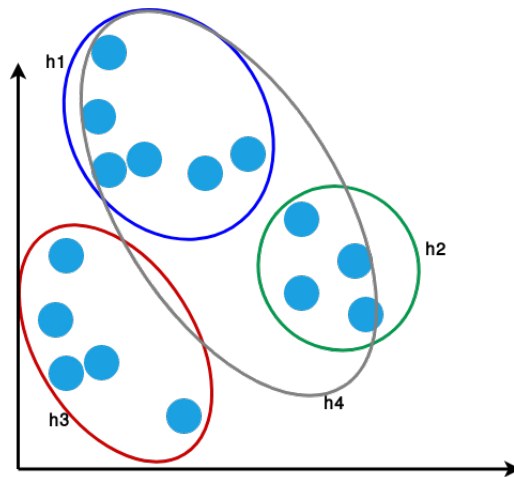


Figura 4 – Exemplos de técnicas diferentes utilizada por algoritmos para dividir em grupos

O que é apresentado na figura 4 são situações de agrupamentos em um conjunto distribuído bidimensional. Dependendo do algoritmo, ou mesmo da configuração, é possível ter 3 (três) grupos formados por h1, h2 e h3, ou 2 (dois) grupos formados por h3 e h4, ou mesmo, um grupo só pela união de h3 e h4.

## 2.2 Discretização

O método de discretização faz a conversão de valores contínuos em valores discretos. A partir de um atributo com valores contínuos, a discretização cria um ponto inicial e final definindo um intervalo e designando uma faixa para cada intervalo. Assim, ao invés de valores contínuos, novos conteúdos representando as faixas de valores.

Segundo alguns autores, a discretização melhora a precisão e deixa um modelo mais rápido em seu conjunto de treinamento (CATLETT, 1991; HWANG; LI, 2002). De acordo com (KOTSIANTIS; KANELLOPOULOS, 2006; DOUGHERTY; KOHAVI; SAHAMI, 1995) os métodos de discretização mais comumente utilizados, no âmbito dos métodos não-supervisionados, são os de Discretização por Larguras Iguais (do inglês: Equal Width Discretization - EWD) e Discretização por Frequências Iguais (do inglês: Equal Frequency Discretization - EFD).

### 2.2.1 Equal Weight Discretization - EWD

O método de Discretização por Larguras Iguais (Equal Weight Discretization - EWD) faz a discretização de um intervalo, entre valores contínuos, dividindo através de um ponto de corte as faixas de tamanhos iguais. Logo se existir um intervalo com valores contínuos  $[a, b]$ , e deseja particionar em  $R$  faixas de tamanhos iguais serão necessários  $R - 1$  pontos de corte (figura 5).

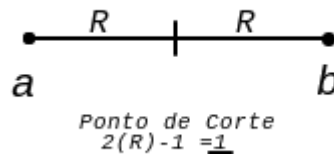


Figura 5 – Ponto de Corte ( $R-1$ );  $2(R)$  significa o valor de  $R=2$ ;  $2 - 1 = 1$

Para exemplificar, a figura 5 exibe uma faixa com início em “[a”, e final “[b]”, e pretende obter a divisão do intervalo  $[a, b]$  em 2 (duas) faixas iguais ( $R = 2$ ), então, utilizando a regra (Número\_De\_Faixas - 1 =  $2-1=1$ ), obtendo sempre, o número de faixas que deseja particionar menos 1 (um).

Para haver o ponto de corte, terá que ser feito primeiro a ordenação dos dados, e logo após, definir a largura de cada faixa  $r_1, \dots, r_R$ . O cálculo realizado na equação 2.8 para encontrar  $w$ , é a diferença entre os limites superior e inferior do intervalo, dividido pela quantidade de faixas ( $R$ ).

$$w = \frac{b - a}{R} \quad (2.8)$$



De acordo com a figura 2.9 a variável  $w$  delimita o tamanho das faixas de valores e determina os pontos de corte  $(c_1, \dots, c_{R-1})$ . O primeiro ponto de corte,  $c_1$ , é obtido através da soma do limite inferior  $a$  com a tamanho de  $w$ , e os pontos de corte seguintes são calculados pela soma do ponto de corte anterior com  $w$ .

$$c_i = \begin{cases} a + w, & \text{se } i = 1 \\ c_{i-1} + w, & \text{caso contrário} \end{cases} \quad (2.9)$$

Seguindo a figura 6, o valor da faixa do intervalo  $[a, c_1]$  será o valor discreto igual ao índice de sua faixa, nesse caso  $r_1$ , então, o valor na faixa  $r_1$  é representado por  $1(um)$ , pois  $i = 1$ . No caso do intervalo  $[c_1, c_2]$  definido na faixa  $r_2$  é representado pelo valor discreto  $2(dois)$ , e conseqüentemente o valor que se encontra em uma faixa qualquer  $r_i$  será representado por  $i$ .

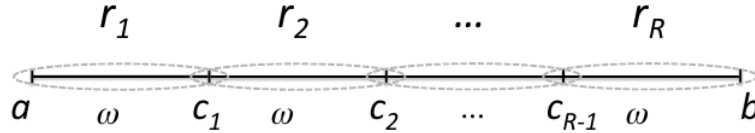


Figura 6 – Discretização EWD. Figura baseada em (LOPES; MACHADO; RABELO, 2016)

A tabela 3 contém uma amostra de valores dividido em duas colunas, onde a primeira coluna significa o número da linha e a segunda coluna o valor propriamente dito. Essa tabela servirá para exemplificar o que foi dito neste método de discretização, e pode ser vista como um vetor de 150 (cento e cinquenta) posições, e dependendo do número de faixas poderá ser dividido em vários pedaços, onde cada pedaço seria uma faixa.

Para este exemplo descrito é definido em 3 (três) o número de faixas ( $R = 3$ ), e aplicado a equação 2.8 na tabela 3, para encontrar a largura (fixa) da faixa, pois o método de discretização EWD, mantém faixas com mesmo tamanho. O intervalo  $[a, b]$  seriam o limite inferior, menor valor ( $a = 4,3$ ) e limite superior, maior valor ( $b = 7.9$ ) respectivamente da tabela de amostra, portanto, logo que calculado o valor da largura  $w$  é utilizado a regra da equação 2.9 para encontrar os pontos de corte de cada faixa.

Utilizando a equação 2.8, encontra-se  $w = 1.2$ , portanto, uma vez em posse da largura, e sendo o primeiro ponto de corte ( $i = 1$ ), simplesmente utiliza-se equação 2.9 para encontrar o primeiro ponto de corte  $a + w = 4.3 + 1.2 = 5.5$ , que está como asterisco na posição  $c_1 = 5.5$ . Os asteriscos na figura 7 delimitam os pontos de cortes, e os pontinhos são todos os valores dispostos nas faixas definidos na tabela 3.

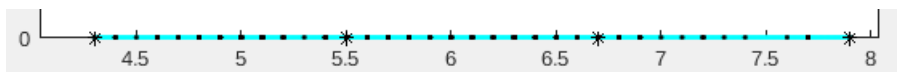


Figura 7 – Discretização EWD de acordo com a amostra da tabela 3

no.	Valor	no.	Valor	no.	Valor	no.	Valor	no.	Valor	no.	Valor
1	5,10	26	5,00	51	7,00	76	6,60	101	6,30	126	7,20
2	4,90	27	5,00	52	6,40	77	6,80	102	5,80	127	6,20
3	4,70	28	5,20	53	6,90	78	6,70	103	7,10	128	6,10
4	4,60	29	5,20	54	5,50	79	6,00	104	6,30	129	6,40
5	5,00	30	4,70	55	6,50	80	5,70	105	6,50	130	7,20
6	5,40	31	4,80	56	5,70	81	5,50	106	7,60	131	7,40
7	4,60	32	5,40	57	6,30	82	5,50	107	4,90	132	7,90
8	5,00	33	5,20	58	4,90	83	5,80	108	7,30	133	6,40
9	4,40	34	5,50	59	6,60	84	6,00	109	6,70	134	6,30
10	4,90	35	4,90	60	5,20	85	5,40	110	7,20	135	6,10
11	5,40	36	5,00	61	5,00	86	6,00	111	6,50	136	7,70
12	4,80	37	5,50	62	5,90	87	6,70	112	6,40	137	6,30
13	4,80	38	4,90	63	6,00	88	6,30	113	6,80	138	6,40
14	4,30	39	4,40	64	6,10	89	5,60	114	5,70	139	6,00
15	5,80	40	5,10	65	5,60	90	5,50	115	5,80	140	6,90
16	5,70	41	5,00	66	6,70	91	5,50	116	6,40	141	6,70
17	5,40	42	4,50	67	5,60	92	6,10	117	6,50	142	6,90
18	5,10	43	4,40	68	5,80	93	5,80	118	7,70	143	5,80
19	5,70	44	5,00	69	6,20	94	5,00	119	7,70	144	6,80
20	5,10	45	5,10	70	5,60	95	5,60	120	6,00	145	6,70
21	5,40	46	4,80	71	5,90	96	5,70	121	6,90	146	6,70
22	5,10	47	5,10	72	6,10	97	5,70	122	5,60	147	6,30
23	4,60	48	4,60	73	6,30	98	6,20	123	7,70	148	6,50
24	5,10	49	5,30	74	6,10	99	5,10	124	6,30	149	6,20
25	4,80	50	5,00	75	6,40	100	5,70	125	6,70	150	5,90

Tabela 3 – Amostra de dados para exemplificar a discretização EWD e EFD

A partir dos cálculos é definido as seguintes faixas de tamanhos iguais:

- Faixa 1 -  $[4.3, 5.5]$
- Faixa 2 -  $]5.5, 6.7]$
- Faixa 3 -  $]6.7, 7.9]$

## 2.2.2 Discretização por Frequência Iguais - EFD

Esse outro método de discretização já possui uma abordagem diferente do EWD, pois a idéia é manter a quantidade de elementos distintos, entre os pontos de corte, com o mesmo número. Dado um intervalo  $[a, b]$  o número de faixas  $R$  e a quantidade de valores distintos  $\xi$ , onde  $\xi \geq R$  o método EFD irá segmentar em  $R$  faixas de valores que possuem a mesma quantidade de elementos distintos  $\lambda$ . Então serão realizados  $R - 1$  pontos de corte gerando  $R$  faixas de valores,  $(r_1, \dots, r_R)$ , com a mesma quantidade de elementos distintos  $\lambda$ . Para encontrar  $\lambda$  calcula-se o valor inteiro da divisão entre a quantidade de elementos

distintos  $\xi$  pela quantidade de faixas de valores  $R$ , obtendo o número de elementos da faixa (equação 2.10).

$$\lambda = \frac{\xi}{R} \quad (2.10)$$

Uma observação nesse método é a ocorrência de uma má distribuição de valores entre as faixas, portanto, caso haja um número significativo de valores repetidos de um atributo, isso causa um desequilíbrio na distribuição dos elementos dentro da faixa. Essa situação reflete em faixas com muitos valores e outras sem nenhuma.

Uma vez no intervalo  $[a, b]$  de elementos ordenados e calculado  $\lambda$  contendo  $R$  elementos  $v_{[R]}$ , pode-se determinar os pontos de corte  $(c_1, \dots, c_{R-1})$  que são os delimitadores das faixas. Cada ponto de corte  $c_i$  pode ser calculado por  $v_{i\lambda}$  – *ésimo* elemento (equação 2.11).

$$c_i = v_{[i\lambda]} \quad (2.11)$$

Igual ao que aconteceu no método EWD, o valor que estiver no intervalo  $[a, c_1]$  terá seu valor associado a um valor discreto igual ao índice  $i$  de sua faixa  $r_i$  conforme figura 8. Então, caso o valor esteja na faixa  $r_2$  ele passará a ter o valor de seu índice  $i$  igual a 2(*dois*). De maneira consecutiva os valores que estiverem na faixa  $r_3 = ]c_2, c_3]$  terão valor 3(*três*). Uma outra observação desse método é que diferente do EWD, os intervalos podem assumir faixas com tamanhos diferentes.

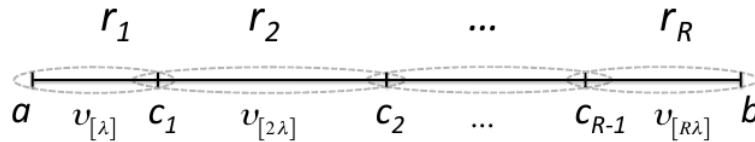


Figura 8 – Discretização EFD. Figura baseada em (LOPES; MACHADO; RABELO, 2016)

Neste exemplo com o método EFD, exibido o resultado na figura 9, a tabela 3 também é utilizada na disposição dos valores em cada faixa, bem como o cálculo de  $\lambda$  (equação 2.10), que é a divisão do total de elementos distintos ( $\xi$ ) pelo o número de faixas  $R = 3$ . Após tal cálculo, é realizado a ordenação dos valores, e uma vez os valores ordenados, soma-se o valor mínimo com  $\lambda$ , encontrando  $c_1$  como primeiro ponto de corte, e assim sucessivamente, para os outros pontos de corte ( $c_2 = c_1 + \lambda$ ), até  $R - 1$  pontos de corte. Na figura 9 pode-se perceber como a distribuição dos valores da amostra se comportam no método EFD. Os asteriscos delimitam o início e fim de cada faixa de valor, e os pontos são os valores propriamente ditos.

O número de elementos distinto na amostra de dados da tabela 3 é de 35 (trinta e cinco) elementos, e dividindo por  $R = 3$ , que é o número de faixas, encontrará  $\lambda = 11$ .



Figura 9 – Discretização EFD de acordo com a amostra da tabela 3

Na lista de números distintos o 11º (décimo primeiro) elemento a partir de  $a$  (menor valor da amostra) será o primeiro ponto de corte ( $c_1$ ). Então todos os valores de 4.3 a 5.3 (identificados por asterisco) fazem parte da primeira faixa.

A seguir é definido as faixas, onde percebe-se que são de tamanhos diferentes ao do método EWD, pois há valores repetidos:

- Faixa 1 -  $[4.3, 5.3]$
- Faixa 2 -  $]5.3, 6.4]$
- Faixa 3 -  $]6.4, 7.9]$

## 2.3 Trabalhos Correlatos

Esta seção propõe relacionar outros trabalhos servindo de complemento teórico envolvendo assuntos como agrupamentos de dados, aprendizado de máquina, classificação e rotulação de dados.

Em (JIRASIRILERD; TANGTISANON, 2018) os autores fazem uso de rotulação automática de textos em Tailandeses retirados de notícias da internet. A utilização é para classificar esses textos com a devida categoria (TI, entretenimento, astronomia, *etc*). Essa pesquisa utiliza para rotulação, vetor de representação de documentos, e na separação das palavras Tailandesas é aplicado algoritmo de Convolutional Neural Network (CNN). Então o modelo faz: i) Conversão de parágrafos e palavras para os vetores utilizando a técnica de representação distribuída. ii) extrai vetores com paragrafos semelhantes. iii) cria um vetor de características. iv) extrai vetores com palavras semelhantes. v) rotula.

Em (YEGANOVA; COMEAU; WILBUR, 2010) é utilizado a ideia de dados naturalmente rotulados em uma abordagem que faz detecção e identificação de abreviações na literatura biomédica utilizando aprendizado de máquina supervisionado. Através dos textos é realizado uma extração de estruturas textuais (formas curtas, formas longas, formas curtas pontenciais e formas longas pontenciais), que quando são extraídos naturalmente em pares *i.g.* (forma curta - forma forma longa), (formas curtas pontenciais - formas longas pontenciais) são tratados como exemplos positivos.

Nesse artigo (CHEN; TSENG; LIANG, 2011) propõe a criação de clusters a partir de textos e documentos através de uma abordagem eficaz de agrupamento de documentos, *Fuzzy Frequent Itemset-based Document Clustering* (F2IDC) que combina a mineração de regras de associação *fuzzy* com conhecimento da WordNet. A WordNet é um banco de

dados léxico em inglês que agrupa palavras (substantivos, verbos, adjetivos, advérbios) em conjunto de sinônimos, e tem com isso o objetivo de melhorar a qualidade dos grupos através dos relacionamentos semânticos.

Nos trabalhos acima citados (JIRASIRILERD; TANGTISANON, 2018; YEGANOVA; COMEAU; WILBUR, 2010; CHEN; TSENG; LIANG, 2011) acontecem processos de agrupamentos e rotulação em textos, sendo um tema bastante estudado, e diferente desta dissertação, que utiliza o conceito de rotulação de dados, no contexto do significado ao grupo formado.

O artigo em (GAN et al., 2013) propõe aprendizado de máquina semi-supervisionado que combina agrupamento e classificação com os devidos algoritmos, *Fuzzy C-Means* e *SVM* respectivamente. A pesquisa utiliza dados rotulados e não rotulados, apostando na análise do *cluster* como diferencial, para compensar a limitação de dados não rotulados, e através do conhecimento adquirido melhorar o treinamento do classificador. Essa pesquisa (GAN et al., 2013) diferente desta dissertação, não envolve a interpretação dos grupos após sua formação.

Outro trabalho de agrupamento de dados pode ser visto em (SUN; YOSHIDA; LIANG, 2011), onde ele aborda o problema de agrupamento de dados e propõe um algoritmo híbrido com o *support vector cluster* e *K-Means*, ambos de aprendizagem não-supervisionada. Em uma primeira etapa é utilizado uma abordagem do *support vector cluster* com objetivo de identificar os *outliers* e os pontos sobrepostos, e na segunda etapa, obtem-se os núcleos removendo os *outliers* e os pontos sobrepostos e aplicando o *K-Means* nos núcleos, para obter o conjunto de dados em clusters. Foi utilizado algumas variáveis de extrema importância para conclusão do trabalho, e essas variáveis foram configuradas empiricamente em fases diferentes, a fim de obter bons resultados nos agrupamentos de dados. O estudo desse trabalho é interessante no conceito da formação de bons grupos de dados, pois em rotulação de grupos, há uma correlação de grupos bem definidos e bons rótulos.

Outro artigo, o autor (IWAMURA; TSUKADA; KISE, 2013) faz rotulação automática de textos de cenas em uma base de dados, são textos que encontram-se em uma imagem - por exemplo, uma imagem da fachada de uma casa que possui um número de identificação. O modelo utiliza imagens contendo caracteres, segmenta essas imagens, e após, classifica o caractere através de uma base de dados com várias amostras armazenadas, fazendo seu reconhecimento. Fica claro que esse artigo faz rotulação, mas não da mesma forma definida nesta dissertação, ao qual faz uso de aprendizado de máquina a fim de rotular grupos, e dando significado a eles.

O trabalho (COSTA et al., 2016) utiliza classificação não-supervisionada, mas possui uma característica diferenciada por utilizar dados *online*. O método faz uso da aprendizagem a partir de uma base de regras vazias com o processamento das amostras



uma técnica de aprendizado semi-supervisionada, onde visa fazer rotulação de dados através de uma pequena amostra rotulada. Logo em (LIMA; MACHADO; LOPES, 2015) é aplicado rotulação a uma base de dados de uma rede social chamada de Scientia.Net, com o objetivo na criação de grupos e identificação dos atributos que podem ser importantes ao ponto de representar estes grupos, chamando-os de rótulos. O Scientia.Net é uma rede social para cientistas com propósito do compartilhamento de suas pesquisas e criação de seus perfis facilitando o contato e troca de informações no ambiente acadêmico entre pesquisadores. Nesse artigo o autor utilizou o modelo de (LOPES; MACHADO; RABELO, 2016) para rotulação de dados com os mesmos algoritmos, o algoritmo *k-means* para agrupamento, e logo na segunda parte a utilização do algoritmo supervisionado, *Artificial Neural Networks* - ANN. Diferente dos trabalhos (LIMA, 2015; LIMA; MACHADO; LOPES, 2015), este texto tem o foco em testes somente nos algoritmos supervisionados nos grupos já formados, de acordo com a origem da base de dados, a fim de gerar rótulos, e assim aferir suas acurácias para cada um desses algoritmos testados.

Outra pesquisa (FILHO; MACHADO; LIRA, 2015) aborda o mesmo problema de rotulação, mas com atuação diferente, pois o modelo utiliza o algoritmo não-supervisionado *Fuzzy C-Means* para composição dos grupos, onde o número de grupos é fornecido na inicialização do algoritmo, e também na definição das faixas de valores de atributos de cada atributo rótulo do grupo formado. A diferença entre esta dissertação e esse artigo de Filho, Machado e Lira (2015) é o modelo de resolução que utiliza somente o algoritmo *Fuzzy C-Means* para criar os grupos e definir as faixas de valores de cada atributo rótulo, e também a ausência da técnica de discretização dos valores dos dados para obter os rótulos.

Outro autor (IMPERES, 2018) em seu trabalho utiliza uma proposta de rotulação semelhante a outra pesquisa (FILHO; MACHADO; LIRA, 2015), mas com outro algoritmo, *K-means* não-supervisionado baseado em distância. Esse modelo é constituído de duas etapas, sendo a primeira, a transformação da distância gerada pelo *K-means* em GP, e logo após, na segunda etapa é realizado a rotulação de dados de acordo com a tabela gerada na primeira etapa, onde é feita várias iterações até encontrar faixas únicas de valores para cada atributo rótulo. Ao se comparar esta dissertação com a pesquisa desse autor (IMPERES, 2018) percebe-se que também não há um processo de discretização de valores dos atributos, e também não há aplicação de dois algoritmos de aprendizado.

Outro trabalho de rotulação (ARAÚJO, 2018), defende que uma etapa de fundamental importância para se ter bons resultados na rotulação de grupos de dados se dá na clusterização. Portanto, quanto mais eficiente for a técnica de agrupamento de dados utilizada, maior será acurácia dos grupos encontrados. A partir do que foi dito, o autor utiliza em conjunto o DAMICORE e DAMICORE-2 no método de rotulação automática criar grupos. DAMICORE é um método de detecção de correlação de dados e tem como característica a não informação do número de *clusters* ao qual o algoritmo é aplicado.

O modelo é dividido em cinco etapas até obter os rótulos. Nas etapas I e II preparam os dados e ajuda a medir a similaridade dos elementos, oferecendo uma maior precisão, e contribuindo para criação de grupos mais significativos. Na etapa II ocorre a clusterização, e como não é preciso informar o número de cluster na utilização do DAMICORE, os resultados na etapa III acabam por superar um número razoável de *clusters* para uma melhor compreensão, e em razão disso, a etapa IV de mesclagem, faz a junção de *clusters* para criação de super-*clusters*, que são *clusters* maiores representando um conceito mais geral e de mais fácil entendimento. Por fim, os super-*clusters* são submetidos ao método de rotulação automática na etapa V, permitindo identificar os atributos mais relevantes e suas respectivas faixas de valores. Essa pesquisa mantém o foco na etapa de formação dos clusters, e diferencia do texto desta dissertação exatamente nisso, pois nesta dissertação o foco é nos algoritmos supervisionados utilizados na etapa de rotulação dos grupos de dados.



## 3 Metodologia

O texto a seguir abordará o problema descrito neste trabalho e, logo em seguida, será apresentado um modelo de resolução utilizando algoritmos supervisionados, e mais ao final deste capítulo é realizado um exemplo de rotulação, onde será feita desde a discretização até o encontro dos rótulos para melhor explicar o conteúdo descrito.

### 3.1 Rotulação de Cluster

A abordagem do problema referente a essa proposta de mestrado segue uma linha já pesquisada que seria o **Problema de Rotulação**. Muitas pesquisas realizadas na área de rotulação fazem referência a classificação dos dados e não da rotulação. Ao agrupar um conjunto de elementos por um determinado critério, está havendo uma classificação desses elementos de mesma similaridade, mas pouco se sabe qual é a compreensão desses grupos já classificados, no sentido de, quais os atributos são mais relevantes dentro desses grupos.

A importância do rótulo em um cluster é transparecer a compreensão do cluster formado, visto que, uma vez os clusters já agrupados não fica claro o critério de criação desses grupos. Para o observador é interessante existir um rótulo de um grupo oferecendo elementos que possam ajudar em alguma tomada de decisão em razão de seu significado, ou seja, o rótulo. Dessa forma serão apresentadas 2 (duas) definições que se complementam.

Na definição 2 é expressa formalmente o comportamento dos clusters, e na definição 3 complementa a definição 2 definindo o comportamento do rótulo.

**Definição 2** *Dado um conjunto de clusters  $C = \{c_1, \dots, c_k | K \geq 1\}$ , de modo que cada cluster contém um conjunto de elementos  $c_i = \{\vec{e}_1, \dots, \vec{e}_{n(c_i)} | n^{(c_i)} \geq 1\}$  que podem ser representados por um vetor de atributos definidos em  $\mathbb{R}^m$  e expresso por  $\vec{e}^{c_i} = (a_1, \dots, a_m)$  e ainda que com  $c_i \cap c_{i'} = \emptyset$  com  $1 \leq i, i' \leq K$  e  $i \neq i'$  (Adaptada de (??)).*

- $K$  é o número de clusters;
- $a$  é o atributo
- $c_i$  é o  $i$ -ésimo cluster qualquer;
- $n^{c_i}$  é o número de elementos do cluster  $c_i$ ;
- $\vec{e}_j^{(c_i)}$  se refere ao  $j$ -ésimo elemento (registro na tabela) pertencente ao cluster  $c_i$ ;
- $m$  é o número de atributos da tabela de dados;

A criação do rótulo é a escolha de uma tupla **atributo** e **faixa de valor**, onde o atributo possui o maior valor de correlacionamento entre os outros atributos, e a faixa

escolhida, uma vez com os dados já discretizados, é aquele valor que mais se repete, dentro do atributo rótulo selecionado - por exemplo, um vetor de valores já discretizados <sup>1</sup>,  $\vec{v}_i = \{1, 1, 1, 2, 2, 2, 2, 3, 3\}$ , sendo  $i \leq m$  e  $(\vec{v})$  representando todos os elementos da coluna representada pelo atributo rótulo ( $a$ ). Neste vetor  $(\vec{v})$  o valor que mais se repete é o número 2, então, a **faixa 2** do atributo rótulo, é a escolhida para compor o rótulo. Isto posto, o rótulo é o atributo representado por  $a$  junto com a representação da faixa 2 (dois), e podendo em outra situação, o rótulo de um cluster ser composto por mais de uma tupla: atributo, faixa. (Definição 2).

## 3.2 O Modelo de Resolução

A partir da definição do problema - *Definição 2* - um estudo foi desenvolvido nesta pesquisa, a fim de ser possível realizar rotulação de dados com algoritmos supervisionados com características distintas.

Este modelo de resolução consiste em apresentar como saída um conjunto de rótulos, onde cada rótulo específico é dado por um conjunto de pares de valores, atributo e seus respectivos intervalos, gerados a partir das frequências dos valores repetidos neste intervalo. Segue *Definição 3* formalizando a saída do modelo:

**Definição 3** *Dado um conjunto de rótulos  $R = \{r_{c1}, \dots, r_{ck}\}$ , no qual cada rótulo específico é dado por um conjunto de pares de valores, correspondendo a um vetor com atributo e seu respectivo intervalo,  $r_{ci} = \{(a_1, [p_1, q_1]), \dots, (a_{m(c_i)}, [p_{m(c_i)}, q_{m(c_i)}])\}$  capaz de melhor expressar o cluster  $c_i$  (Adaptada de (LOPES; MACHADO; RABELO, 2016)).*

- $k$  número de rótulos;
- $R$  representa o conjunto de rótulos na saída do modelo;
- $a$  é o atributo
- $c_i$  é o  $i$ -ésimo cluster;
- $r_{ci}$  é o rótulo referente ao cluster  $c_i$ ;
- $[p_{m(c_i)}, q_{m(c_i)}]$  representa o intervalo de valores do atributo  $a_{m(c_i)}$ , onde  $p_{m(c_i)}$  é o limite inferior e  $q_{m(c_i)}$  é o limite superior;
- $m$  é o número de atributos da tabela de dados;

Como apresentado na seção 2.3, o autor (LOPES; MACHADO; RABELO, 2016) foca em rotulação automática de grupos utilizando a estratégia de aprendizagem de máquina supervisionada, com paradigma connexionista, para realizar seu trabalho. Porém, nesta pesquisa, foi aplicado no modelo de resolução três algoritmos supervisionados com

<sup>1</sup> seção 2.2

atuações diferentes do que já havia sido testado anteriormente, e realizando a rotulação de dados. Logo na figura 11 é apresentado o modelo de resolução desta pesquisa.

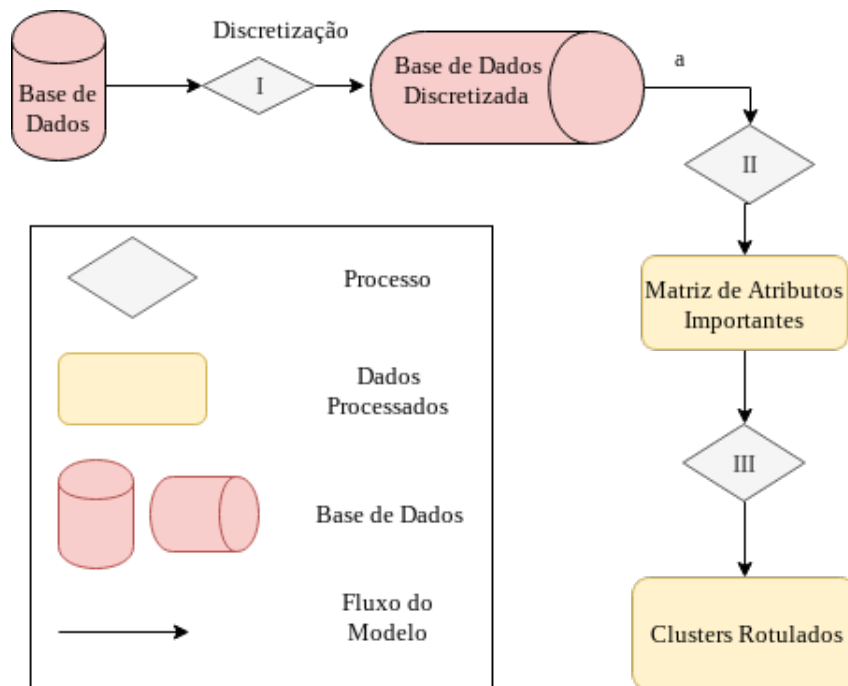


Figura 11 – Modelo de Resolução Proposto

A base de dados<sup>2</sup> do modelo (figura 11) conterá valores contínuos, contudo, conforme modelo, aplicará o método de discretização (I). Uma vez com a base discretizada ocorrerá a divisão em clusters, que nada mais é do que a separação da base em grupos já classificados mostrado na figura no fluxo *a*. É importante ressaltar que no fluxo *a* serve para mostrar como será a entrada no passo II, pois os clusters já são definidos na própria base.

No passo II será executado o algoritmo de aprendizagem supervisionada, já visto nas subseções 2.1.1.1, 2.1.1.2 e 2.1.1.3. Essa etapa utiliza a técnica de correlação de atributos, que neste modelo de resolução é considerado um processo de grande importância, e será visto na seção 3.3, mas já é possível através da figura 12 verificar seu funcionamento. Nesta figura ocorre um exemplo do funcionamento da técnica de correlação, onde uma base fictícia possuindo três atributos se correlacionam entre si. Como neste exemplo são três o número de atributos (atr1,atr2 e atr3), pode-se perceber que na figura 12 a quantidade de vezes que o algoritmo é executado é a mesma do número de atributos.

A saída do processo II gera uma matriz de atributos com seus respectivos valores, e através desses valores armazenados é escolhido um, ou mais, atributos de maior relevância. Para gerar essa matriz, conforme modelo figura 11, o algoritmo supervisionado será aplicado em uma quantidade igual ao número de atributos, ilustrado na figura 12 como exemplo.

No processo (III) acontecerá a escolha do atributo mais relevante (maior valor), e

<sup>2</sup> UCI - Machine Learning Repository. <http://archive.ics.uci.edu/ml/>

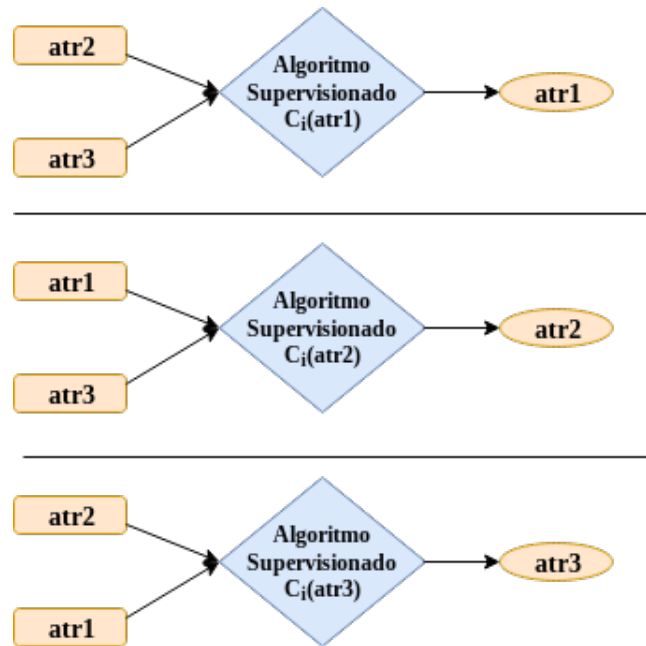


Figura 12 – Exemplo da técnica de correlação aplicada aos atributos atr1, atr2 e atr3.

podendo, em caso de mesmo valor, ter mais de um atributo relevante. Esta seleção será feita a partir da matriz (**Atributos Importantes**) criada pela implementação dos algoritmos supervisionados utilizando a técnica de correlação entre atributos seção (3.3), junto com o valor mais frequente desse(s) atributo(s). Após essa etapa é criado um conjunto de rotulos para cada clusters.

### 3.3 Técnica de Correlação entre Atributos através de Algoritmos Supervisionados

Essa técnica (LOPES; MACHADO; RABELO, 2016) utiliza por analogia a aprendizagem supervisionada, no qual os atributos de entrada são correlacionados com um atributo classe (atributo de saída). Conforme o número de atributos do cluster o atributo classe seria alterado seguindo uma sequência do primeiro ao último atributo desse cluster. Através desse processo cada atributo seria classe em relação aos outros atributos gerando um valor que seria armazenado em uma matriz.

De acordo com essa técnica os atributos de um cluster, descartando o atributo classe, seriam percorridos um por um, até o último, e a cada iteração de um atributo enquanto classe, seria armazenado o valor desse atributo em uma matriz chamada de atributos importantes. Essa matriz após montada mostraria os valores de cada atributo enquanto classe, e quanto maior o valor, mais relevante será este atributo em relação aos demais. Essa matriz, vista na figura ?? é definida em linhas, onde serão armazenadas os clusters, e compondo cada coluna os resultados dos atributos classe. Então caso uma base

possua 2 (dois) clusters e 3 (três) atributos, a matriz seria de ordem 2x3 (2 linhas e 3 colunas).

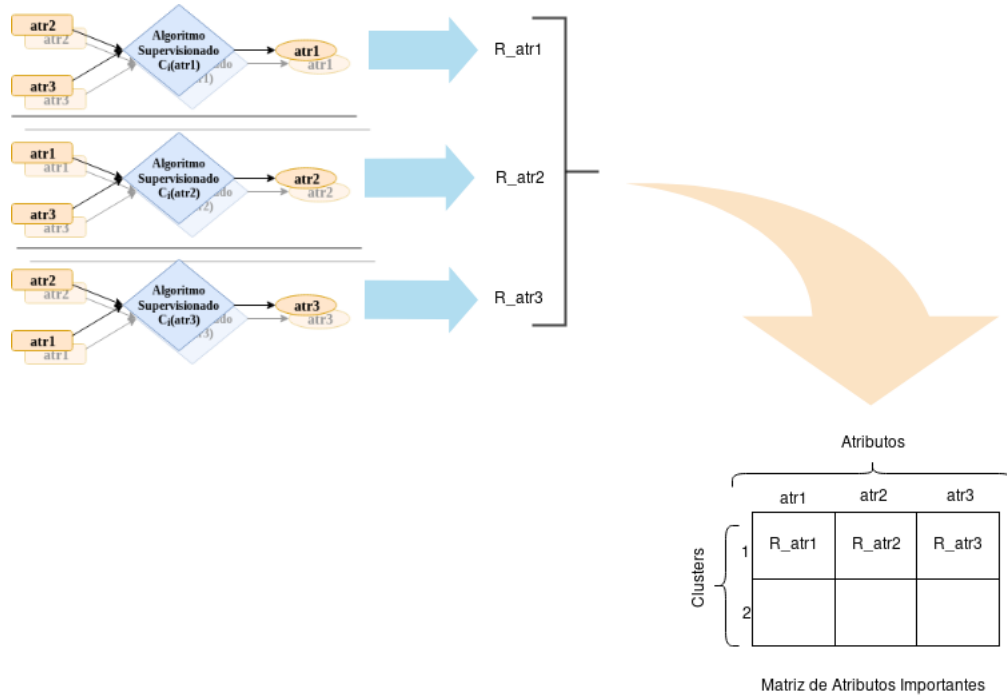


Figura 13 – Montagem da Matriz de Atributos Importantes

Tal técnica possui um grau de processamento diretamente proporcional a quantidade de características expressa na base de dados definido em  $R^m$  descrita na definição 3, onde  $R$  representa o conjunto de rótulos e  $m$  o número de atributos (dimensão do problema). Ela implica em utilizar todos os atributos, menos o definido como classe, para fazer uma correlação entre eles junto ao algoritmo.

Utilizando como exemplo uma base com os seguintes atributos: **atr1**, **atr2**, **atr3** e **classe**. Retirando o atributo classe, e atribuindo a cada iteração, um novo atributo classe, portanto, a base possui três atributos, então o algoritmo será aplicado três vezes, um para cada atributo gerando um valor que será armazenado em uma matriz. Em um primeiro processamento de três, o primeiro atributo **atr1** se torna classe e executado com os outros dois atributos, **atr2**, **atr3** de entrada com um algoritmo supervisionado, figura 14.



Figura 14 – Exemplo da técnica de correlação aplicada ao atributo,  $\text{atr1}$ , sendo classe

O resultado da correlação entre os atributos **atr2**, **atr3** em relação ao **atr1** (figura 14) é armazenado em uma matriz, denominada de **Atributos Importantes**, de acordo

com figura ???. Por conseguinte é realizado a aplicação do algoritmo com **atr2** sendo classe, e assim sucessivamente até o último atributo (**atr3**). Essa etapa só é finalizada quando todos os atributos tiverem a chance de ser classe, e armazenado seus valores em porcentagem na tabela, e quanto maior sua porcentagem, mais bem correlacionado é o atributo em relação aos demais.

### 3.4 Exemplo

Para melhor esclarecer as etapas do modelo de resolução exibido na figura 11, será utilizado a tabela 4 como exemplo no modelo proposto nesta pesquisa. Essa tabela é composta por cinquenta linhas e quatro atributos, sendo o último, um atributo classe representando o cluster. Logo na primeira coluna da tabela, possui o índice da linha da tabela identificando cada registro, e outros campos são atributos que definem características do registro identificado pelo índice da primeira coluna até a quinta coluna representando a classe de cada registro.

Tabela 4 – Base de Dados Modelo

n.	atr1	atr2	atr3	classe	n.	atr1	atr2	atr3	classe
1	2.08	92.11	22.07	2	26	1.42	53.51	19.64	3
2	1.26	85.03	20.45	1	27	1.12	62.71	19.07	1
3	2.00	108.36	22.68	2	28	2.09	60.58	20.20	1
4	1.74	43.78	18.72	3	29	1.95	69.23	19.68	1
5	1.82	100.20	23.09	2	30	1.03	47.81	19.47	3
6	1.43	77.59	21.80	1	31	1.75	90.92	21.39	2
7	1.53	44.01	20.98	3	32	1.72	42.35	22.89	3
8	1.14	107.77	18.99	2	33	1.47	101.77	19.20	2
9	1.97	98.00	22.32	2	34	1.53	41.16	22.67	3
10	1.50	39.67	21.78	3	35	1.44	93.61	21.03	2
11	1.74	55.86	20.31	3	36	1.51	98.65	19.24	2
12	1.80	65.72	19.62	1	37	1.06	68.82	21.68	1
13	1.33	82.01	19.82	1	38	1.48	80.40	21.43	1
14	1.66	103.93	21.10	2	39	1.14	61.59	19.90	1
15	1.42	66.14	21.61	1	40	1.08	91.93	20.81	2
16	1.87	88.36	22.45	2	41	1.62	79.21	18.43	1
17	1.11	107.82	19.32	2	42	1.68	80.87	18.42	1
18	2.08	67.66	20.74	1	43	1.81	98.24	22.13	2
19	1.85	82.65	20.35	1	44	1.30	69.27	18.83	1
20	1.04	102.62	19.46	2	45	1.80	101.21	21.61	2
21	1.97	100.37	21.94	2	46	1.79	72.02	22.02	1
22	1.95	45.70	22.10	3	47	1.56	81.71	22.10	1
23	1.77	50.04	20.16	3	48	1.98	77.16	21.71	1
24	1.97	81.57	19.83	1	49	1.86	89.12	22.84	2
25	1.52	93.13	20.61	2	50	1.55	76.01	19.74	1

Seguindo a definição 2 um elemento é expresso por um vetor de dimensão  $m$ , com tamanho igual ao número de atributos. Um exemplo do elemento 2 da tabela 4, pode ser representado por  $\vec{e}_2 = (1.26, 85.03, 20.45)$ .

### 3.4.1 Processo (I) - Discretização

Segundo (CATLETT, 1991; HWANG; LI, 2002) através de resultados experimentais, na conversão em atributos discretos ordenados de vários domínios constatou que a mudança de representação da informação na maioria das vezes pode aumentar a acurácia do sistema de aprendizado. Dessa maneira a etapa de discretização ganha um papel importante no modelo, e também no processo de Rotulação (III), pois é utilizada uma inferência na faixa discretizada para encontrar o intervalo na faixa.

Utilizando como exemplo a tabela 4 será utilizada a técnica de discretização por frequências iguais - EFD - e divisão de números de faixas igual a  $R=3$ . Na figura 15 poderá ser visualizado como é feita a discretização.

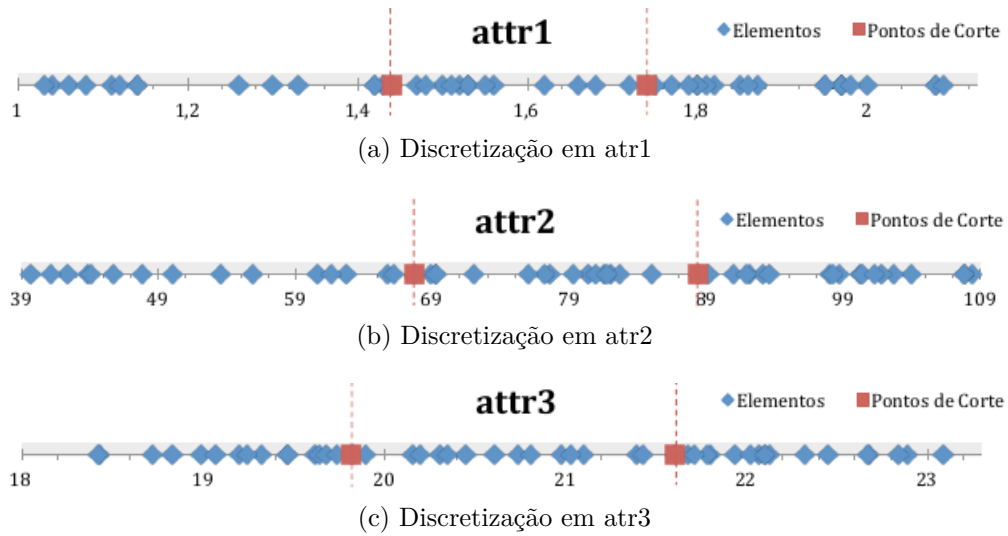


Figura 15 – Discretização de atributos utilizando EFD com  $R = 3$  (Figura adaptada de (LOPES; MACHADO; RABELO, 2016))

Através da figura 15 fica claro o conteúdo da faixa 1, contendo o valor inicial, 1(um), até o primeiro ponto de corte. Na faixa 2, o valor inicial é o primeiro número após o primeiro ponto de corte (término da faixa 1) até o segundo ponto de corte, incluindo o próprio ponto de corte. E na faixa 3 contém todos valores a partir do segundo ponto de corte.

A tabela 5 é o resultado após a discretização de todos os atributos. Para cada base de dados será definido o número de faixas de acordo com a configuração inicial antes da execução. Nessa configuração do sistema o número de faixas serve para toda a base de dados e não para cada atributo, então nesse exemplo o valor de  $R = 3$  conforme figura 15,

Tabela 5 – Base de Dados Modelo Discretizada

	atr1	atr2	atr3	classe		atr1	atr2	atr3	classe
1	3	3	3	2	26	1	1	1	3
2	1	2	2	1	27	1	1	1	1
3	3	3	3	2	28	3	1	2	1
4	2	1	1	3	29	3	2	1	1
5	3	3	3	2	30	1	1	1	3
6	1	2	3	1	31	3	3	2	2
7	2	1	2	3	32	2	1	3	3
8	1	3	1	2	33	2	3	1	2
9	3	3	3	2	34	2	1	3	3
10	2	1	3	3	35	1	3	2	2
11	2	1	2	3	36	2	3	1	2
12	3	1	1	1	37	1	2	3	1
13	1	2	1	1	38	2	2	2	1
14	2	3	2	2	39	1	1	2	1
15	1	1	2	1	40	1	3	2	2
16	3	2	3	2	41	2	2	1	1
17	1	3	1	2	42	2	2	1	1
18	3	1	2	1	43	3	3	3	2
19	3	2	2	1	44	1	2	1	1
20	1	3	1	2	45	3	3	2	2
21	3	3	3	2	46	3	2	3	1
22	3	1	3	3	47	2	2	3	1
23	3	1	2	3	48	3	2	3	1
24	3	2	2	1	49	3	3	3	2
25	2	3	2	2	50	2	2	1	1

Tabela 6 – Valores das faixas com R=3 da Base de Dados Modelo

	Faixa 1	Faixa 2	Faixa 3
<b>atr1</b>	[ 1.03 ~1.44 ]	] 1.44 ~1.74 ]	] 1.74 ~2.09 ]
<b>atr2</b>	[ 39.67 ~67.66 ]	] 67.66 ~88.36 ]	] 88.36 ~108.36 ]
<b>atr3</b>	[ 18.42 ~19.82 ]	] 19.82 ~21.61 ]	] 21.61 ~23.09 ]

onde  $R$  é o número de faixas a ser dividido tanto no **atr1** como também no **atr2** e **atr3** possuem os valores conforme tabela 6.

### 3.4.2 Processo (II) - Algoritmos Supervisionados

Ao chegar nessa etapa, Processo (II) da figura 11, já se tem uma base discretizada e clusters formados como visto na tabela 5. A partir desta etapa é feita a execução do algoritmo de aprendizado supervisionado obtendo como saída um valor, em porcentagem, informando o grau de correlacionamento entre os atributos compondo uma matriz, cuja função é armazenar o resultado da execução dos algoritmos utilizando a técnica de correlação de



atributos (3.3).

O algoritmo irá selecionar cluster por cluster, percorrendo todos os atributos destes clusters, onde a cada iteração um atributo será a classe da vez. Nesse exemplo, primeiramente o atributo **atr1** será classe e os demais irão participar como entrada junto ao algoritmo, verificando seu grau de importância entre eles. Depois o atributo **atr2** irá ser classe, e depois o **atr3**, fechando o ciclo de todos os atributos do cluster, visualizado na figura 12.

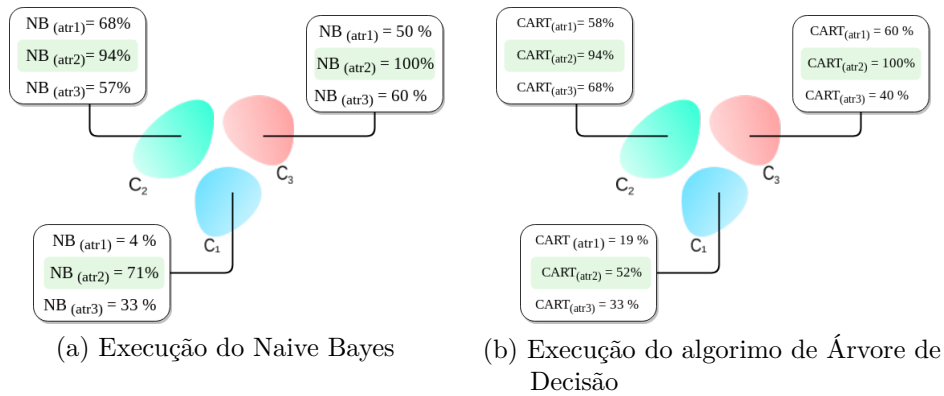


Figura 16 – Resultado dos Algoritmos

A figura 16a mostra o resultado da execução do Naive Bayes trabalhando com a base modelo (tabela 5) exibindo os resultados em porcentagem de acerto de cada atributo em relação aos demais. O mesmo acontece com a figura 16b onde é aplicado um algoritmo de Árvore de Decisão - CART - exibindo o resultado de todas as taxas de acerto, em porcentagem, dos atributos de seus respectivos clusters.

Uma forma de eliminar uma possível ambiguidade entre os clusters foi adicionar na implementação uma variável  $V$ . Essa variável é utilizada para seleção dos atributos rótulos de um clusters, caso aconteça dos rótulos se repetirem em clusters diferentes. Logo, todos os atributos que tiverem até uma diferença  $V$  em relação ao atributo de maior taxa de acerto, expresso em porcentagem, serão escolhidos como rótulo. Isto posto, se o atributo de maior taxa de acerto possuir 90%, e o  $V = 10\%$  então todos outros atributos que tiverem valores a partir de 80% são selecionados como rótulo do cluster.

O valor da variável  $V$  é subjetivo e irá ser arbitrado de acordo com os resultados em cada aplicação do algoritmo em um conjunto de dados. Nesse exemplo caso fosse utilizado a variância  $V = 12$  na matriz de atributos importantes representada pela figura 16a, teriam os atributos, por clusters,  $r_{c_1} : r_{c_1} = \{atr2\}$ ,  $r_{c_2} = \{atr2\}$ ,  $r_{c_3} = \{atr2\}$ .

### 3.4.3 Processo (III) - Rotulação

No processo de rotulação os rótulos de cada cluster ( $c_i$ ) serão compostos conforme o equação 3.1.

$$r_{ci} = \{(a_1, [p_1, q_1]), \dots, (a_{m(c_i)}, [p_{m(c_i)}, q_{m(c_i)}])\} \quad (3.1)$$

Cada rótulo é composto pela tupla: atributo de maior relevância (matriz de atributos importantes) e a faixa de valor desse atributo que mais se repete. Na figura 16 os rótulos em destaque são os que possuem maior valor, ademais, cada atributo que faz parte do rótulo possui um vetor de valores, de onde será escolhido a faixa de maior ocorrência. Uma vez calculado e definido a faixa, será determinado os limites inferiores ( $p_{m(c_i)}$ ) e superiores ( $q_{m(c_i)}$ ) de acordo com a tabela discretizada (exemplo 6).

Por exemplo, utilizando a Base Modelo, mais especificamente o cluster 1 ( $c_1$ ), cujo resultado é apresentado na figura 16a, o rótulo apresentado é o atributo **atr2** com a **faixa 2**, faixa esta encontrada após cálculo dos elementos de maior ocorrência, conforme descrito no parágrafo acima.

O rótulo apresentado ao final do processo terá a substituição do número da faixa pelos valores do intervalo conforme a tabela 6. Os rótulos dos clusters descrito neste exemplo - conforme figura 16a e figura 16b - aplicado na BD Modelo são:

- $r_{c_1} = (atr2, [67.66, 88.36]);$
- $r_{c_2} = (atr2, [88.36, 108.36])$  ;
- $r_{c_3} = (atr2, [39.67, 67.66]);$

A representação acima do rótulo informa que no rótulo do cluster 1 é uma tupla composta pelo atributo, **atr2**, e a faixa variando de valor maior que 67,66 até 88,36. No cluster 2 possui o rótulo composto também pelo atributo **atr2**, mas com faixa diferente, variando de um valor maior que 88,67 até 108,36. E por último o rótulo do cluster 3 com a faixa variando de 39,67 até 67,66. Isso significa que qualquer registro com valor no atr2 no intervalo  $]67.66, 88.36]$ , será agrupado no cluster 1, podendo ajudar um analista a interpretar ou tomar decisão em conformidade a esse rótulo. Então, caso esse atr2 fosse localização e esse intervalo determinasse uma região, isso poderia ser utilizado para tomada de decisão, dependendo do problema que queira resolver.

O algoritmo 1 exibe a rotina em forma de pseudocódigo para melhor entendimento, mas as variáveis  $V$ ,  $R$  e  $TipoDiscretização$ , não foram inicializadas por serem variáveis que dependem de testes para melhor otimização dos rótulos. No caso da variância  $V$  por padrão é inicializada com 0 (zero), porque ela é utilizada caso haja empate nos valores da matriz de importância na escolha dos rótulos, e caso assuma outro valor, ela dependerá dos valores da matriz. A variável  $R$  é utilizada para definir em quantas faixas será dividido os

valores do atributo na discretização, e por fim a variável *TipoDiscretização* que também dependerá do comportamento dos valores do atributo.

---

**Algorithm 1:** Rotina de Rotulação
 

---

```

1 Carrega_valores_auxiliares( $V, R, TipoDiscretização$ );
2 Carrega_BD;
3 Discretiza_BD;
4 Separa_em_clusters_de_acordo_com_classificação_BD;
5 while existir clusters do
6   while existir atributos do
7     atributo_classe=seleciona_nova_classe(atributos) ;
8     Aplica_algoritmo_supervisionado(atributo_classe, atributos_naoClasse);
9   Calcula_matriz_de_porcentagem_de_acertos;
10  if  $V \neq 0$  then
11    Carrega_atributos_importantes_considerando_V;
12  Associa_valores_aos_intervalos;
13 Exibe_rótulos_todos_clusters;

```

---

## 4 Resultados

Foram realizados testes com algumas bases de dados da UCI Machine Learning<sup>1</sup>, um repositório de dados a serviço da comunidade de aprendizado de máquina. Criado por estudantes de pós-graduação na UC Irvine em 1987, essas bases são utilizadas por estudantes mas e por educadores e pesquisadores como fonte primária de aplicações de aprendizado de máquina.

\* lembrar que as bases são essas (iris, ...) por serem já estudadas, e de prévio conhecimento, onde a partir de estudos em bases assim pode-se ter resultados conclusivos e expandir para outras bases. É como se testasse com base conhecida para depois pegar uma nova.

As bases de dados foram escolhidas não só por critério comparativo de outros trabalhos que também já as utilizaram servindo de referência para os resultados, como também, um cuidado de só escolher bases que estão classificadas, uma vez que esta pesquisa trabalhará com os clusters já formados e não na criação de grupos.

A divisão deste capítulo iniciará por uma explanação da implementação do trabalho explicando as ferramentas utilizadas no desenvolvimento, e quais configurações de algumas variáveis. Cada seção refere-se a uma base de dados utilizada, sendo esta, dividida em subseções para os algoritmos: Naive Bayes, CART e KNN.

### 4.1 Implementação

Para gerar os resultados aqui escritos foram realizadas implementações utilizando a ferramenta MATLAB<sup>2</sup>, sendo possível utilizar suas funções de aprendizado de máquina já implementadas na *Statistics and Machine Learning Toolbox*. Por apresentar linguagem técnica e funções já prontas direcionada para aprendizado de máquina essa ferramenta foi escolhida para colocar em prática essa pesquisa.

Ao longo da pesquisa foram realizados vários testes, porém, houveram alterações de algumas variáveis e métodos de discretização, sempre com o objetivo de obter os melhores resultados. As variáveis alteradas: variância “V”, número de faixas “R”, e métodos de discretização “R” (EWD,EFD).

Como dito na subseção 3.4.2 a variação  $V$  existe para evitar a ambiguidade dos rótulos, ou seja, quando rótulos apresentarem os mesmos resultados: atributo e faixa de

---

<sup>1</sup> <http://archive.ics.uci.edu/ml/>

<sup>2</sup> <http://www.mathworks.com/products/matlab/> ; versão: R2016a(9.0.0.341360); 64-bit (glnxa64)

valor. Além de evitar a ambiguidade dos rótulos a variável  $V$  pode ser utilizada também para selecionar mais de um atributo para ser o rótulo do cluster.

É importante ressaltar a criação da tabela de correlação de atributos (Matriz de Atributos Importantes). Essa tabela é implementada conforme a técnica de correlação entre atributos com algoritmos supervisionados, seção 3.4.2, onde cada célula da tabela é preenchida através da execução de um algoritmo supervisionado. Estas execuções são realizadas em todos os atributos da cada cluster existente na base de dados.

Após a tabela preenchida, o atributo rótulo será selecionado a partir do maior valor em relação aos outros atributos do grupo, que é representado pela linha da tabela (matriz de atributos). Também poderá ser selecionado como rótulo os atributos que possuam o valor entre a diferença de  $V$  com o atributo de maior valor (mais relevante). Por exemplo, se o valor de  $V = 5\%$  e o atributo de maior valor é **95%**, então todos os atributos que possuírem o valor a partir de **90%** serão considerados rótulos também.

A cada base de dados descritas nas seções seguintes, são configuradas algumas variáveis, método de discretização e implementado dois algoritmos de aprendizado supervisionado com paradigmas diferentes para fazer rotulação. Cada algoritmo terá como resultado o rótulo por cluster de dados.

## 4.2 Seeds - Identificação de Tipos de Semente

Essa base pertence a UCI Machine Learning, e composta por sete atributos definindo suas características e mais um atributo classe responsável por identificar os tipos de sementes (CHARYTANOWICZ et al., 2010). Em seus atributos seus valores são todos contínuos e não existem valores em branco, possuindo um total de 210 registros classificados em três categorias:

- 70 elementos do tipo Kama;
- 70 elementos do tipo Rosa;
- 70 elementos do tipo Canadian.

Para classificar as sementes, como Kama, Rosa e Canadian foi utilizada uma técnica de raio X, que é relativamente mais barata que outras técnicas de imagem, como microscopia ou tecnologia a laser. O material foi colhido de campos experimentais, explorados no Instituto de Agrofísica da Academia Polonês de Ciências em Lublin.

Como já mencionado neste capítulo, na seção 4.1, antes de executar o algoritmo algumas configurações são necessárias. A primeira, é a configuração do método de discretização para o tipo EFD, e a segunda é a divisão dos valores dos atributos em faixas, com

$R = 3$  para todos os atributos. Tanto a discretização como o valor do número de faixas foram testados e escolhidos para alcançar melhores resultados.

### 4.2.1 Naive Bayes

Tabela 7 – Resultado da rotulação com o algoritmo Naive Bayes

Cluster	Rótulos		Relevância(%)	Fora da Faixa	Acurácia Cluster(%)
	Atributos	Faixa			
1	area	] 12.78 ~ 16.14 ]	92%	14	80%
2	area	] 16.14 ~ 21.18 ]	95%	6	91,4%
3	perimetro	[ 12.41 ~ 13.73 ]	95%	5	92,8%

Na tabela 7 é apresentado os resultados de rotulação do algoritmo Naive Bayes. Essa tabela é composta por colunas que informam os **Clusters**, **Rótulos** que integram **Atributo** e sua **Faixa** de valor, além da coluna **Relevância** exibida em porcentagem, bem como as colunas **Fora da Faixa** e **Acurácia Cluster** que exibem a quantidade de elementos que não estão dentro da faixa designada pelo do rótulo encontrado, e a acurácia do cada cluster respectivamente.

A coluna **Relevância** demonstra o maior valor entre os atributos de cada cluster, e caso esses valores sejam ambíguos, serão exibidos na coluna todos estes atributos. Para ter maior clareza na escolha desses atributos foi inserida a tabela 8 que exibem os valores de correlação entre eles.

Já na coluna, **Fora da Faixa**, tem a função de exibir, em números, a quantidade de valores que não estão participando da faixa definida pelo rótulo. Através de experimentos percebeu-se o mérito de apresentar em números a quantidade de elementos que não estão sendo representados pelo rótulo gerando mais realidade as informações, ao invés de exibir em porcentagem.

Na última coluna, **Acurácia Cluster**, apresenta em porcentagem o grau de acerto, por cluster, dos registros que são representados pelo rótulo. Foi possível expor estas informações visto que cada cluster já apresenta a quantidade e quais registros fazem parte de cada cluster.

Analisando a coluna Rótulos da tabela 7, nota-se que o atributo **area** aparece tanto no cluster 1 como também no cluster 2. A rotulação de dados envolve não só o atributo mais relevante, como também, a faixa de valores que mais se repete dentro do atributo. Nesse caso pode-se observar que na coluna **Atributo** o atributo **area** se repete entre os cluster 1 e 2, mas no cluster 1 a faixa de valores difere do cluster 2, sendo considerados rótulos distintos.

A tabela 8 é um exemplo da matriz de atributos importantes gerada pela técnica de correlação entre atributos. É formada por clusters representado pelas linhas, e atributos

Tabela 8 – Resultado da Correlação dos atributos pelo Naive Bayes; Legenda dos Atributos: (A)area, (B)perimetro, (C)compactness, (D)Lkernel, (E)Wkernel, (F)asymetry, (G)lkgroove

		Atributos						
		A	B	C	D	E	F	G
Clusters	1	92.8	87.1	50.0	75.7	85.7	60.0	65.7
	2	95.7	91.4	47.1	92.8	90.0	28.5	85.7
	3	91.4	95.7	71.4	85.7	91.4	64.2	58.5

representado por colunas, onde esses valores são representados em porcentagem (%).

Essa tabela é fruto da aplicação do Naive Bayes na base de dados **Seeds**, e a partir dela é retirado o(s) atributo(s) rótulo(s). Uma análise pode ser feita através desses dados e ajudar a definir um valor para a variável  $V$  caso necessário. Percebe-se que algumas características são mais bem correlacionadas que outras, através dos valores mais altos indicando o grau de relacionamento entre os atributos após a aplicação do algoritmo.

Para provar empiricamente os resultados, na tabela 9 é exposto 4 (quatro) execuções do Algoritmo Naive Bayes. Pode-se constatar que mesmo havendo algumas alterações nos valores dos atributos em cada execução, a correlação entre os atributos não oferece muita alteração. Como exemplo, o atributo **area** nos clusters 1 e 2, possuem o melhor grau de correlacionamento em seus grupos, mesmo nas quatro execuções, como mostrado na tabela 9.

A informação passada pela tabela 9 tem a intensão de mostrar para o analista que estiver aplicando a rotulação de dados, como seus atributos se comportam em cada cluster, ao utilizar o método de rotulação de dados.

Segue abaixo o resultado do algoritmo Naive Bayes na base de dados **Seeds** com seus rótulos:

- $r_{c_1} = \{(area, ]12.78 \sim 16.14])\}$
- $r_{c_2} = \{(area, ]16.14 \sim 21.18])\}$
- $r_{c_3} = \{(perimetro, [12.41 \sim 13.73])\}$

#### 4.2.2 CART

Na tabela 10, que segue o mesmo modelo da tabela 7, tem-se o resultado da aplicação do algoritmo supervisionado na base Seeds. O CART é utilizado pela toolbox do MATLAB como algoritmo de classificação de árvore de decisão. O que se pretende fazer é seguir a pesquisa e testar a base de dados com um paradigma diferente para fazer rotulação nos clusters.

Já na tabela 11 são exibidas algumas execuções do algoritmo CART na base

Tabela 9 – Resultado de 4 (quatro) execuções do algoritmo Naive Bayes; Legenda dos Atributos: (A)area, (B)perimetro, (C)compactness, (D)Lkernel, (E)Wkernel, (F)asymetry, (G)lkgroove

(a) 1a. Execução

1a. Execução		Atributos						
		A	B	C	D	E	F	G
Clusters	1	92.8	87.1	48.5	77.1	82.8	57.1	65.7
	2	94.2	90.0	45.7	92.8	90.0	38.5	87.1
	3	91.4	95.7	72.8	85.7	91.4	64.2	60.0

(b) 2a. Execução

2a. Execução		Atributos						
		A	B	C	D	E	F	G
Clusters	1	92.8	87.1	47.1	77.1	87.1	60.0	65.7
	2	94.2	90.0	47.1	92.8	91.4	32.8	87.1
	3	91.4	95.7	72.8	85.7	92.8	64.2	60.0

(c) 3a. Execução

3a. Execução		Atributos						
		A	B	C	D	E	F	G
Clusters	1	94.2	85.7	48.5	77.1	82.8	61.4	65.7
	2	92.8	90.0	50.0	92.8	90.0	32.8	87.1
	3	91.4	95.7	72.8	85.7	92.8	64.2	60.0

(d) 4a. Execução

4a. Execução		Atributos						
		A	B	C	D	E	F	G
Clusters	1	91.4	88.5	54.2	75.7	85.7	62.8	61.4
	2	95.7	90.0	50.0	92.8	90.0	38.5	85.7
	3	91.4	95.7	72.8	85.7	94.2	64.2	57.1

Tabela 10 – Resultado da aplicação do algoritmo CART

Cluster	Rótulos		Relevância(%)	Fora da Faixa	Acurácia Cluster(%)
	Atributos	Faixa			
1	perimetro	[ 13.73 ~ 15.18 ]	94%	14	80%
2	area	] 16.14 ~ 21.18 ]	98%	6	90%
	perimetro	] 15.18 ~ 17.25 ]	98%	7	
3	wkernel	[ 2.63 ~ 3.049 ]	97%	9	87,1%

de dados. O mesmo comportamento entre execuções pode ser visto no algoritmo de paradigma estatístico, subseção 4.2.1, realizado nessa pesquisa. O comportamento de ambos os algoritmos foram bem semelhantes, como também, seus valores nas execuções que não se alteraram muito a cada iteração.

O resultado da rotulação utilizando o algoritmo CART na base de dados **Seeds** tem como rótulos:



Tabela 11 – Resultado de 4 (quatro) execuções do algoritmo Naive Bayes; Legenda dos Atributos: (A)area, (B)perimetro, (C)compactness, (D)Lkernel, (E)Wkernel, (F)asymetry, (G)lkgroove

(a) 1a. Execução

1a. Execução		Atributos						
		A	B	C	D	E	F	G
Clusters	1	91.4	94.2	58.5	80.0	74.2	55.7	60.0
	2	98.5	98.5	50.0	90.0	88.5	41.4	90.0
	3	92.8	95.7	80.0	88.5	97.1	55.7	77.1

(b) 2a. Execução

2a. Execução		Atributos						
		A	B	C	D	E	F	G
Clusters	1	91.4	94.2	62.8	78.5	81.4	61.4	57.1
	2	98.5	98.5	54.2	90.0	88.5	40.0	90.0
	3	92.8	95.7	80.0	88.5	97.1	60.0	77.1

(c) 3a. Execução

3a. Execução		Atributos						
		A	B	C	D	E	F	G
Clusters	1	93.8	93.6	61.8	83.2	89.2	53.2	71.0
	2	98.2	98.3	61.9	93.0	90.5	25.2	90.1
	3	95.5	96.3	82.4	90.9	97.7	59.3	77.0

(d) 4a. Execução

4a. Execução		Atributos						
		A	B	C	D	E	F	G
Clusters	1	92.8	94.2	60.0	80.0	84.2	64.2	60.0
	2	98.5	98.5	47.1	91.4	90.0	42.8	88.5
	3	91.4	95.7	80.0	88.5	97.1	55.7	77.1

- $r_{c_1} = \{(perimetro, ]13.73 \sim 15.18])\}$
- $r_{c_2} = \{(area, ]16.14 \sim 21.18]), (perimetro, ]15.18 \sim 17.25])\}$
- $r_{c_3} = \{(wkernel, [2.63 \sim 3.049])\}$

## 4.3 Iris - Identificação de Tipos de Plantas

A base de dados **Iris**, também pertencente a UCI Machine Learning, é muito conhecida em outras pesquisas ??Filho, Machado e Lira (2015), como também na literatura em reconhecimentos de padrões, por utilizar classes de plantas bem definidas. Contêm 3 classes de 50 instâncias cada, totalizando 150 registros de amostra de plantas. O atributo classe classifica o tipo de planta em 3 tipos (FISHER, 1936):

- 50 elementos da classe Iris-setosa ;

- 50 elementos da classe Iris-versicolour;
- 50 elementos da classe Iris-virginica.

Os atributos correspondentes são comprimento da sepala - SL, largura da sepala - SW, comprimento da pétala - PL e largura da pétala - PW. Através dessas características há uma classificação para dizer qual tipo de planta.

Para alcançar os resultados do algoritmo na base de dados, foram aplicadas algumas configurações. Estas configurações foram o método de discretização, tipo EFD, seção 2.2.2, e a divisão em três faixas de valores  $R = 3$  para todos os atributos, e inserido o valor de variação  $V = 0\%$ , tabela 13.

Seguindo a análise, semelhante da base de dados anterior, serão realizados testes utilizando os algoritmos, Naive Bayes e CART. Seus resultados serão exibidos em tabelas. Também foi posto nas tabelas 13 e 15 os resultados da técnica de correlações entre os atributos de cada grupo, servindo de informação para decisão do valor de  $V$ , caso fosse necessário. E também apresentado os resultados de outras iterações de cada algoritmo, para mostrar o comportamento dos atributos entre eles no grupo.

### 4.3.1 Naive Bayes

Através da tabela 12 os resultados da rotulação são exibidos após a aplicação do algoritmo. Com essa base de dados nota-se que no cluster 1 houve um acerto de 100% da rotulação. O cluster 2 e cluster 3 obtiveram rótulos distintos, cada um com grau de relevância acima de 80% em relação aos outros atributos de cada grupo.

Tabela 12 – Resultado da aplicação do algoritmo Naive Bayes

Cluster	Rótulos		Relevância(%)	Fora da Faixa	Acurácia Cluster(%)
	Atributos	Faixa			
1	petallength	[ 1.0 ~ 3.7 ]	100%	0	100%
	petalwidth	[ 0.1 ~ 1.0 ]	100%	0	
2	petallength	] 3.7 ~ 5.1 ]	84%	7	86%
3	petalwidth	] 1.7 ~ 2.5 ]	90%	5	90%

A porcentagem representada na coluna de relevância não pode ser analisada isoladamente. Para isso a tabela 13 possui os valores de correlação de todos os atributos. Todos os números estão representados em porcentagem para melhor análise do grau de relacionamento entre os outros atributos.

Na tabela 13 foram inseridas quatro resultados de execuções do algoritmo. Foi escolhida na tabela 13a a 1a. execução para montar a tabela de rótulos, tabela 12. A partir dessas execuções o pesquisador poderá arbitrar sobre o valor de  $V$  para melhor adaptá-lo a base. Das várias execuções expostas na tabela 13, percebe-se que não há muita diferença entre os valores de cada execução. Isso mostra um padrão de valores de acordo com a base.

Tabela 13 – Resultado (em %) de 4 (quatro) execuções do algoritmo Naive Bayes; Legenda dos Atributos: (SL)sepalength, (SW)sepalwidth, (PL)petallength, (PW)petalwidth

(a) 1a. Execução					
1a. Execução		Atributos			
		SL	SW	PL	PW
Clusters	1	80	68	<b>100</b>	<b>100</b>
	2	72	76	<b>84</b>	82
	3	76	74	68	<b>90</b>

(b) 2a. Execução					
2a. Execução		Atributos			
		SL	SW	PL	PW
Clusters	1	80	68	100	100
	2	72	76	88	84
	3	70	74	70	90

(c) 3a. Execução					
3a. Execução		Atributos			
		SL	SW	PL	PW
Clusters	1	80	68	100	100
	2	72	74	84	84
	3	74	74	68	90

(d) 4a. Execução					
4a. Execução		Atributos			
		SL	SW	PL	PW
Clusters	1	80	68	100	100
	2	72	74	86	82
	3	70	74	70	92

No caso da 1a. execução (tabela 13a) os valores escolhidos como rótulo estão destacados em cada cluster.

Os rótulos com o algoritmo Naive Bayes na base de dados **Iris** são dados abaixo:

- $r_{c_1} = \{(petallength, [1.0 \sim 3.7]), (petalwidth, [0.1 \sim 1.0])\}$
- $r_{c_2} = \{(petallength, ]3.7 \sim 5.1])\}$
- $r_{c_3} = \{(petalwidth, ]1.7 \sim 2.5])\}$

### 4.3.2 CART

A aplicação do algoritmo CART na base de dados **Iris** gerou a tabela 14 como resultado, e ao examinar pode-se observar uma semelhança com a subseção anterior onde foi aplicado o Naive Bayes.

Tabela 14 – Resultado da aplicação do algoritmo CART

Cluster	Rótulos		Relevância(%)	Fora da Faixa	Acurácia Cluster(%)
	Atributos	Faixa			
1	petallength	[ 1.0 ~ 3.7 ]	100%	0	100%
	petalwidth	[ 0.1 ~ 1.0 ]	100%	0	
2	petalwidth	] 1.0 ~ 1.7 ]	90%	8	84%
3	petalwidth	] 1.7 ~ 2.5 ]	90%	5	90%

Ao observar a tabela 14 percebe-se que o resultado de rotulação no cluster 1 e 3 são idênticos ao do algoritmo apresentado anteriormente, mas no cluster 2 o rótulo é diferenciado pelo atributo petalwidth que atinge valores mais altos em todas as execuções, como mostra a tabela 15.

Segue abaixo os rótulos na base de dados **Iris** aplicado pelo algoritmo CART:

Tabela 15 – Resultado de 4 (quatro) iterações do algoritmo CART; Legenda dos Atributos: (SL)sepalength,(SW)sepalwidth,(PL)petallength,(PW)petalwidth

(a) 1a. Execução						(b) 2a. Execução					
1a. Execução		Atributos				2a. Execução		Atributos			
		SL	SW	PL	PW			SL	SW	PL	PW
Clusters	1	80	68	<b>100</b>	<b>100</b>	Clusters	1	80	68	100	100
	2	74	76	88	<b>90</b>		2	74	76	88	90
	3	68	68	74	<b>90</b>		3	70	70	74	90

(c) 3a. Execução						(d) 4a. Execução					
3a. Execução		Atributos				4a. Execução		Atributos			
		SL	SW	PL	PW			SL	SW	PL	PW
Clusters	1	80	68	100	100	Clusters	1	80	68	100	100
	2	72	74	84	84		2	72	74	86	90
	3	74	74	68	90		3	68	66	78	90

- $r_{c_1} = \{(petallength, [1.0 \sim 3.7]), (petalwidth, [0.1 \sim 1.0])\}$
- $r_{c_2} = \{(petalwidth, ]1.0 \sim 1.7])\}$
- $r_{c_3} = \{(petalwidth, ]1.7 \sim 2.5])\}$

## 4.4 Glass - Identificação de Tipos de Vidros

Essa base ficou conhecida por Vina Spiehler, Ph.D. da DABFT Diagnostic Products Corporation, onde conduziu pesquisas e testes de comparação em seu sistema baseado em regras determinando, se o tipo de vidro era temperado ou não. Institutos de investigação criminológica motivaram os estudos de classificação de tipos de vidros, porque em uma cena de crime, uma classificação de tipos de vidro corretamente identificada pode ser utilizada como prova, ajudando diretamente na investigação (EVETT; SPIEHLER, 1988).

Possui um total de 214 instancias, caracterizados por 9 atributos (RI, Na, Mg, Al, Si, K, Ca, Ba e Fe), sendo que o atributo **RI** indica o índice de refração, e quanto aos demais atributos são valores correspondentes a porcentagem do óxido.

Os tipos de vidro (atributo classe) foram divididos em 7 grupos distintos:

- 1 janelas de construção - vidro temperado: 70 registros
- 2 janelas de construção - vidro não-temperado: 76 registros
- 3 janelas de veículos - vidro temperado: 17 registros
- 4 janelas de veículos - vidro não-temperado: 0 registro
- 5 recipientes: 13 registros

- 6 louças de mesa: 9 registros
- 7 lâmpadas: 29 registros

Para execução dos algoritmos foram definidos a quantidade de faixas ( $R$ ) que serão divididos os valores dos atributos, qual o método de discretização e o valor de variação  $V$  caso haja ambiguidade. Nos teste desenvolvidos nesta pesquisa os valores de referência foram,  $R = 3$  para o número de faixas, o método de discretização EWD e o valor  $V = 0\%$ .

#### 4.4.1 Naive Bayes

Ao observar a tabela 16 percebe-se que a coluna **Relevância** obteve porcentagens altas, ressaltando nos rótulos de cada grupo os atributos que mais bem se relacionaram. E em específico no **cluster 5** atributo **Na**, o valor da coluna de **Relevância = 100%**, mas na coluna, **Fora da Faixa**, apresentam 2(dois) elementos que não estão sendo representados pelo rótulo.

Essa situação dita no parágrafo acima segue a Definição 3, mas é um exemplo prático que não aconteceu em outros testes das outras bases de dados, e por isso segue um esclarecimento. A definição é que cada rótulo específico é dado por um conjunto de pares de valores, tendo como saída um vetor com atributo e seu respectivo intervalo,  $r_{ci} = \{(a_1, [p_1, q_1]), \dots, (a_{m(c_i)}, [p_{m(c_i)}, q_{m(c_i)}])\}$  capaz de melhor expressar o cluster  $c_i$ . Então caso a coluna **Relevância** seja igual a 100%, isso não implica que todos os elementos tenham que estar dentro da faixa  $p_{m(c_i)}$  (limite inferior) e  $q_{m(c_i)}$  (limite superior), e sim, a maioria dos elementos, mostrando que o rótulo é capaz de melhor representar o cluster.

Além de apresentar dados desbalanceados o **Cluster 5** apresentado na tabela 16 conta com o total de nove elementos, e entre estes, nenhum participa da 1a. faixa, dois estão na 2a. faixa e os restantes (sete) estão na 3a. faixa. Dessa maneira justifica-se o porquê dos dois elementos estarem de fora do rótulo, pois a faixa rótulo escolhida é a 3a. faixa, onde contém a maioria dos elementos, por conseguinte, escolhida para representar o rótulo.

Os resultados da tabela 17, assim como nos resultados de bases anteriores, indicam uma sequência de execuções onde é possível observar o comportamento das variáveis que são escolhidas como rótulo. Nestes exemplos fica claro que não foi necessária a utilização de uma variação  $V$  para a escolha dos rótulos, logo porque não houve ambiguidade entre eles. Por outro lado, quando testes utilizaram o outro método de discretização, EFD, retornaram rótulos ambíguos obrigando o uso da variação  $V$ . Em consequência disto foi definindo o método de discretização EWD como padrão para a rotulação de dados.

De acordo com a aplicação do Naive Bayes na base de dados **Glass** os rótulos são os seguintes:

Tabela 16 – Resultado da aplicação do algoritmo Naive Bayes

Cluster	Rótulos		Relevância(%)	Fora da Faixa	Acurácia Cluster(%)
	Atributos	Faixa			
1	Mg	[ 2.245 ~ 4.490 ]	100%	0	100%
	K	[ 0.0 ~ 1.5525 ]	100%	0	
	Ba	[ 0.0 ~ 0.7875 ]	100%	0	
2	K	] 0.0 ~ 1.5525 ]	100%	0	100%
3	Mg	] 2.245 ~ 4.490 ]	100%	0	100%
	K	] 0.0 ~ 1.5525 ]	100%	0	
	Ca	] 8.12 ~ 10.81 ]	100%	0	
	Ba	[ 0.0 ~ 0.7875 ]	100%	0	
4	Al	[ 1.0925 ~ 1.895 ]	92%	4	69,2%
	K	[ 0.0 ~ 1.5525 ]	92%	3	
	Ba	[ 0.0 ~ 0.7875 ]	92%	1	
5	Na	[14.055 ~ 17.38 ]	100%	2	77,7%
	K	[ 0.0 ~ 1.5525 ]	100%	0	
	Ba	[ 0.0 ~ 0.7875 ]	100%	0	
	Fe	[ 0.0 ~ 0.1275 ]	100%	0	
6	Fe	[ 0.0 ~ 0.1275 ]	100%	0	100%

- $r_{c_1} = \{(Mg, [2.245 \sim 4.490]), (K, [0.0 \sim 1.5525]), (Ba, [0.0 \sim 0.7875])\}$
- $r_{c_2} = \{(K, [0.0 \sim 1.5525])\}$
- $r_{c_3} = \{(Mg, [2.245 \sim 4.490]), (K, [0.0 \sim 1.5525]), (Ca, [8.12 \sim 10.81]), (Ba, [0.0 \sim 0.7875])\}$
- $r_{c_4} = \{(Al, [1.0925 \sim 1.895]), (K, [0.0 \sim 1.5525]), (Ba, [0.0 \sim 0.7875])\}$
- $r_{c_5} = \{(Na, [14.055 \sim 17.380]), (K, [0.0 \sim 1.5525]), (Ba, [0.0 \sim 0.7875]), (Fe, [0.0 \sim 0.1275])\}$
- $r_{c_6} = \{(Fe, [0.0 \sim 0.1275])\}$

#### 4.4.2 CART

Ao utilizar o algoritmo CART logo percebe-se a semelhança com os resultados apresentados na subseção 4.4.1. Apesar dessa semelhança os **Clusters 4 e 5** tiveram diferenças nos resultados em comparação ao algoritmo Naive Bayes.

Ao verificar a **linha 4** da tabela 17 do Naive Bayes, correspondente ao **Cluster 4**, os atributos **Al**, **K**, **Ba** apresentaram sempre os mesmos valores, mas já na tabela 19, também na **linha 4** de cada execução, só o valor de **Ba** coincide já os outros atributos tiveram valores mais baixos, fazendo com que eles não participassem da composição do rótulo.

No **Cluster 5** o atributo **Na** não faz parte do rótulo, e diferente do Naive Bayes na tabela 17, verifica-se que os valores de **Na** são sempre 100% de correlação entre os outros atributos. No CART os valores apresentados de **Na** nas execuções da tabela 19, **linha 5**, são abaixo dos 78%. Na **1a. Execução** da tabela 19a os atributos que compõem o rótulo do **Cluster 5** apresentam também 100%, portanto qualquer atributo com valor abaixo de 100% não será escolhido para compor o rótulo.

Tabela 17 – Resultado de 4 (quatro) execuções do algoritmo Naive Bayes.

(a) 1a. Execução

1a. Exec		Atributos								
		RI	Na	Mg	Al	Si	K	Ca	Ba	Fe
Clusters	1	87.1	92.8	100	81.4	82.8	100	90.0	100	78.5
	2	65.7	86.8	85.5	82.8	56.5	100	73.6	98.6	61.8
	3	82.3	82.3	100	76.4	58.8	100	100	100	82.3
	4	84.6	69.2	30.76	92.3	76.9	92.3	76.9	92.3	69.2
	5	77.7	100	33.3	66.6	44.4	100	55.5	100	100
	6	58.6	79.3	79.3	72.4	79.3	93.1	93.1	13.7	100

(b) 2a. Execução

2a. Exec		Atributos								
		RI	Na	Mg	Al	Si	K	Ca	Ba	Fe
Clusters	1	87.1	92.8	100	81.4	81.4	100	90.0	100	78.5
	2	65.7	92.1	88.1	82.8	63.1	100	72.3	97.3	61.8
	3	72.4	82.3	100	76.4	47	100	100	100	82.3
	4	84.6	69.2	23	92.3	76.9	92.3	76.9	92.3	61.5
	5	77.7	100	33.3	66.6	44.4	100	55.5	100	100
	6	58.6	79.3	79.3	68.9	79.3	93.1	93.1	17.2	100

(c) 3a. Execução

3a. Exec		Atributos								
		RI	Na	Mg	Al	Si	K	Ca	Ba	Fe
Clusters	1	87.1	92.8	100	81.4	84.2	100	90.0	100	78.5
	2	68.4	89.4	86.8	84.2	60.5	100	72.3	98.6	64.4
	3	76.4	82.3	100	76.4	52.9	100	100	100	82.3
	4	84.6	69.2	23	92.3	76.9	92.3	76.9	92.3	76.9
	5	77.7	100	33.3	66.6	44.4	100	55.5	100	100
	6	58.6	79.3	79.3	68.9	79.3	93.1	89.6	13.7	100

(d) 4a. Execução

4a. Exec		Atributos								
		RI	Na	Mg	Al	Si	K	Ca	Ba	Fe
Clusters	1	87.1	92.8	100	81.4	84.2	100	90.0	100	78.5
	2	65.7	90.7	86.8	82.8	59.2	100	76.3	98.6	63.1
	3	76.4	82.3	100	76.4	52.4	100	100	100	82.3
	4	84.6	53.8	23	92.3	76.9	92.3	76.9	92.3	69.2
	5	77.7	100	33.3	66.6	44.4	100	55.5	100	100
	6	58.6	82.7	79.3	72.4	79.3	93.1	82.7	6.8	100

De acordo com a aplicação do CART na base de dados **Glass** os rótulos são os seguintes:

- $r_{c_1} = \{(Mg, [2.245 \sim 4.490]), (K, [0.0 \sim 1.5525]), (Ba, [0.0 \sim 0.7875])\}$
- $r_{c_2} = \{(K, [0.0 \sim 1.5525])\}$
- $r_{c_3} = \{(Mg, [2.245 \sim 4.490]), (K, [0.0 \sim 1.5525]), (Ca, [8.12 \sim 10.81]), (Ba, [0.0 \sim 0.7875])\}$

Tabela 18 – Resultado da aplicação do algoritmo CART

Cluster	Rótulos		Relevância(%)	Fora da Faixa	Acurácia Cluster(%)
	Atributos	Faixa			
1	Mg	[ 2.245 ~ 4.490 ]	100%	0	100%
	K	[ 0.0 ~ 1.5525 ]	100%	0	
	Ba	[ 0.0 ~ 0.7875 ]	100%	0	
2	K	] 0.0 ~ 1.5525 ]	100%	0	100%
3	Mg	] 2.245 ~ 4.490 ]	100%	0	100%
	K	] 0.0 ~ 1.5525 ]	100%	0	
	Ca	] 8.12 ~ 10.81 ]	100%	0	
	Ba	[ 0.0 ~ 0.7875 ]	100%	0	
4	Ba	[ 0.0 ~ 0.7875 ]	92%	1	92,3%
5	K	[ 0.0 ~ 1.5525 ]	100%	0	100%
	Ba	[ 0.0 ~ 0.7875 ]	100%	0	
	Fe	[ 0.0 ~ 0.1275 ]	100%	0	
6	Fe	[ 0.0 ~ 0.1275 ]	100%	0	100%

- $r_{c_4} = \{(Ba, [0.0 \sim 0.7875])\}$
- $r_{c_5} = \{((K, [0.0 \sim 1.5525]), (Ba, [0.0 \sim 0.7875]), (Fe, [0.0 \sim 0.1275]))\}$
- $r_{c_6} = \{(Fe, [0.0 \sim 0.1275])\}$



Tabela 19 – Resultado de 4 (quatro) execuções do algoritmo CART.

(a) 1a. Execução

1a. Exec		Atributos								
		RI	Na	Mg	Al	Si	K	Ca	Ba	Fe
Clusters	1	88.5	90.0	100	92.8	84.2	100	92.8	100	75.7
	2	72.3	82.8	94.7	82.8	71.0	100	77.6	98.6	68.4
	3	76.4	70.5	100	47.0	76.4	100	100	100	76.4
	4	69.2	84.6	76.9	61.5	69.2	76.9	76.9	92.3	84.6
	5	77.7	77.7	44.4	66.6	66.6	100	55.5	100	100
	6	72.4	75.8	68.9	72.4	75.8	86.2	86.2	51.7	100

(b) 2a. Execução

2a. Exec		Atributos								
		RI	Na	Mg	Al	Si	K	Ca	Ba	Fe
Clusters	1	85.7	87.1	100	92.8	84.2	100	92.8	100	74.2
	2	76.3	86.8	96.0	82.8	64.4	100	76.3	98.6	68.4
	3	76.4	82.3	100	47.0	76.4	100	100	100	76.4
	4	76.9	84.6	76.9	69.2	69.2	76.9	76.9	92.3	84.6
	5	77.7	77.7	44.4	66.6	66.6	100	55.5	100	100
	6	72.4	75.8	65.5	72.4	75.8	93.1	93.1	51.7	100

(c) 3a. Execução

3a. Exec		Atributos								
		RI	Na	Mg	Al	Si	K	Ca	Ba	Fe
Clusters	1	88.5	85.7	100	92.8	84.2	100	92.8	100	75.7
	2	71.1	80.2	94.7	78.9	68.4	100	78.9	98.6	65.7
	3	76.4	82.3	100	58.8	76.4	100	100	100	82.3
	4	76.9	84.6	76.9	61.5	69.2	76.9	76.9	92.3	84.6
	5	77.7	77.7	44.4	66.6	66.6	100	55.5	100	100
	6	72.4	68.9	65.9	68.9	75.8	89.6	93.1	55.1	100

(d) 4a. Execução

4a. Exec		Atributos								
		RI	Na	Mg	Al	Si	K	Ca	Ba	Fe
Clusters	1	88.7	87.1	100	92.8	84.2	100	92.8	100	75.7
	2	78.9	84.2	94.7	81.5	69.7	100	76.3	98.6	65.7
	3	76.4	82.3	100	64.7	76.4	100	100	100	76.4
	4	76.9	84.6	61.5	69.2	69.2	76.9	76.9	92.3	84.6
	5	77.7	77.7	44.4	66.6	66.6	100	55.5	100	100
	6	72.4	68.9	68.9	68.9	75.8	93.1	93.1	51.7	100

## 5 Conclusões, Trabalhos Futuros e Cronograma

Este capítulo abordará as conclusões dessa proposta de mestrado referentes aos resultados do capítulo 4, bem como uma seção de Trabalhos Futuros e Cronograma. Na seção de Conclusão serão feitas considerações finais dos resultados de cada base de dados apresentadas, e logo após, em Trabalhos Futuros tem a pretensão de melhorar e expandir tudo que fora realizado nesta pesquisa, e expor que existe uma continuidade para todo esse estudo aqui elaborado. Já no Cronograma, será criada uma tabela temporal onde esta será dividida em meses e tarefas definindo os passos a serem seguidos até a conclusão da dissertação.

### 5.1 Conclusão

No capítulo 4 foram aplicados algoritmos supervisionados em algumas bases de dados a fim de provar se o problema de rotulação de dados mencionado por este trabalho foi solucionado. Uma vez conhecido o problema, foi executado dois algoritmos supervisionados servindo de amostra para provar que era possível fazer rotulação de dados com estes algoritmos (Naive Bayes e CART), tema deste trabalho. E já comparando ao trabalho de rotulação de clusters elaborado por ??) utilizando o algoritmo de Redes Neurais, este estudo demonstra de forma empírica a execução de outros algoritmos com paradigmas diferentes, para provar que essa técnica também funciona com outros algoritmos supervisionados testados.

Como o cerne da pesquisa é a rotulação de dados, foram apresentados dois algoritmos com paradigmas diferentes, e em ambos, suas execuções nas bases de dados resultaram em respostas satisfatórias no âmbito da rotulação. Embora os rótulos encontrados em cada base de dados não tenham sido totalmente idênticos, tanto um algoritmo como outro mostraram semelhanças em vários rótulos gerados, como exemplo das bases IRIS e GLASS.

O processo de rotulação é composto por um, ou vários atributos, de maior relevância entre eles junto com sua(s) faixa(s) de valor(es) que mais se repetem, conteúdo já visto na subseção 3.4.3. Seguindo esse modelo foram adicionadas a cada resultado tabelas mostrando em porcentagem o grau de correlacionamento entre os atributos. Essas tabelas tem como objetivo de passar o comportamento dos atributos através da aplicação da técnica de correlação entre eles na escolha do atributo rótulo.

No modelo de resolução proposto foi inicialmente utilizado na base de dados Seeds o

algoritmo Naive Bayes (seção 4.2.1). Onde inicialmente foi escolhido um ou vários atributos que tiveram maior valor no resultado da aplicação da técnica de correlação dos atributos na tabela 9. Após a escolha do atributo que fará parte do rótulo, o segundo passo é a escolha da faixa de valores do atributo. Essa segunda etapa é dependente totalmente da discretização, visto na seção 2.2, e independente da primeira etapa. O método é capaz de gerar a faixa de maior repetição de valores de qualquer atributo, mas nesta pesquisa a faixa escolhida é do atributo rótulo. Para ter mais confiabilidade no rótulo o método escolhe a faixa de valores que mais se repetem. No caso desse algoritmo o resultado na tabela 7 consegue provar uma boa eficiência, pois em cada 70 elementos do cluster 1, somente 14, ficaram de fora dessa faixa. No cluster 2, somando os dois atributos rótulos tem-se 12 elementos que não estão dentro da representatividade do rótulo. Outro valor pequeno em relação aos 70 elementos. E no cluster 3, somente 5 elementos não estão dentro da faixa considerada rótulo.

No cenário da execução do algoritmo CART, os resultados foram diferentes dos apresentados pelo Naive Bayes, mas nem por isso foram insatisfatórios. Contudo uma breve análise sobre as execuções das tabelas 9 e 11 podem ser observadas nos gráficos da figura 17. O comportamento dos valores do correlacionamento dos atributos ao longo das execuções mostram-se equilibradas, figura 17b. O gráfico do CART tem um movimento semelhante ao do aplicado do Naive Bayes (figura 17a), embora a variável **asymetry** saia um pouco do padrão nada alterou nos rótulos, pois seus valores são baixos, contudo o valor de **perimetro** ficou bastante encostado ao valor da **area**, fazendo o rótulo **perimetro** aparecer nos grupos 1 e 2. E também só não foi escolhido pelo grupo 3, pois a variável **Wkernel** estava com valor mais alto. E no gráfico percebe-se que **Wkernel** mantém valores altos em todas as execuções do grupo 3.

De acordo com o exposto no parágrafo anterior pode-se dizer sobre os resultados que o Naive Bayes acabou sendo um pouco melhor, pois no que diz respeito ao número de elementos fora da faixa definida pelo rótulo, o CART acabou por ter mais elementos fora da faixa de rótulo comparado aos resultados do Naive Bayes. Isso implica dizer que o rótulo deixa de representar mais elementos usando o CART ao invés do Naive Bayes, em outras palavras, o Naive Bayes representou mais elementos concordante com o rótulo do que o CART.

Já na base de dados IRIS, os dois algoritmos supervisionados testados apresentaram os mesmos rótulos nos clusters 1 e 3. Nos gráficos da figura 18 pode-se acompanhar como os valores dos atributos se comportam em seus clusters em quatro execuções.

Os algoritmos aplicados na base IRIS tem resultados nos gráficos bastantes semelhantes ao da base SEEDS, e logo percebe-se que a base IRIS contém características que possuem mais atributos bem correlacionados em relação ao da base SEEDS, pois nenhum atributo possui valor abaixo da linha 65(%) de relacionamento entre eles. Embora no

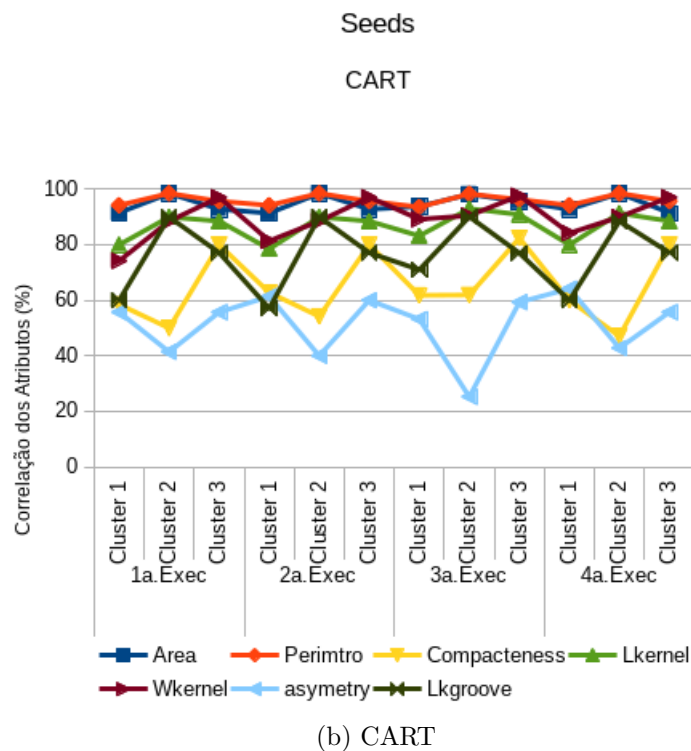
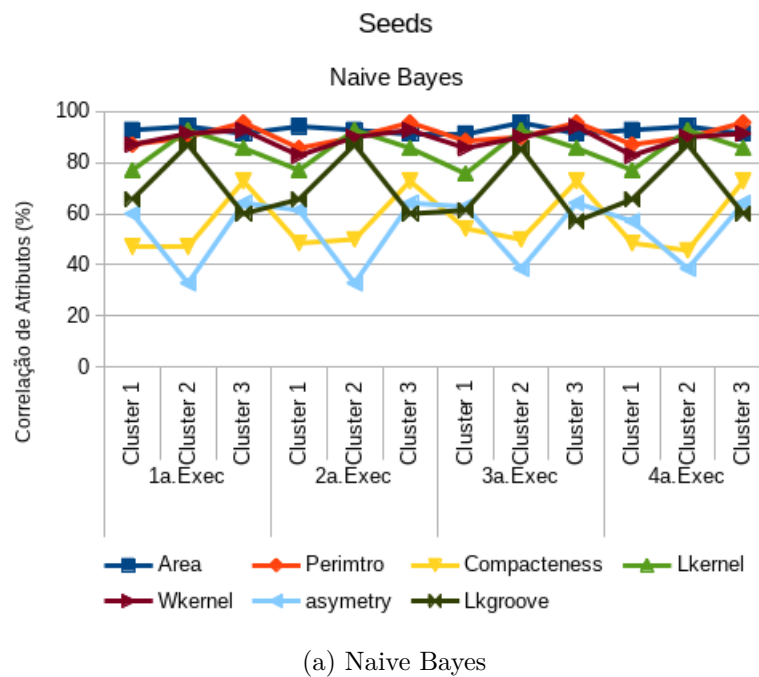


Figura 17 – Gráfico de Execuções dos algoritmos supervisionados na base de dados SEEDS.

gráfico as linhas referentes aos comportamentos dos atributos nos clusters 1 e 3 não sejam totalmente iguais em cada figura (18a e 18b), não modificou o resultado dos rótulos como resposta.

Conforme resultados das tabelas 12 e 14 apresentadas pela execução dos dois algoritmos os rótulos escolhidos no cluster 1 foram dois atributos: **petalwidth** e **petal-**

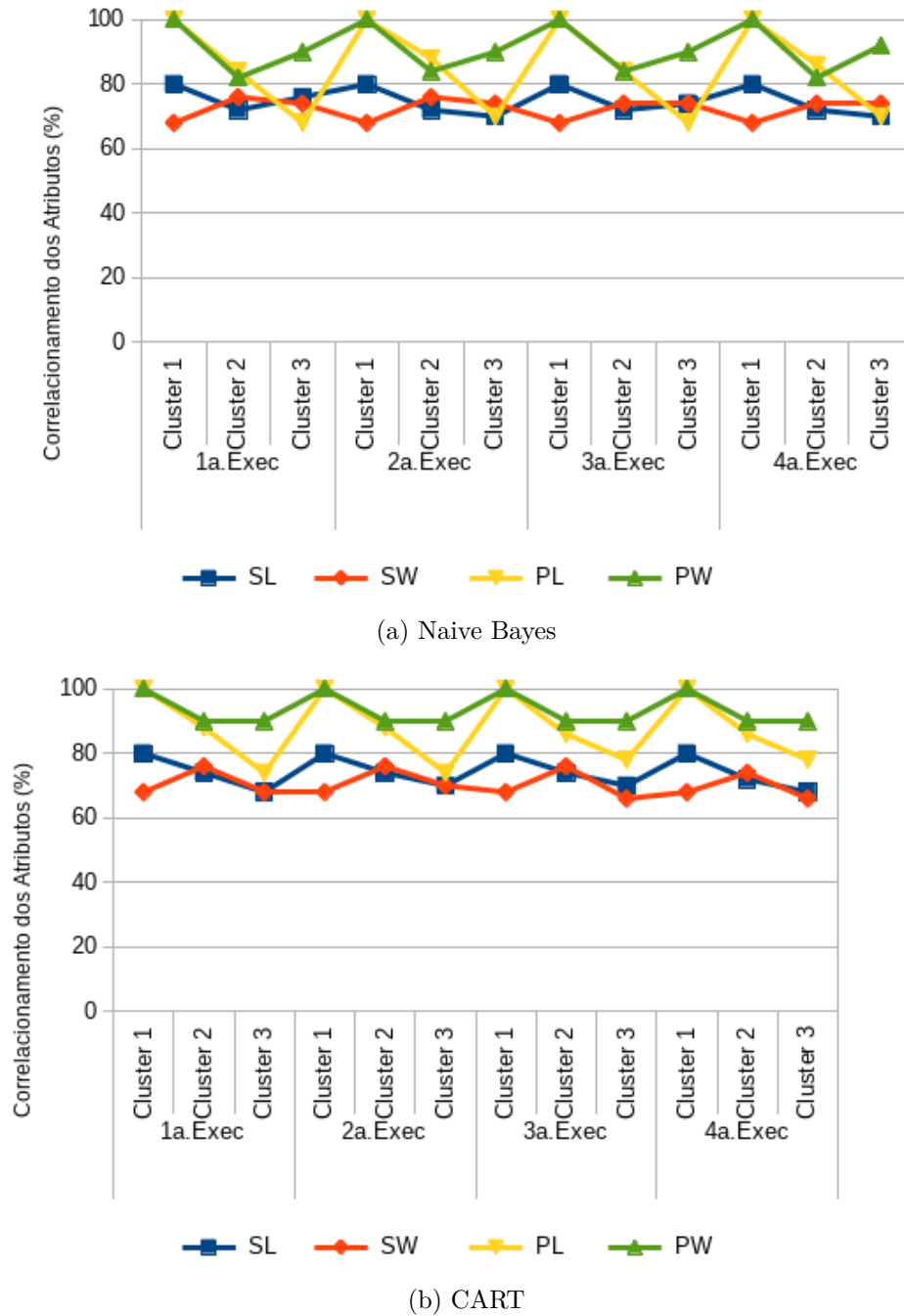


Figura 18 – Gráfico de Execuções dos algoritmos supervisionados na base de dados IRIS.

**length.** Onde cada um deles definiram faixas de valor que foi possível abranger 100% dos elementos. Já no cluster 2 cada algoritmo teve um atributo rótulo diferente, e embora não tivesse a mesma acurácia do cluster 1, obteve um total, de 86% de acurácia e deixando de representar 7 elementos do rótulo **petallength** pelo Naive Bayes, e 84% de acurácia deixando de representar 8 elementos do rótulo **petalwidth** com CART. E no cluster 3 o atributo escolhido para compor o rótulo foi o **petalwidth** em ambos os algoritmos. Logo percebe-se a importância do atributo rótulo no cluster 3, pois o rótulo representa 45 elementos no total de 50 dentro do cluster, deixando somente 5 elementos fora dessa faixa representada pelo rótulo.

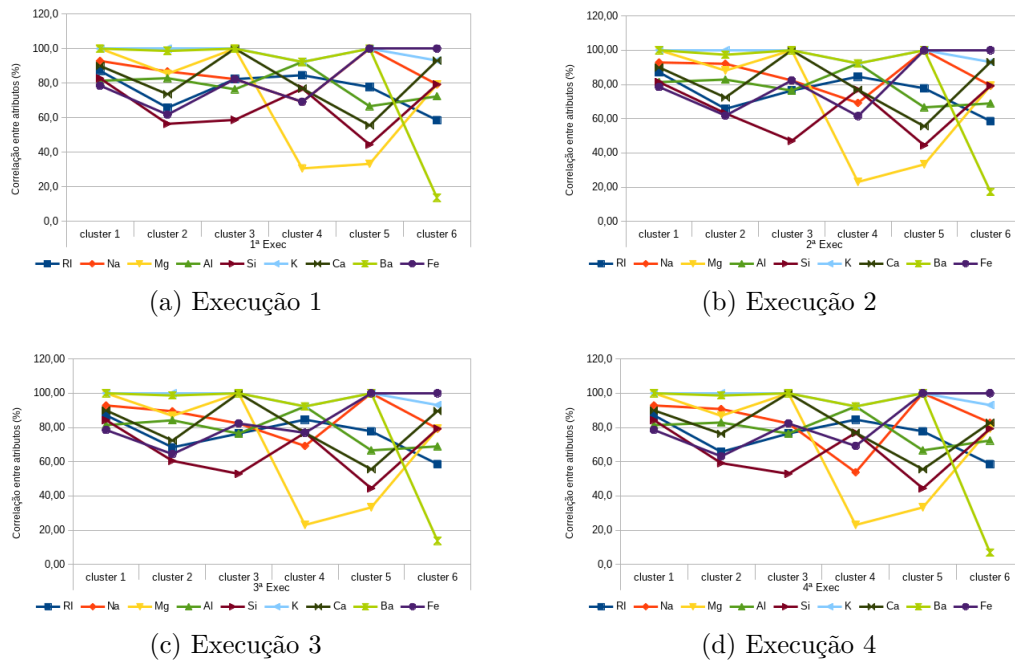


Figura 19 – Gráfico de Execuções do algoritmo supervisionado Naive Bayes na base de dados GLASS.

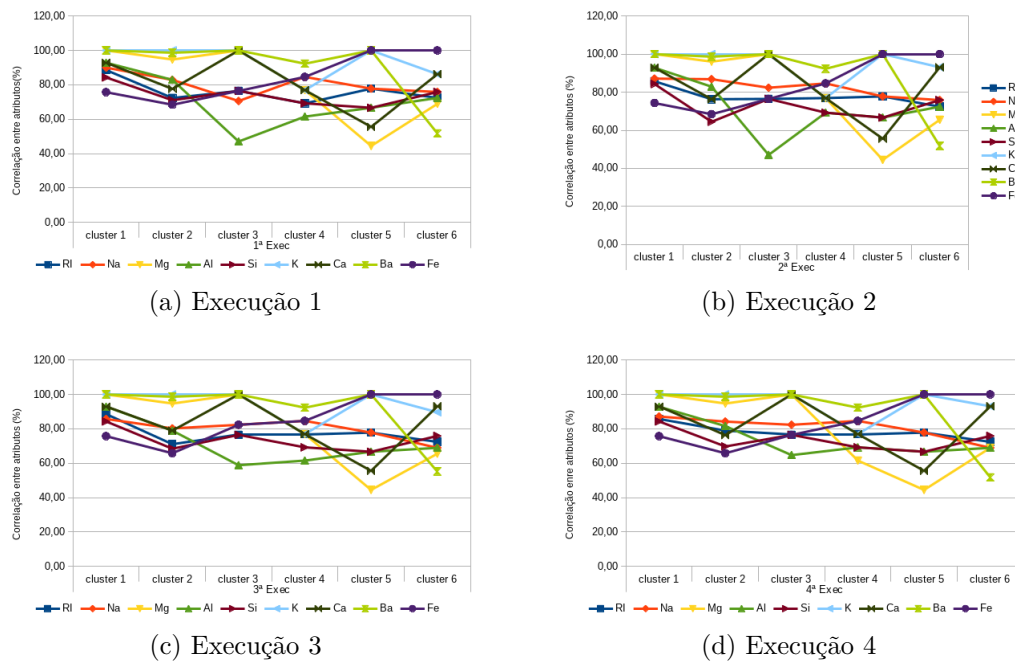


Figura 20 – Gráfico de Execuções do algoritmo supervisionado CART na base de dados GLASS.

A avaliação da base de dados GLASS referente a rotulação apresentada na tabela 16 do Naive Bayes, não foi tão bem sucedida quanto ao CART. Dos seis clusters definidos na rotulação somente dois deles não tiveram 100% de acurácia, e dentre esses dois clusters foi onde obtiveram os mais baixos valores de acurácia.

Nos gráficos da figura 19 e 20 são apresentados os comportamentos de correlacionamento dos atributos rótulos do Naive Bayes e CART respectivamente, e mesmo havendo semelhança nos gráficos os valores de correlação dos atributos no CART foram melhores, e por conseguinte teve melhor acurácia comprovado no gráfico 21.

Como conclusão, foi constatado nesta pesquisa que quanto melhores são balanceados os clusters, melhores são os resultados com o algoritmo estatístico Naive Bayes. Podendo ser comprovado nas bases SEEDS e IRIS. Já na base GLASS que é uma base mais desbalanceada pode-se notar que na figura 20, de execuções do CART, há um comportamento dos clusters melhor que no Naive Bayes. Então através de testes foi detectado que o método de rotulação de dados quando em bases mais balanceadas tiveram melhores resultados com algoritmo estatístico, e quando em bases não tão balanceadas, obtiveram melhores resultados com o algoritmo de árvore de decisão. Isso pode ser constatado no gráfico 21, onde o Naive Bayes só perde nos clusters da base GLASS.

Por fim, ao analisar os resultados após a aplicação dos dois algoritmos supervisionados pode-se afirmar que é possível fazer rotulação de cluster, conforme resultados demonstrado na figura 21. Através desta figura visualiza-se uma acurácia de 80% na maioria dos resultados, provando que os rótulos encontrados representam bem os clusters testados.

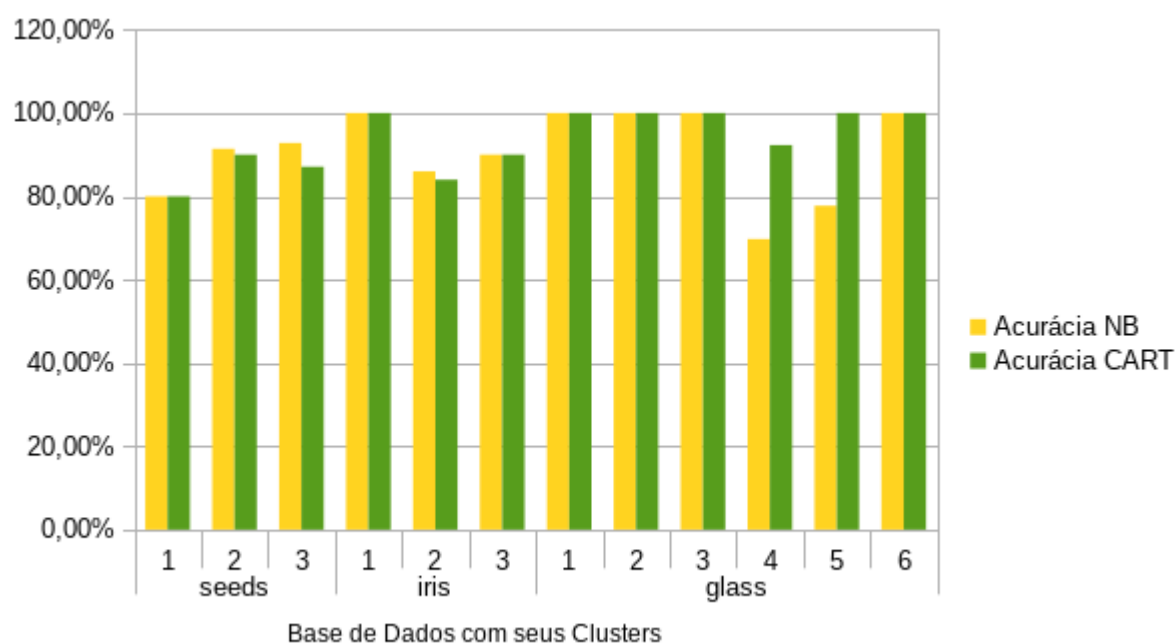


Figura 21 – Acurácia por Clusters (Os clusters estão numerados em ordem crescente em cada Base de Dados)

No trabalho de ??) é utilizado um processo de agrupamento da dados para criação de clusters, portanto seus clusters diferem em números de registros comparado ao desta pesquisa.

Tabela 20 – Resultado da rotulação utilizando Redes Neurais (??) referente a base de dados SEEDS.

Cluster	Num_Elem	Atributo	Erro
1	67	A	8
		P	9
2	82	A	12
		P	10
3	61	P	0
		WK	3
		LK	1
		A	0
Total			43

Tabela 21 – Resultado da rotulação utilizando Naive Bayes referente a base de dados SEEDS.

Cluster	Num_Elem	Atributo	Erro
1	70	A	14
2	70	A	6
3	70	P	5
Total			25

Tabela 22 – Resultado da rotulação utilizando CART referente a base de dados SEEDS.

Cluster	Num_Elem	Atributo	Erro
1	70	P	14
2	70	A	6
		P	7
3	70	WK	9
Total			36

Por apresentar essas características, resolveu-se apresentar uma tabela comparativa onde o principal argumento é o número de registros que não estão sendo representados pelo rótulo denominado nas colunas das tabelas como **Fora da Faixa** ou **Erro**, em razão disso foi possível fazer comparações entre os trabalhos.

A métrica destacada nesta análise comparativa, na base de dados SEEDS, leva em consideração o total de erros. Nesta situação as tabelas 21 e 22 obtiveram um número menor de erros, atestando que o modelo desta pesquisa adquiriu bons resultados em comparação a tabela 20.

## 5.2 Trabalhos Futuros

A pesquisa ainda precisa de mais divulgação na esfera acadêmica, e para isso a publicação de um artigo sobre os resultados apresentados aqui é uma consolidação dessa



proposta de mestrado já voltada para a dissertação propriamente dita.

Fazer testes com mais bases de dados provando que esse método pode ser utilizado em várias bases com características diferentes.

Outro ponto importante é inserir nos teste mais algoritmos, que pertençam a paradigmas diferentes dos que já foram utilizados.

## 5.3 Cronograma

Tabela 23 – Cronograma de atividades

Atividades	Meses		
	Junho	Julho	Agosto
1. Testes com Novas Bases de Dados			
2. Modificar Números de Faixa (R)			
3. Testar com outros Algoritmos			
4. Preparar Artigo			
5. Escrita da Dissertação			

No primeiro item, serão adicionados testes com novas bases que possuam características diferentes quanto ao balanceamento de atributos, bases com muito atributos, bases com muito registros verificando a viabilidade de processamento.

No segundo item serão realizados mais teste com números de faixas diferentes, pois com o processo de discretização o número de faixa poderá influenciar diretamente no ganho de informação.

No terceiro item seria a realização de testes com novos algoritmos suportados pela *Statistic and Machine Learning Toolbox* e comparar os resultados.

No último mês seria a dedicação para preparação da dissertação deste trabalho (item quatro), e por fim, a escrita de um artigo referente a rotulação de dados (item 5) com os testes apresentados nesta pesquisa.

# Referências

ARAÚJO, F. N. C. *Rotulação Automática de Clusters Baseados em Análise de Filogenias*. Teresina - PI: [s.n.], 2018. 48 p. Citado na página 21.

BARBER, D. *Bayesian Reasoning and Machine Learning*. [s.n.], 2011. ISSN 9780521518147. ISBN 9780511804779. Disponível em: <<http://ebooks.cambridge.org/ref/id/CBO9780511804779>>. Citado 2 vezes nas páginas 5 e 13.

BREIMAN, L. et al. *Classification and Regression Trees*. Taylor & Francis, 1984. (The Wadsworth and Brooks-Cole statistics-probability series). ISBN 9780412048418. Disponível em: <<https://books.google.com.br/books?id=JwQx-WOmSyQC>>. Citado na página 7.

CATLETT, J. *On changing continuous attributes into ordered discrete attributes*. Springer, Berlin, Heidelberg: Springer Verlag, 1991. 164–178 p. Citado 2 vezes nas páginas 14 e 29.

CHARYTANOWICZ, M. et al. Complete gradient clustering algorithm for features analysis of X-ray images. *Advances in Intelligent and Soft Computing*, v. 69, p. 15–24, 2010. ISSN 18675662. Citado na página 35.

CHEN, C.-L.; TSENG, F. S. C.; LIANG, T. An integration of fuzzy association rules and WordNet for document clustering. *Knowledge and Information Systems*, v. 28, n. 3, p. 687–708, sep 2011. ISSN 0219-1377. Disponível em: <<http://link.springer.com/10.1007/s10115-010-0364-2>>. Citado 2 vezes nas páginas 18 e 19.

COSTA, B. S. J. et al. Unsupervised classification of data streams based on typicality and eccentricity data analytics. *2016 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2016*, p. 58–63, 2016. Citado na página 19.

DOUGHERTY, J.; KOHAVI, R.; SAHAMI, M. Supervised and Unsupervised Discretization of Continuous Features. *Machine Learning Proceedings 1995*, Stanford, v. 0, p. 194–202, 1995. ISSN 0717-6163. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/B9781558603776500323>>. Citado na página 14.

EVETT, I. W.; SPIEHLER, E. J. Knowledge based systems. In: DUFFIN, P. H. (Ed.). New York, NY, USA: Halsted Press, 1988. cap. Rule Induction in Forensic Science, p. 152–160. ISBN 0-470-21260-8. Disponível em: <<http://dl.acm.org/citation.cfm?id=67040.67055>>. Citado na página 42.

FILHO, V. P. R.; MACHADO, V. P.; LIRA, R. d. A. Rotulação de Grupos Utilizando Conjuntos Fuzzy. In: UNIVERSIDADE FEDERAL DO PIAUÍ. *XII Simpósio Brasileiro de Automação Inteligente (SBAI)*. Natal, RN, 2015. Disponível em: <<http://pubs.acs.org/doi/abs/10.1021/ja103937v>>. Citado 2 vezes nas páginas 21 e 39.

FISHER, R. A. the Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, v. 7, n. 2, p. 179–188, 1936. ISSN 20501420. Disponível em: <<http://doi.wiley.com/10.1111/j.1469-1809.1936.tb02137.x>>. Citado na página 39.

- GAN, H. et al. Using clustering analysis to improve semi-supervised classification. *Neurocomputing*, v. 101, p. 290–298, feb 2013. ISSN 09252312. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0925231212006601>. Citado na página 19.
- HWANG, G. J.; LI, F. A Dynamic Method for Discretization of Continuous Attributes. *Lecture Notes in Computer Science - Intelligent Data Engineering and Automated Learning - IDEAL 2002: Third International Conference*, v. 2412/2002, p. 506, 2002. ISSN 16113349. Disponível em: <http://www.springerlink.com/content/4n05b2n6x0cx4tlk>. Citado 2 vezes nas páginas 14 e 29.
- IMPERES, F. d. C. *Rotulação de Grupos em Algoritmos de Agrupamento Baseados em Distância Utilizando Grau de Pertinência*. 2018. 60 p. Citado na página 21.
- IWAMURA, M.; TSUKADA, M.; KISE, K. Automatic Labeling for Scene Text Database. In: *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 2013. p. 1365–1369. ISBN 978-0-7695-4999-6. Disponível em: <http://ieeexplore.ieee.org/document/6628837/>. Citado na página 19.
- JIRASIRILERD, W.; TANGTISANON, P. Automatic Labeling for Thai News Articles Based on Vector Representation of Documents. In: *2018 International Conference on Engineering, Applied Sciences, and Technology (ICEAST)*. IEEE, 2018. p. 1–4. ISBN 978-1-5386-4956-5. Disponível em: <https://ieeexplore.ieee.org/document/8434457/>. Citado 2 vezes nas páginas 18 e 19.
- KOTSIANTIS, S.; KANELLOPOULOS, D. Discretization Techniques : A recent survey. *GESTS International Transactions on Computer Science and Engineering*, v. 32, n. 1, p. 47–58, 2006. Citado na página 14.
- KUMAR, A.; ANDU, T.; THANAMANI, A. S. Multidimensional Clustering Methods of Data Mining for Industrial Applications. *International Journal of Engineering Science Invention*, v. 2, n. 7, p. 1–8, 2013. Citado na página 1.
- LACHI, R. L.; ROCHA, H. V. *Aspectos básicos de clustering: conceitos e técnicas*. Campinas, SP, 2005. 1–26 p. Disponível em: <http://www.ic.unicamp.br/~reltech/2005/05-03.p>. Citado na página 11.
- LIMA, B. V. A. Dissertação (Programa de Pós-graduação em Ciência da Computação), *Método Semissupervisionado de Rotulação e Classificação Utilizando Agrupamento por Sementes e Classificadores*. Teresina - PI: [s.n.], 2015. 47 p. Citado 2 vezes nas páginas 20 e 21.
- LIMA, B. V. A. de; MACHADO, V. P.; LOPES, L. A. Automatic labeling of social network users Scientia.Net through the machine learning supervised application. *Social Network Analysis and Mining*, v. 5, n. 1, p. 44, dec 2015. ISSN 1869-5450. Disponível em: <http://link.springer.com/10.1007/s13278-015-0285-x>. Citado 2 vezes nas páginas 20 e 21.
- LOPES, L. A.; MACHADO, V. P.; RABELO, R. D. A. L. Automatic Labeling of Groupings through Supervised Machine Learning. *Knowledge-Based Systems*, v. 106, p. 231–241, 2016. Citado 10 vezes nas páginas 9, 1, 2, 15, 17, 20, 21, 24, 26 e 29.

LUCCA, G. et al. Uma implementação do algoritmo Naïve Bayes para classificação de texto. In: CENTRO DE CIÊNCIAS COMPUTACIONAIS DA UNIVERSIDADE FEDERAL DO RIO GRANDE. *IX Escola Regional de Banco de Dados – ERBD 2013*. Rio Grande - RS, 2013. p. 1–4. Disponível em: <<http://ifc-camboriu.edu.br/erbd2013>>. Citado na página 9.

MADUREIRA, D. F. *Análise de sentimento para textos curtos*. Tese (Doutorado) — Fundacao Getulio Vargas, Rio de Janeiro, 2017. Citado na página 9.

MCCALLUM, A.; NIGAM, K. A Comparison of Event Models for Naive Bayes Text Classification. 1997. Citado na página 9.

MITCHELL, T. M. *Machine learning*. [S.l.]: McGraw-Hill Science/Engineering/Math, 1997. 432 p. ISSN 10450823. ISBN 9781577354260. Citado 2 vezes nas páginas 4 e 8.

MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. *Foundations Machine Learning*. [S.l.: s.n.], 2012. ISBN 9780262018258. Citado na página 5.

RAIMUNDO, L. R.; MATTOS, M. C. D.; WALESKA, P. O Algoritmo de Classificação CART em uma Ferramenta de Data Mining. *IV Congresso Sul Brasileiro de Ciências da Computação - SULCOMP*, 2008. Citado na página 7.

RUSSEL, S.; NORVIG, P. *Inteligência Artificial*. 3ª. ed. Rio de Janeiro: [s.n.], 2013. ISBN 9780136042594. Citado 4 vezes nas páginas 4, 5, 6 e 9.

SUN, L.; YOSHIDA, S.; LIANG, Y. A support vector and k-means based hybrid intelligent data clustering algorithm. *IEICE Transactions on Information and Systems*, E94-D, n. 11, p. 2234–2243, 2011. ISSN 17451361. Citado na página 19.

WU, X. et al. *Top 10 algorithms in data mining*. [S.l.: s.n.], 2008. v. 14. 1–37 p. ISSN 02191377. ISBN 1011500701. Citado na página 9.

YEGANOVA, L.; COMEAU, D. C.; WILBUR, W. J. Identifying Abbreviation Definitions Machine Learning with Naturally Labeled Data. In: *2010 Ninth International Conference on Machine Learning and Applications*. IEEE, 2010. p. 499–505. ISBN 978-1-4244-9211-4. Disponível em: <<http://ieeexplore.ieee.org/document/5708877/>>. Citado 2 vezes nas páginas 18 e 19.

YOHANNES, Y.; WEBB, P. *Classification and Regression Trees, CART: A User Manual for Identifying Indicators of Vulnerability to Famine and Chronic Food Insecurity*. International Food Policy Research Institute, 1999. (Microcomputers in policy research). ISBN 9780896293373. Disponível em: <<https://books.google.com.br/books?id=7iuq4ikyNdoC>>. Citado na página 7.