



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

Rotulação com Algoritmos Supervisionados

Tarcísio Franco Jaime

Número de Ordem PPGCC: M001

Teresina-PI, Janeiro de 2017

Tarcísio Franco Jaime

Rotulação com Algoritmos Supervisionados

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

Janeiro de 2017

Tarcísio Franco Jaime

Rotulação com Algoritmos Supervisionados/ Tarcísio Franco Jaime. – Teresina-PI, Janeiro de 2017-

38 p. : il. (algumas color.) ; 30 cm.

Orientador: Vinicius Ponte Machado

Qualificação (Mestrado) – Universidade Federal do Piauí – UFPI

Centro de Ciências da Natureza

Programa de Pós-Graduação em Ciência da Computação, Janeiro de 2017.

1. Rotulação. 2. Algoritmos Supervisionados. 3. CART. 4. Naive Bayes. I. Vinicius Ponte Machado. II. Universidade Federal do Piauí. III. Rotulação com Algoritmos Supervisionados.

CDU 02:141:005.7

Tarcísio Franco Jaime

Rotulação com Algoritmos Supervisionados

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.

Trabalho aprovado. Teresina-PI, 01 de janeiro de 2018:

Vinicius Ponte Machado
Orientador

Co-Orientador

Professor
Convidado 1

Professor
Convidado 2

Professor
Convidado 3

Teresina-PI
Janeiro de 2017

*Aos meus pais XXXXXXXX e YYYYYYY,
por sempre estarem comigo em todos os momentos.*

Agradecimentos

Agradeço a Deus.

Agradeço aos meus pais, XXXXX e YYYYY, por ...

Aos meus irmãos, por.....

Agradeço ao meu orientador, XXXXXXXXX, por todos os conselhos, pela paciência e ajuda nesse período.

Aos meus amigos ...

Aos professores ...

À XXXXXX pelo apoio financeiro para realização deste trabalho de pesquisa.

*“Não sei o que,
não sei o que,
não sei o que lá.”
(Autor Desconhecido)*

Resumo

Segundo a ABNT, o resumo deve ressaltar o objetivo, o método, os resultados e as conclusões do documento. A ordem e a extensão destes itens dependem do tipo de resumo (informativo ou indicativo) e do tratamento que cada item recebe no documento original. O resumo deve ser precedido da referência do documento, com exceção do resumo inserido no próprio documento. (...) As palavras-chave devem figurar logo abaixo do resumo, antecidas da expressão Palavras-chave:, separadas entre si por ponto e finalizadas também por ponto.

Palavras-chaves: latex. abntex. editoração de texto.

Abstract

This is the english abstract.

Keywords: latex. abntex. text editoration.

Lista de ilustrações

Figura 1 – Hipóteses ajustadas	7
Figura 2 – Ponto de Corte (R-1)	11
Figura 3 – Discretização EWD	11
Figura 4 – Discretização EFD	12
Figura 5 – Modelo (LOPES; MACHADO; RABELO,)	13
Figura 6 – Comportamento da base de dados a cada iteração. Método (FILHO, 2015)	13
Figura 7 – Modelo (FILHO, 2015)	14
Figura 8 – Modelo de Resolução Proposto	17
Figura 9 – Exemplo da técnica aplicada ao atr1 sendo classe	18
Figura 10 – Discretização de atributos utilizando EFD com $R = 3$	19
Figura 11 – Exemplo da técnica aplicada aos 3(<i>três</i>) atributos, cada um sendo classe em determinada iteração	21
Figura 12 – Resultado dos Algoritmos	21
Figura 13 – Gráfico de Execuções dos algoritmos supervisionados na base de dados SEEDS.	34
Figura 14 – Gráfico de Execuções dos algoritmos supervisionados na base de dados IRIS.	35

Lista de tabelas

Tabela 1	– Base de Dados Modelo	19
Tabela 2	– Base de Dados Modelo Discretizada	20
Tabela 3	– Resultado da aplicação do algoritmo Naive Bayes	26
Tabela 4	– Resultado da Correlação dos atributos pelo Naive Bayes; Legenda dos Atributos: (A)area, (B)perimetro, (C)compacteness, (D)Lkernel, (E)Wkernel, (F)asymetry, (G)lkgroove	27
Tabela 5	– Resultado de 4(<i>quatro</i>) execuções do algoritmo Naive Bayes; Legenda dos Atributos: (A)area, (B)perimetro, (C)compacteness, (D)Lkernel, (E)Wkernel, (F)asymetry, (G)lkgroove	27
Tabela 6	– Resultado da aplicação do algoritmo CART	28
Tabela 7	– Resultado da Correlação dos atributos pelo CART; Legenda dos Atributos: (A)area, (B)perimetro, (C)compacteness, (D)Lkernel, (E)Wkernel, (F)asymetry, (G)lkgroove	29
Tabela 8	– Resultado de 4(<i>quatro</i>) iterações do algoritmo CART; Legenda dos Atributos: (A)area, (B)perimetro, (C)compacteness, (D)Lkernel, (E)Wkernel, (F)asymetry, (G)lkgroove	29
Tabela 9	– Resultado da aplicação do algoritmo Naive Bayes	30
Tabela 10	– Resultado de 4(<i>quatro</i>) execuções do algoritmo Naive Bayes; Legenda dos Atributos: (SL)sepallength,(SW)sepalwidth,(PL)petallength,(PW)petalwidth	31
Tabela 11	– Resultado da aplicação do algoritmo CART	32
Tabela 12	– Resultado de 4(<i>quatro</i>) iterações do algoritmo CART; Legenda dos Atributos: (SL)sepallength,(SW)sepalwidth,(PL)petallength,(PW)petalwidth	32
Tabela 13	– Cronograma de atividades	36

Lista de abreviaturas e siglas

EWD	Discretização por Larguras Iguais
EFD	Discretização por Frequências Iguais
CART	Algoritmo Classification and Regression Trees

Sumário

	Introdução	1
Motivação		1
	Objetivo	2
	Rotulação	2
	Disposição do Documento	3
1	REFERENCIAL TEÓRICO	5
1.1	Aprendizado de Máquina	5
1.1.1	Aprendizado Supervisionado	6
1.1.1.1	Algoritmo Classification and Regression Trees - CART	7
1.1.1.2	Algoritmo Naive Bayes	8
1.1.2	Aprendizado Não-Supervisionado	9
1.2	Discretização	10
1.2.1	Discretização por Larguras Iguais - EWD	10
1.2.2	Discretização por Frequência Iguais - EFD	11
1.3	Trabalhos Correlatos	12
2	METODOLOGIA / MATERIAIS E MÉTODOS	15
2.1	Considerações do Problema	15
2.2	O Modelo de Resolução	16
2.3	Técnica de Correlação entre Atributos	17
2.4	Exemplo com Base Modelo Fictícia	18
2.4.1	Processo (I) - Discretização	18
2.4.2	Processo (II) - Algoritmos Supervisionados	20
2.4.3	Processo (III) - Rotulação	22
3	RESULTADOS	25
3.1	Implementação	25
3.2	Seeds - Identificação de Tipos de Semente	26
3.2.1	Naive Bayes	26
3.2.2	CART	28
3.3	Iris - Identificação de Tipos de Plantas	29
3.3.1	Naive Bayes	30
3.3.2	CART	31

Conclusões	33
Trabalhos Futuros	36
Cronograma	36

REFERÊNCIAS	37
-----------------------	----

Introdução

Esta pesquisa apresenta uma proposta de mestrado envolvendo como tema principal a rotulação de dados com algoritmos supervisionados. O foco é apresentar através de amostragem, que é possível fazer rotulação de dados com quaisquer algoritmos supervisionados. Sabendo da inviabilidade deste trabalho em testar todos os algoritmos supervisionados, foram realizados testes utilizando algoritmos com paradigmas diferentes para poder confirmar a proposta deste estudo.

Motivação

Embora rotulação de dados seja uma área bem definida, o modo de abordagem aplicado nesta pesquisa vem alterando a maneira de como Aprendizagem de Máquina define este termo, chamado rotulação. Em pesquisas realizadas nesta área, sob supervisão do orientador deste trabalho, e outros orientandos do mesmo, vem desenvolvendo estudos e definindo rotulação como sendo algo diferente da classificação dos dados. Apesar de várias literaturas ([BARBER, 2011](#); [MITCHELL, 1997](#)) entre outras citarem o termo rotulação como um sinônimo de classificação, aqui é não é tratado assim. Muitos trabalhos feitos aqui neste laboratório estão redefinindo o termo rotulação, como um complemento da classificação, possuindo propriedades diferentes das apresentadas na classificação.

A classificação é dada com um identificador do registro conforme suas características. Um especialista através de seu conhecimento técnico avalia esses registros e os define por algum classificador: macho ou fêmea, raça X, Y ou Z, etc. Mas na rotulação a intenção é pegar estes dados já classificados e definir algum significado para esses grupos.

Os clusters são formados por algum tipo de similaridade entre os atributos que o compõe, e tão diferentes quanto os atributos que pertencem a clusters distintos. E embora os atributos já estivessem agrupados, seria importante esses grupos possuírem algum tipo de rótulo, onde o próprio analista ao percebê-los pudesse tecnicamente notar um significado nos grupos, e com essa informação, poder tomar alguma decisão sobre eles.

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 um base de dados pelas operadoras. Uma vez em posse desses dados, são criados clusters. E seria interessante a aplicabilidade da rotulação para personalizar alguns serviços para esses grupos já formados. Então, caso o rótulo de um cluster fosse o atributo localização, e os valores desse atributo escolhido para compor o rótulo, fossem as coordenadas geográficas, que definiriam a localizações

de academias. Logo percebe-se que os participante desse grupo possuem característica de frequentar academias. Já em posse dessa informação seria inteligente, da parte do analista, tomar alguma decisão personalizada para este grupo.

Objetivo

Muito se tem escrito sobre algoritmos de agrupamentos de dados, mas poucos falam em dar significado aos grupos formados. Apesar de parecer lógico que a formação do cluster é por algum tipo de correlação das características do grupo, não fica claro qual característica se apresenta mais significativa dentro desse grupo. E este trabalho tem como cerne apresentar essa característica de maior importância no grupo , aplicando a técnica de rotulação de dados.

Rotulação

Dentre o modelo de rotulação duas etapas são ditas como principais: discretização e correlação de atributos. Nessa técnica o rótulo é composto pelo resultado dessas duas etapas. Na correlação de atributos é onde acontece a aplicação do algoritmo supervisionado na base de dados, e gerado uma tabela como resultado. Nessa tabela as linhas representam os grupos, e as colunas os atributos, contudo os valores são expressos em porcentagens informando o grau de relevância do atributo em relação aos outros.

Na segunda etapa os valores passam por uma técnica de discretização e após isso, já com o atributo escolhido através da tabela de correlação, verifica-se qual valor da faixa mais se repete neste atributo. Uma vez escolhido a faixa, é apresentado os limites dessa faixa, e com esses valores o rótulo é expresso por um conjunto de tuplas formadas pelo atributo, de maior relevância, e com os limites da faixa que mais se repetem.

O rótulo do cluster é uma composição do(s) atributo(s) mais bem relacionados com a faixa de valor que mais se repete dentro na coluna do atributo escolhido. E mesmo com cunho semelhante, o trabalho de ([LOPES; MACHADO; RABELO](#),) fez o método de rotulação utilizando um algoritmo com paradigma conexionista, diferente ao empregado aqui. Mas nesta pesquisa, será aplicado dois algoritmos com paradigmas diferentes servindo de amostragem para provar que a técnica de rotulação funciona para quaisquer algoritmos supervisionados. Os algoritmos escolhidos foram: Naive Bayes (paradigma estatístico) e o CART (paradigma simbólico).

Disposição do Documento

Na sessão 2.1 é abordada a definição do problema da pesquisa. A partir da definição do problema um modelo de resolução é apresentado através do fluxograma, figura 8. Neste modelo é aplicado as duas etapas principais: Processos (I) e (II). Essas fases serão utilizadas no processo de rotulação (III). No processo (I) o cluster é submetido a uma técnica de discretização, dentre as duas apresentadas: Discretização por Frequências Iguais - EFD ou Discretização por Larguras Iguais - EWD. A outra etapa, processo (II), é aplicado o algoritmo supervisionado na base de dados e gerado uma tabela, em porcentagem, de valores com seus relacionamentos entre os outros atributos, onde quanto maior o valor mais bem correlacionado o atributo é em relação aos outros.

Ainda na sessão 2, uma base de dados fictícia é utilizada para exemplificar a execução dos processos do modelo de resolução. Esta base é dividida em 3(*três*) classes e submetida aos dois algoritmos supervisionados, gerando os rótulos de cada cluster.

Em seguida os resultados são apresentados por cada base de dados. Os algoritmos supervisionados são aplicados a cada base de dados e gerados os rótulos. E por fim a conclusão onde é realizado uma interpretação dos resultados.

1 Referencial Teórico

Será abordado neste capítulo o conteúdo base na compreensão deste trabalho dividido em 3 sessões: Aprendizado de Máquina, Discretização e Trabalhos Correlatos.

A primeira seção contempla os principais tipos de aprendizados indutivos dando ênfase ao aprendizado supervisionado, foco da proposta deste mestrado. O aprendizado indutivo utiliza uma amostra do todo para tirar uma conclusão. Caso os exemplos retirados de uma base de dados não forem suficientes, talvez o conhecimento derivado destes exemplos não mostrem a verdade.

O segundo item dissertará sobre a técnica de discretização adotada nesta pesquisa. Possuindo grande contribuição para os resultados gerados, e ganhando assim uma seção própria para explanação de como funciona essa técnica. Já na terceira seção, serão abordados trabalhos com mesmas características particulares para melhor elucidar o motivo da elaboração dessa proposta de mestrado.

1.1 Aprendizado de Máquina

Aprendizagem de máquina é a capacidade do aprendizado automático com utilização de algoritmos atuando em cima de uma base de dados. Diz-se que o computador está aprendendo quando existe uma melhora de desempenho de tarefas que ele utilizou como exemplo (MITCHELL, 1997). Um exemplo seria a realização do reconhecimento facial de uma pessoa utilizando aprendizado de máquina. Não seria necessário a implementação de várias linhas de código informando, que a cor dos olhos são azuis com orelhas e cabelos grandes seriam de uma certa pessoa. Ao invés disso é observada várias fotos tituladas de uma certa pessoa, e após vários exemplos o computador seria capaz de prever uma foto nova, se é, ou não, da determinada pessoa através de aprendizado anterior.

Existem alguns motivos, onde justificam, que não é possível simplesmente exigir que o projetista implemente melhorias no sistema de forma que ele esteja robusto bastante para lidar com todas as situações (RUSSEL; NORVIG, 2013). Um desses motivos seria a incapacidade da antecipação de todas as situações possíveis de implementação por parte do programador. Fazendo um resumo, aprendizado de máquina seriam algoritmos capazes de aprender automaticamente através de determinados exemplos, ou comportamentos.

A partir desta síntese, tem-se uma observação. A classificação de dados no contexto de aprendizado de máquina, são compostos por dois pilares. Um, seriam os **dados** a serem classificados, e outro, o **algoritmo** que irá atuar nessa base de dados. Existem vários algoritmos como exemplo: redes neurais, árvores de decisão, Suport Vector Machine –

SVM, etc. Qualquer um destes algoritmos são utilizados para solucionar essa classificação. E a escolha apropriada, desse algoritmo, se dará através de métricas que avaliarão o desempenho de cada um, e a melhor métrica, será o algoritmo apropriado para aquele problema de classificação de dados.

Uma analogia referente do que foi dito acima seria um “problema”, comparado a um “motor”, e os algoritmos disponíveis seriam as "ferramentas" para concertar esse motor. A partir daí a ferramenta que fosse mais eficaz, considerando métricas de desempenho, para fazer o motor funcionar, seria a ferramenta(algoritmo) escolhida. Tendo assim a escolha certa para um determinado problema.

1.1.1 Aprendizado Supervisionado

Nesta seção será abordado um método que através de uma banco de dados já classificado por especialistas, será feita uma predição de novos registros com base em vários desses exemplos já classificados. Os responsáveis por essas predições de novos registros são algoritmos de aprendizado supervisionados projetados para determinados fins.

O termo "Supervisionado" indica que existe um supervisor para cada registro de entrada especificando uma saída para esse registro. Considerando uma base de dados de imagens de rostos, onde cada imagem possui uma saída representado 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, uma vez que são inúmeras as diferenças que difere as faces masculinas e femininas. Mas uma alternativa seria dar exemplos de rostos com suas classificações de fazer que automaticamente a máquina "aprenda" uma regra para predizer se é masculino ou feminino (BARBER, 2011).

Em (RUSSEL; NORVIG, 2013) os autores fazem uma apresentação formal do funcionamento da aprendizagem supervisionada. Dado um conjunto de treinamento

$$(x_1, y_1), (x_2, y_2), \dots (x_n, y_n), \quad (1.1)$$

onde cada y_j foi gerado por $y = f(x)$ desconhecida. Encontrar uma função h que se aproxime da função f real.

A função h é uma hipótese onde prevê um melhor desempenho entre as hipóteses possíveis através dos conjuntos de exemplos, que são diferentes do conjunto de treinamento 1.1.

Na figura 1a existe um sobre ajuste da função com o conjunto de dados de treinamento. Esse exemplo acabou exibindo uma função mais complexa para se molda de acordo com os sete pontos do gráfico, especificando para esse conjunto de dados.

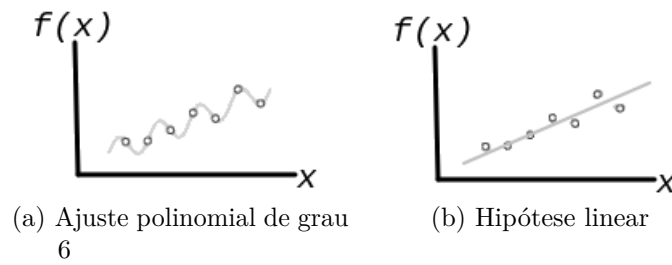


Figura 1 – Hipóteses ajustadas

Ja na figura 1b o ajuste da função se torna mais simples e mesmo não passando por todos os pontos, acabou generalizando melhor o conjunto de treinamento, tornando talvez, um melhor resultado da predição de novos valores.

A figura 1 mostra duas hipóteses que tentam se aproximar ao máximo da função verdadeira, que é desconhecida. Mesmo parecendo que na figura 1a obteve-se melhor resultado, pois todos os pontos são contemplados pela função, mas esta função h acabou ficando muito específica e isso não retrata os dados em um mundo real. Então quanto mais generalizado for h , melhor será para prevê os valores de y para novos conjuntos de dados.

Antes de falar dos algoritmos utilizados nesse texto a aprendizagem supervisionada detém dois tipos de caso: regressão e classificação. A classificação, contém variáveis com valores discretos, onde as amostras destas variáveis de saída estão na forma de categorias. Como exemplo poderia ser masculino e feminino. Já no tipo regressão, possuem valores contínuos: quantidade de água em ml, velocidade de um carro, altura de uma pessoa.

1.1.1.1 Algoritmo Classification and Regression Trees - CART

Esse algoritmo constroi modelos de previsão a partir de dados de treinamento onde seus resultados podem ser representados em uma árvore de decisão. No caso de não ser probabilístico o grau de confiança em seu modelo de predição será embasada em respostas semelhantes em outras circunstâncias antes analisadas.

Inicialmente todas as amostras se concentram no nó raiz, e a partir daí é apresentado uma questão, onde a intenção é separar o nó raiz em dois grupos mais homeogêneos. Dependendo da questão as amostras irão para a folha esquerda ou direita do nó raiz.

O CART faz essa divisão em função da regra Gini¹ (BREIMAN, 1984), parecida com a regra da entropia usada no algoritmo ID3². O índice Gini varia de 0 a 1, definindo o grau de pureza do nó.

$$Gini(S) = 1 - \sum p^2(j/t) \quad (1.2)$$

¹ O CART pode utilizar outros critérios de divisão de dados como: entropia e critério de Twoing

² Algoritmo abordado por (QUINLAN, 1986)

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

Para construção de uma árvore existem três componente importantes (YOHANNES; WEBB, 1999):

- 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;

Abaixo segue um algoritmo de como o critério Gini é aplicado nas variáveis (RAIMUNDO; MATTOS; WALESKA, 2008):

Algorithm 1: Rotina de funcionamento do CART

```

1 melhorGini; /* cria a variável */
2 divisaoCorrente  $\leftarrow$  4.9; /* Ex. recebe o 1º valor do atributo */
3 direita  $\leftarrow$  0;
4 esquerda  $\leftarrow$  6; /* Ex. recebe o total de dados existentes para o
   atributo */
5 while existirem dados do
6   if 1ª Dado Lista do Atributo MAIOR divisaoCorrente then
7      $\lfloor$  valorGini  $\leftarrow$  calculaGini(divisaoCorrente);
8   else
9      $\lfloor$  valorGini  $\leftarrow$  calculaGini(1ª Dado Lista);
10  if Primeiro Gini encontrado then
11     $\lfloor$  melhorGini  $\leftarrow$  valorGini;
12  else
13    if valorGini > melhorGini then
14       $\lfloor$  melhorGini  $\leftarrow$  valorGini
15  divisaoCorrente  $\leftarrow$  5.4; /* recebe o próximo dado do atributo */
16  direita recebe o que possui +1 e esquerda o -1;
17  (valorGini + divisaoCorrente)/2; /* encontrar ponto de divisão */

```

1.1.1.2 Algoritmo Naive Bayes

É um algoritmo considerado rápido, em relação a outros algoritmos de classificação, mesmo com grandes volumes de dados no seu conjunto de treinamentos. Utiliza modelo probabilístico, Teorema de Bayes, e possui característica de independência dos atributos, onde as classes não dependem 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.

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 dados. Ele agrupa objetos de uma certa classe em razão da probabilidade do objeto pertencer a esta classe.

$$P(c/x) = \frac{P(x/c)P(c)}{P(x)} \quad (1.3)$$

$$P(c/x) = P(x_1|c) * P(x_2|c) * \dots * P(x_n|c) * P(c) \quad (1.4)$$

- $P(c/x)$ 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). Mas mesmo atingido boa popularidade possui pontos negativos. A suposição de ter preditores independentes não acontece muito na vida real, pois acaba sendo difícil ter uma amostra de dados que sejam inteiramente independentes.

Outra situação é caso exista uma variável categórica que não foi observada na amostra tirada para o conjunto de treinamento, então poderá o modelo atribuir probabilidade 0(zero), não sendo capaz de fazer uma previsão. Quando isso acontecer uma técnica de alisamento é aplicada, chamada estimativa de Laplace, utilizadas em probabilidades condicionadas.

1.1.2 Aprendizado Não-Supervisionado

No Aprendizado Não-Supervisionado, não existe uma tentativa de se encontrar uma função que se aproxime da real. Logo porque os registros não são classificados, então 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 tenta agrupá-los de forma a ter algum sentido deles estarem juntos.

Quando o algoritmo encontram dados com mesma similaridade ele os agrupa formando clusters. Os números de clusters encontrados irão depender de como os algoritmos funcionam, junto com o grau de dissimilaridade entre elementos de grupos diferentes. Como não existe uma variável classe no Aprendizado Não-Supervisionado, então (BARBER,

2011) diz que o maior interesse seria em uma perspectiva probabilística de distribuição $p(x)$ de um determinado conjunto de dados.

$$D = \{x_n, n = 1, \dots, N\} \quad (1.5)$$

Uma vez que no conjunto, 1.5, não existe classe y , encontrado em um conjunto de treinamento, 1.1, o algoritmo precisa encontrar padrões nos atributos para fazer os agrupamentos.

1.2 Discretização

A discretização acontece em duas etapas no modelo defendido nesse trabalho, por isso a preocupação na explanação de seu funcionamento aqui nesta seção. O método de discretização faz a conversão de valores contínuos em valores discretos. A partir de um atributo com valores contínuos, a discretização irá forçar um ponto inicial e final definindo um intervalo e designando uma faixa para cada intervalo. Assim, ao invés de valores contínuos em cada atributo, será relacionado a faixa que aquele atributo pertence, definindo assim seu novo valor. O melhor método de discretização seria encontrar o conjunto de valores contínuos por faixa de intervalos pequenos (KOTSIANTIS; KANELLOPOULOS, 2006)

A partir de alguns autores (CATLETT, 2006; HWANG; LI, 2002) a discretização melhora a precisão e deixa um modelo mais rápido em seu conjunto de treinamento. Aqui nesse trabalho é utilizado a técnica de discretização antes da execução dos algoritmos e as faixas selecionadas são usadas para identificar o rótulo. Após o conhecimento do rótulo o valor da faixa é trocado pelo início e fim do intervalo.

Os métodos de discretização mais comumente utilizados no âmbito dos métodos não-supervisionados de acordo com (KOTSIANTIS; KANELLOPOULOS, 2006; DOUGHERTY; KOHAVI; SAHAMI, 1995) são os métodos de Discretização por Larguras Iguais(EWD) e Discretização por Frequências Iguais (EFD).

1.2.1 Discretização por Larguras Iguais - EWD

O método de Discretização por Larguras Iguais (EWD) faz a discretização de um intervalo, entre valores contínuos, dividindo em 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 2.

Para haver o ponto de corte antes tem que ser realizado a ordenação dos dados. A largura de cada faixa r_1, \dots, r_R na equação 1.6 é representada por w , que é calculada pela

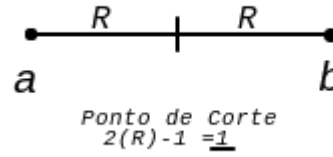


Figura 2 – Ponto de Corte (R-1)

diferença entre os limites superior e inferior do intervalo, dividido pela quantidade R de valores a serem gerados.

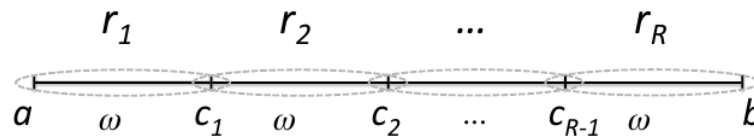
$$w = \frac{b - a}{R} \quad (1.6)$$

A variável w determina os pontos de corte (c_1, \dots, c_{R-1}) que irão delimitar o tamanho das faixas de valores. 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 .

O valor de cada faixa será representado por i , onde i é o índice indicando a faixa. De acordo com a figura 3 para dividir o intervalo $[a, b]$ em R faixas será necessário de $R - 1$ pontos de corte.

$$c_i = \begin{cases} a + w, & \text{se } i = 1 \\ c_{i-1} + w, & \text{caso contrário} \end{cases} \quad (1.7)$$

O valor da faixa do intervalo $[a, c_1]$ será o valor discreto igual ao índice de sua faixa r_1 . Então, um valor na faixa r_1 terá o valor representado por $1(um)$, pois $i = 1$ é o limite inferior mais largura da faixa, equação 1.7. E seguindo o mesmo raciocínio o valor da faixa $r_2 =]c_1, c_2]$ é representado por $2(dois)$, e consequentemente o valor que se encontra em uma faixa qualquer r_i será representado por i .

Figura 3 – Discretização EWD ³

1.2.2 Discretização por Frequência Iguais - EFD

Esse outro método de discretização já possui uma abordagem diferente a do EWD, pois a idéia é manter a quantidade de elementos distintos, entre os pontos de corte, com o

³ Figura extraída de (LOPES; MACHADO; RABELO,)

mesmo número. Dado um intervalo $[a, b]$ o número de faixas R e a quantidade de valores distintos ξ , onde $\xi \geq R$ o método EFD irá segmentar em R faixas de valores que possuem a mesma quantidade de elementos distintos λ . 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 λ . Para encontrar λ calcula-se o valor inteiro da divisão entre a quantidade de elementos distintos ξ pela quantidade de faixas de valores R , obtendo o número de elementos da faixa 1.8.

$$\lambda = \frac{\xi}{R} \quad (1.8)$$

Uma observação nesse método é a ocorrência em amostras que possuem uma má distribuição de valores de um dado atributo. Como um número significativo de repetições, causando um desequilíbrio nas distribuições dos elementos.

Uma vez no intervalo $[a, b]$ de elementos ordenado e calculado λ 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, 1.9.

$$c_i = v_{[i\lambda]} \quad (1.9)$$

Como na seção anterior do 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 4. 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, as faixas podem assumir faixas com tamanhos diferentes.

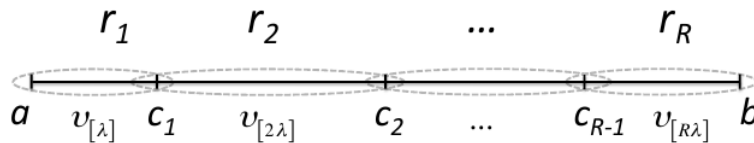


Figura 4 – Discretização EFD⁴

1.3 Trabalhos Correlatos

Esta seção propõe relacionar outros trabalhos servindo de complemento teórico, como também leitura imprescindível, para entender a variedade de aplicações referente ao assunto de rotulação de dados. Mas ao longo da escrita desta proposta de mestrado verificou-se uma carência de pesquisas no âmbito de rotulação de dados, referente ao tema

⁴ Figura extraída de (LOPES; MACHADO; RABELO,)

aqui proposto neste trabalho, tema este, que acabou redefinindo o termo de rotulação, não mais sendo como um classificador, e sim um significado um grupo formado.

O trabalho escrito por (LOPES; MACHADO; RABELO,) fez um estudo abordando o tema de rotulação de dados bastante significativo. Foi apresentado nesse trabalho o Problema de Rotulação, que representa também o problema proposto por esse trabalho, mas com abrangência e execução diferente do modelo apresentado na figura 5 . Nessa pesquisa foi utilizado como entrada um conjunto dados onde é feito um agrupamento automático, com algoritmos não-supervisionados formando os clusters. Logo após é utilizado algoritmos supervisionados nos grupos, e apresentado como saída um rótulo específico que melhor define o grupo formado. Esses rótulos são formados pela faixa de valor em conjunto com os atributos mais relevantes.

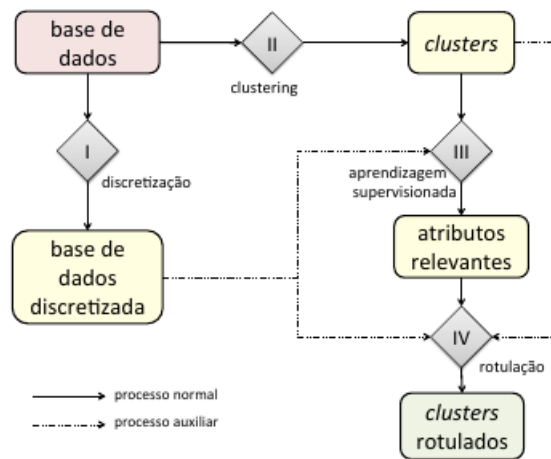


Figura 5 – Modelo (LOPES; MACHADO; RABELO,)

Em (LIMA, 2015) o problema em questão é fazer classificação e rotulação em uma base que possuem poucos elementos classificados. O método inicia com uma base dividida em elementos classificados(L) e não classificados(U). Após cada iteração o grupo L vai crescendo e automaticamente diminuindo o grupo U até que não tenha mais nenhum elemento em U, 6. Após isso é realizado uma etapa de agrupamento, sem levar em consideração os dados classificados anteriormente. Terminada essa etapa é feito uma validação para saber quais os rótulos foram considerados corretos.

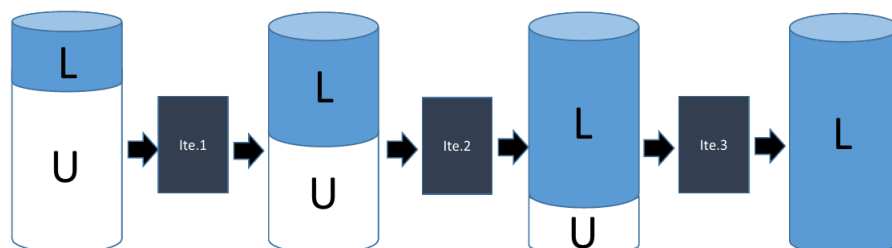


Figura 6 – Comportamento da base de dados a cada iteração. Método (FILHO, 2015)

Outra pesquisa aplicada em rotulação está em (FILHO, 2015) onde aborda o mesmo Problema de Rotulação. Mas a atuação é diferenciada, pois o modelo, figura 7 procura diferenças existentes em cada grupo através da seleção dos elementos que representam o grupo, e depois é construído a faixa de valores. Os grupos são formados pelo algoritmo Fuzzy C-Means e após isso que é selecionado os atributos.

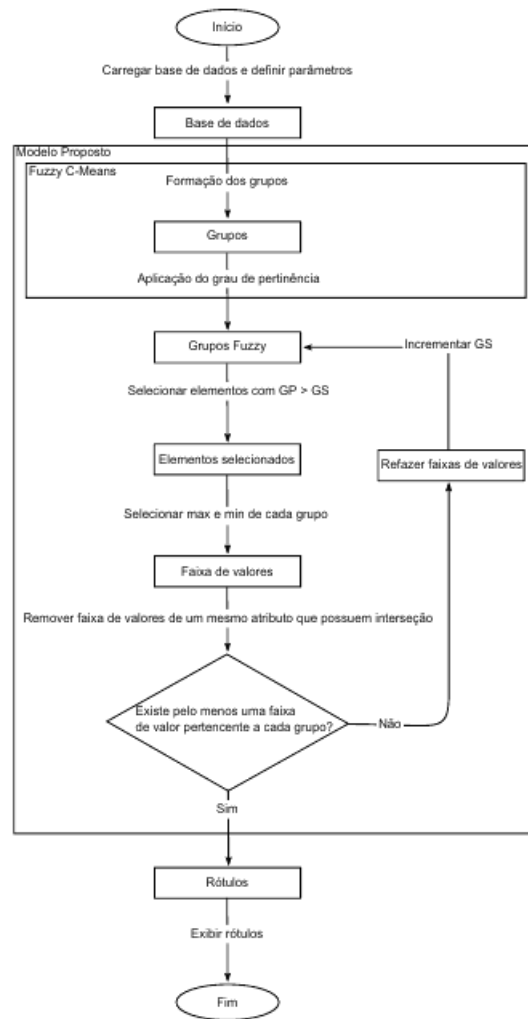


Figura 7 – Modelo (FILHO, 2015)

2 Metodologia / Materiais e Métodos

Esse capítulo abordará em uma sessão o problema proposto por esse trabalho, e logo em seguida, será apresentado um modelo de resolução. O objetivo ao final deste capítulo é poder resolver o problema, exibindo seus passos, e atribuir a qualquer outro pesquisador todo o conhecimento necessário para replicar este trabalho através das informações produzidas aqui.

2.1 Considerações do Problema

A abordagem do problema referente a essa proposta de mestrado segue uma linha já pesquisada por (LOPES; MACHADO; RABELO,), que seria o **Problema de Rotulação**. Esse conceito, rotulação de dados, já é estudado na literatura na área de aprendizagem não-supervisionada, sessão 1.1.2, onde é comum os algoritmos lidarem com os agrupamentos dos dados, e a criação de clusters a partir dos graus de similaridade entre os elementos.

Muitas pesquisas realizadas na área de rotulação fazem referencia, de fato, a classificação dos dados, e não da rotulação, nos termos desse trabalho. Ao agrupar um conjunto de elementos por um determinado critério, esta havendo uma classificação desses elementos escolhidos, mas pouco se sabe, qual é a compreensão desses grupos, já classificados.

Existe uma importância na criação dos clusters, contudo para o espectador é interessante existir um rótulo, desse grupo formado, oferecendo elementos em alguma tomada de decisão em razão de seu significado(rótulo).

Tem-se então o real problema de rotulação, contudo é necessário existir algum elemento definindo o porquê daquele grupo formado. O elemento é um rótulo composto por um, ou vários, atributo(s) de maior relevância no cluster, junto com uma faixa de valores. Essa faixa é um intervalo de valores definido pela discretização 1.2, onde o intervalo escolhido, seria a faixa que apresenta os valores que se repetem com a maior frequência.

O Problema de Rotulação é formalmente definido como segue abaixo:

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 $\vec{e}^{c_i} = (a_1, \dots, a_m)$ e ainda que com $c_i \cap c_{i'} = \{0\}$ com $1 \leq i, i' \leq K$ e $i \neq i'$.¹

¹ Adaptada de (LOPES; MACHADO; RABELO,)

- K é o número de clusters;
- c_i é o i -ésimo cluster qualquer;
- n^{c_i} é o número de elementos do cluster c_i ;
- $\vec{e}_{n^{(c_i)}}$ se refere ao j -ésimo elemento pertencente ao cluster c_i ;
- m é a dimensão do problema;

2.2 O Modelo de Resolução

Uma vez já conhecido a definição do problema - *Definição 1* - é possível situar a abrangência abordada aqui nessa pesquisa, pois a intenção do estudo científico desenvolvido aqui é provar a realização de **rotulação de dados com qualquer algoritmo supervisionado**, utilizando as técnicas abordadas neste texto.

O Modelo aqui proposto 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 2* formalizando a saída do modelo:

Definição 2 Dado um conjunto de rótulos $R = \{r_{c1}, \dots, r_{ck}\}$, no qual cada rótulo específico é dados por um conjunto de pares de valores, tem como saída um vetor com 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 .²

- 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_{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;
- m é a dimensão do problema;

Como apresentado na sessão 1.3, o autor foca em rotulação automática de grupos utilizando a estratégia de aprendizagem de máquina supervisionada, e paradigma conexionista, para provar seu trabalho. Mas aqui nessa pesquisa foi aplicado no modelo um acréscimo de 2(*dois*) algoritmos com paradigmas de aprendizado diferente do que já foi utilizado, compondo uma base para afirmar, que a partir dessas amostras pode-se fazer rotulação com qualquer algoritmo supervisionado.

² Adaptada de (LOPES; MACHADO; RABELO,)

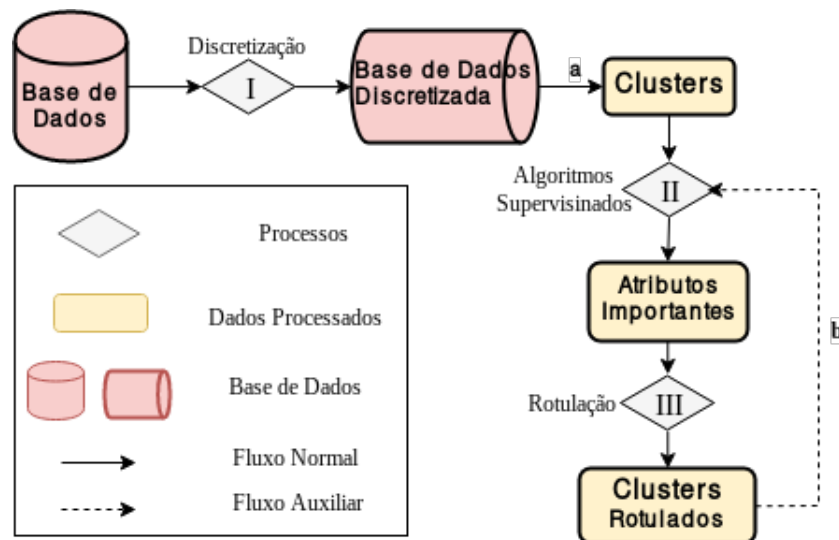


Figura 8 – Modelo de Resolução Proposto

O modelo, figura 8, inicialmente mostra a Base de Dados já classificada, pois o cerne desta pesquisa é conceder ao grupo um significado, a rotulação, através da técnica de correlação entre atributos, sessão 2.3. Essa base conterá valores contínuos, contudo, conforme modelo será necessário aplicar o método de discretização (I).

Uma vez com a base discretizada ocorre somente a separação dos clusters já classificados de acordo com a própria base de dados³. Isso é o funcionamento do fluxo (a), que nada mais é do que a separação da base em grupos que já classificados.

No passo (II) é onde serão executados os algoritmos de aprendizagem supervisionado, já visto nas sessões 1.1.1.1 e 1.1.1.2. Essa etapa é umas das mais importantes do método. O algoritmo supervisionado é aplicado várias vezes de acordo com o número de atributos do conjunto de dados, expresso no vetor de tamanho m . Onde o número de vezes será a quantidade de atributos da base de dados, formando uma tabela contendo valores de zero a cem, de acordo com a importância de cada atributo para o cluster.

Seguindo para o processo (III) acontecerá a escolha do(s) atributo(s) mais relevante(s), selecionado na tabela de atributos importantes, junto com o valor mais frequente desse atributo. Após essa etapa é criado um conjunto de rotulos para cada clusters. O fluxo (b) será utilizado enquanto houver outros algoritmos para serem executados.

2.3 Técnica de Correlação entre Atributos

Essa técnica ⁴ possui um grau de processamento diretamente proporcional a quantidade de características expressa na base de dados definido em R^m . Ela implica em

³ UCI - Machine Learning Repository. <http://archive.ics.uci.edu/ml/>

⁴ Extraída de (LOPES; MACHADO; RABELO,)

utilizar todos os atributos, menos o definido como classe, para fazer uma correlação entre eles junto ao algoritmo.

Pegando como exemplo uma base com os seguintes atributos: **atr1,atr2,atr3,classe**. Exclui o atributo classe, obtêm-se os 3(*três*) primeiros atributos, onde cada um deles será utilizado como classe em referência aos outros atributos.

Em um primeiro processamento de três, o primeiro atributo **atr1** se torna classe e executado com os outros dois atributos restantes com um algoritmo supervisionado. O resultado da correlação entre os atributos **atr2, atr3** em relação ao **atr1**(figura 9) é armazenado em uma matriz, e logo depois é realizado com **atr2** sendo classe e assim sucessivamente até o último atributo.



Figura 9 – Exemplo da técnica aplicada ao atr1 sendo classe

2.4 Exemplo com Base Modelo Fictícia

Para melhor esclarecer as etapas da figura 8, a tabela 1 contém uma base de dados que será utilizada para exemplificar todo o processo do modelo de resolução proposto nesta pesquisa. Logo na primeira coluna da tabela, retém um índice da linha da tabela responsável por identificar cada registro. Os outros campos são atributos que definem características do registro identificado pelo índice da primeira coluna.

Segundo a definição 1 um elemento é expresso por um vetor de dimensão m , com tamanho igual ao número de atributos. Um exemplo do elemento 2 da tabela 1, pode ser representado por $\vec{e}_2 = (1.26, 85.03, 20.45)$.

2.4.1 Processo (I) - Discretização

Segundo (CATLETT, 2006; HWANG; LI, 2002) o processo de discretização na etapa de treinamento pode aumentar a acurácia do algoritmo de aprendizado supervisionado. 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.

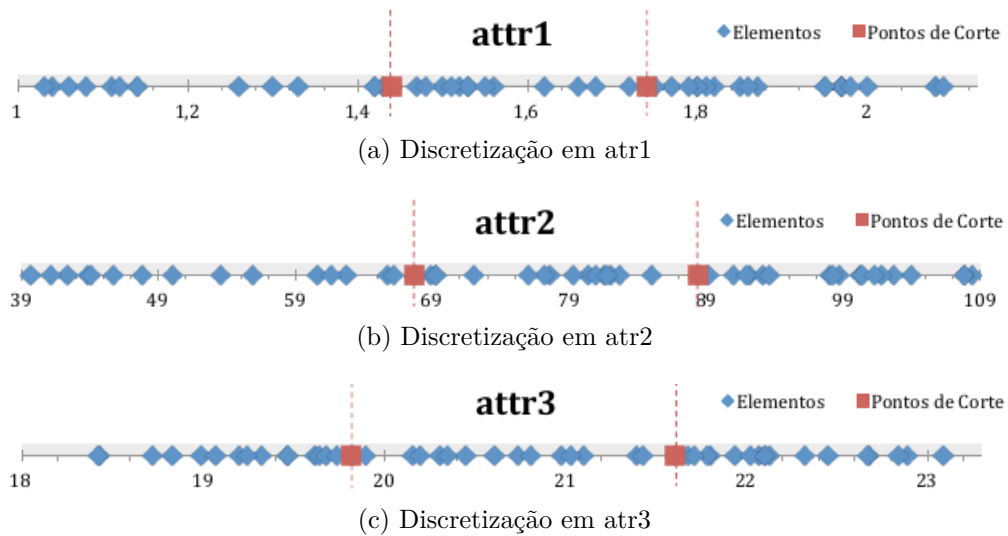
Para esse exemplo será utilizada a técnica de discretização por frequências iguais - EFD - e divisão de números de faixas, igual a $R=3$. Na figura⁵ 10 poderá ser visualizado

⁵ Figura adaptada de (LOPES; MACHADO; RABELO,)

Tabela 1 – Base de Dados Modelo

	atr1	atr2	atr3	classe		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

como é feita a discretização. Através da figura 10 fica claro o conteúdo da faixa 1, contendo

Figura 10 – Discretização de atributos utilizando EFD com $R = 3$

os valores iniciais até o primeiro ponto de corte, na faixa 2, são os valores após o primeiro

ponto de corte até o segundo ponto de corte. E na faixa 3 contém todos valores a partir do segundo ponto de corte.

Tabela 2 – 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

A tabela 2 é o resultado após a discretização de todos os atributos. Contudo sabe-se que ao se lidar com valores discretos onde cada intervalo representa uma faixa de valores poderá o algoritmo está perdendo um pouco de informação, mas por outro lado essa decisão tornará o aprendizado mais fácil de interpretar e com respostas mais rápidas.

2.4.2 Processo (II) - Algoritmos Supervisionados

Ao chegar nessa etapa, Processo (II) da figura 8, já se tem uma base discretizada e clusters formados, tabela 2. Agora é feita a execução do algoritmo de aprendizado supervisionado e identificado os atributos de maior importância de cada cluster.

Uma vez com o conhecimento do cluster, serão percorridos todos os atributos, 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, e verificar

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. Como visualizado na figura 11

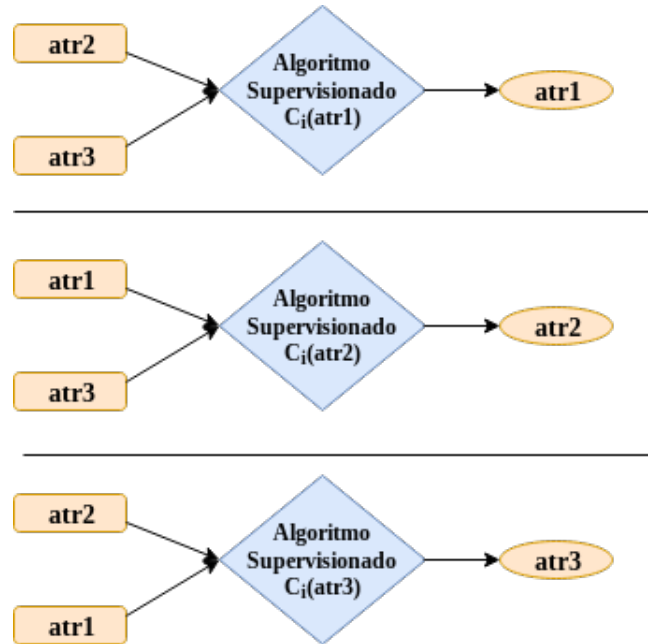


Figura 11 – Exemplo da técnica aplicada aos 3(três) atributos, cada um sendo classe em determinada iteração

Essa correlação entre os atributos junto com a aplicação dos algoritmos geram uma matriz de atributos importantes. O quão relevante o atributo será em relação ao cluster $c_i(\text{atr})$, será dado em uma porcentagem de acerto quando aplicado como saída na execução de um algoritmo supervisionado. Quanto maior sua porcentagem, mais correlacionado é o atributo em relação ao demais(figura 12), logo ele é considerado um atributo bem relevante. Sendo assim esse atributo poderá resumir as características do problema, podendo ser considerado um atributo importante e escolhido como rótulo.

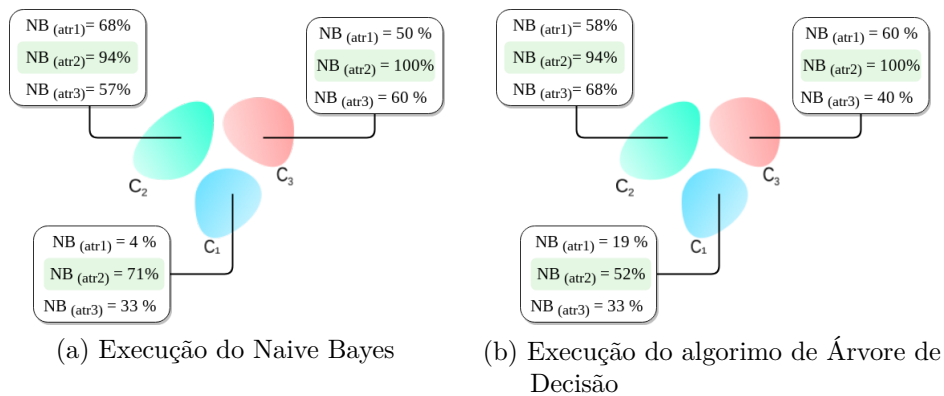


Figura 12 – Resultado dos Algoritmos

Na figura 12a mostra o resultado da execução do Naive Bayes em cima da Base Modelo e exibe os resultados em porcentagem de acerto de cada atributo em relação aos

demaís. O mesmo acontece com a figura 12b onde é aplicado um algoritmo de Árvore de Decisão, 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 uma variável V . Com essa variável a seleção dos atributos rótulos de um clusters, serão todos os atributos que tiverem até uma diferença V em relação ao atributo de maior taxa de acerto, expresso em porcentagem. Portanto 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% serão 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 cima de um conjunto de dados. Nese exemplo os atributos importantes com $V = 12$ utilizando a figura 12a por cluster, teriam os rótulos $r_{c_i} : r_{c_1} = \{atr2\}, r_{c_2} = \{atr2\}, r_{c_3} = \{atr2\}$.

2.4.3 Processo (III) - Rotulação

Nesse processo de rotulação serão calculados os intervalos dos atributos que estão na figura 8 como Atributos Importantes, selecionados na etapa anterior. Para compor o rótulo r_{c_i} do cluster c_i é calculado a faixa do atributo que tiver maior frequência. É possível verificar neste exemplo, da Base Modelo, o resultado da figura 12a, onde o rótulo r_{c_1} é o **atr2**=]67.66, 88.36], porque o valor da faixa de maior frequência do cluster c_1 em relação ao atributo **atr2** é a faixa 2 (figura 10c), que é representa o limite inferior]67.66s e o limite superior, 88.36].

Uma vez terminado o processo (III) de rotulação, o fluxo b do modelo da figura 8, só é seguido caso seja necessário para executar outro algoritmo.

Contudo pode-se definir os rótulos nesta etapa da seguinte maneira:

- Algoritmo Naive Bayes 12a aplicado na BD Modelo

$$r_{c_1} = (atr2,]67.66, 88.36]);$$

$$r_{c_2} = (atr2,]88.36, 108.36]) ;$$

$$r_{c_3} = (atr2, [39.67, 67.66]);$$

- Algoritmo de Árvore de Decisão 12b aplicado na BD Modelo

$$r_{c_1} = (atr2,]67.66, 88.36]);$$

$$r_{c_2} = (atr2,]88.36, 108.36]) ;$$

$$r_{c_3} = (atr2, [39.67, 67.66]);$$

Logo abaixo o algoritmo 2 exibe a rotina em forma de pseudocódigo para melhor entendimento.

Algorithm 2: 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     prepara_vetor_atributos/classe;  
8     Aplica_algoritmo_supervisionado;  
9     Calcula_matriz_de_porcentagem_de_acertos;  
10    Carrega_atributos_importantes_considerando_V;  
11    Associa_valores_aos_intervalos;  
12 Exibe_rótulos_todos_clusters;
```

3 Resultados

Os resultados obtidos aqui neste capítulo foram referentes a aplicação do método de rotulação em 3(*três*) bases de dados distintas. Um dos primeiros passos na análise de aprendizagem de máquina é quando o analista prepara os dados para poder utilizar um método de aprendizagem apropriado.

Então a escolha da base de dados também tem influência direta em bons resultados. E sabendo disso a escolha dos conjuntos de dados utilizados nesta pesquisa foi por conta delas apresentarem características diferentes, e também por serem conhecidas, facilitando a análise e servindo de amostra a outras base.

3.1 Implementação

Para conseguir gerar os resultados aqui escritos foram feitas implementações utilizando a ferramenta MATLAB ¹, onde junto a ela é possível utilizar suas funções de aprendizado de máquina já prontas. MATLAB possui uma linguagem técnica, e de fácil implementação por já possuir uma gama de funções² preparadas para aprendizado de máquina. Por esses motivos essa ferramenta foi escolhida para colocar em prática essa pesquisa.

Foram realizados vários testes com o intuito de tentar otimizar resultados e poder compará-los a outras pesquisas já escritas. Seguindo essa linha foi determinado a escolha de 3(*três*) bases de dados já conhecidas, onde na implementação de cada uma delas surgiu algumas alterações, dependendo da base, na variável(V), quantidade de faixas(R) e método de discretização(EWD,EFD). Essas mudanças para cada base servirão para otimizar os resultados.

Cada base de dados será aplicado dois algoritmos de aprendizado supervisionado que possuem paradigmas diferentes para servir de amostra e poder assim tirar conclusões sobre a rotulação em quaisquer algoritmos supervisionados.

Os algoritmos utilizados foram o Naive Bayes, sessão 1.1.1.2, com paradigma estatístico. E também o algoritmo Classification e Regression Trees - CART, 1.1.1.1, com paradigma simbólico de árvore de decisão.

¹ <http://www.mathworks.com/products/matlab/>

² versão: R2016a(9.0.0.341360); 64-bit (glnxa64)

3.2 Seeds - Identificação de Tipos de Semente

Essa base foi extraída da UCI Machine Learning³, composta por 7(*sete*) atributos definindo suas características e mais uma definindo sua classificação, sendo este último, um atributo classe responsável por identificar o tipo de semente. Possuindo um total de 210 registros classificados em 3(*três*) categorias:

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

Na configuração de implementação foi utilizado o método EFD de discretização com divisão em três faixas, $R = 3$ para todos os atributos, e inserido o valor de variação $V = 3\%$.

Na tabela 3 e tabela 6 são apresentados os resultados com a execução do algoritmo Naive Bayes e CART respectivamente. Elas são formadas por uma coluna informando os **Clusters**, **Rótulos** compostos pelo **Atributo** e sua **Faixa** de valor. Junto também a coluna **Relevância** exibindo a resposta do algoritmo em porcentagem, da correlação do atributo em relação aos outros atributos do cluster, retirado da tabela 4 e da tabela 7 respectivamente. E por último a coluna **Elem Fora da Faixa** que mostra a quantidade de elementos que não estão dentro da faixa do rótulo. Essa última coluna tem a função de exibir em números a quantidade de valores que não estão participando da porcentagem da coluna de **Relevância**. Para quem está analisando a tabela é interessante mais esse dado, pois pode compara com o total de elementos do grupo.

3.2.1 Naive Bayes

Tabela 3 – Resultado da aplicação do algoritmo Naive Bayes

Cluster	Rótulos		Relevância(%)	Elem fora da Faixa
	Atributos	Faixa		
1	area] 12.78 ~ 16.14]	92%	14
2	area] 16.14 ~ 21.18]	95%	6
	lkernel] 5.826 ~ 6.675]	92%	6
3	perimetro	[12.41 ~ 13.73]	95%	5

Analisando a coluna rótulo da tabela 3, nota-se que o atributo **area** aparece tanto no cluster 1 como também no cluster 2. A técnica envolve não só o rótulo como também a faixa que os valores mais se repetem dentro do atributo. Nesse caso pode-se observar que o atributo se repete entre os clusters. Mas no cluster 1, a faixa de valores difere do cluster 2, sem comentar que no cluster 2 existe outro atributo compondo o rótulo, **lkernel**.

³ <http://archive.ics.uci.edu/ml/>

A seleção dos atributos rótulos acontece da diferença da variável $V = 3\%$ em relação ao atributo de maior relevância. Caso essa variável tenha o valor alterado, os rótulos dos clusters poderão sofrer mudanças, pois poderá aumentar ou diminuir o número de atributos dos rótulos, dependendo do valor inserido em V . Através da tabela 4 é possível analisar todos os valores de relevância gerados para os atributos e analisar qual valor pode-se inserir em V para montar o rótulo.

Tabela 4 – 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

A tabela 4 é formada por clusters representado pelas linhas, e colunas representado por atributos. Essa tabela é fruto da implementação do Naive Bayes em cima dessa base de dados, e foi gerada para auxiliar a retirada dos atributos rótulos. Uma análise pode ser feita através desses dados e ajudar a definir um valor para a variável V . Percebe-se que algumas características são mais bem correlacionadas que outras, através de seus valores mais altos. Isso indica o grau de relacionamento entre os atributos após a aplicação do algoritmo.

Tabela 5 – 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

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

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

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

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

Para conseguir ter uma idéia mais ampla dessas informações, na tabela 5 é exposto o resultado de 4(*quatro*) execuções do Algoritmo Naive Bayes, e pode-se constatar que mesmo havendo algumas alterações em seus valores nos atributos em cada execução, a correlação entre os atributos não oferece muita alteração. Como exemplo, o atributo **area**, possui o melhor grau de relacionamento em seu grupo, mesmo testado em quatro execuções, como mostrado na tabela 5.

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

- $r_{c_1} = \{(area,]12.78 \sim 16.14])\}$
- $r_{c_2} = \{(area,]16.14 \sim 21.18]), (Lkernel,]5.826 \sim 6.675])\}$
- $r_{c_3} = \{(perimetro, [12.41 \sim 13.73])\}$

3.2.2 CART

Já na tabela 6, tem-se o resultado da aplicação do algoritmo supervisionado CART. Ele é implementado para solucionar casos de árvore de decisão pelo MATLAB. O intuito é testar a base de dados no paradigma simbólico.

Tabela 6 – Resultado da aplicação do algoritmo CART

Cluster	Rótulos		Relevância(%)	Elem fora da Faixa
	Atributos	Faixa		
1	area] 12.78 ~ 16.14]	91%	14
	perimetro	[13.73 ~ 15.18]	94%	14
2	area] 16.14 ~ 21.18]	95%	6
	perimetro] 15.18 ~ 17.25]	98%	7
3	perimetro	[12.41 ~ 13.73]	95%	5
	wkernel	[2.63 ~ 3.049]	97%	9

Pode-se verificar na tabela 6 que os clusters 1 e 2 possuem o mesmo conjunto de atributos selecionados no campo de rótulo. Mas isso não implica dizer que os dois grupos são identificados pelo mesmo rótulo. O rótulo é composto pelos atributos e pelas faixas, onde a faixa é escolhida é a faixa onde se tem o maior número de valores que se repetem nessa faixa. Então, caso exista um vetor de elementos já discretizados, $\vec{e}_{(c_i)} = \{1, 1, 1, 2, 2, 2, 2, 3, 3\}$. Neste vetor o valor que mais se repete é o 2, então a faixa 2 foi a que mais se repetiu e com isso é a escolhida para compor o rótulo com o atributo mais relevante.

Para entender a escolha desses atributos no campo de rótulos, a tabela 7 exhibe o resultado gerado na execução do algoritmo em cima da base. Cada valor desses é o resultado da aplicação do algoritmo enquanto o atributo era a classe da vez conforme figura 11, sessão 2.4.2. O atributo de maior valor junto com os atributos da diferença de V com o mais relevante, são escolhidos para ser rótulos. Na linha(cluster) 1 o maior valor é o atributo perimetro. Pega o valor encontrado em perimetro, e subtrai de $V = 3$. A partir daí o(s) atributo(s) que possui(rem) um valor que está entre este resultado até o mais alto, irá compor o rótulo.

Foram realizadas vários teste, onde alguns deles estão na tabela 8. Essas operações foram execuções do algoritmo CART em cima da base, para provar que a técnica de correlação de atributos, 2.3, é funcional para este algoritmo. O mesmo pode ser visto no

Tabela 7 – Resultado da Correlação dos atributos pelo CART; 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	91.4	94.2	58.5	80.0	81.4	61.4	61.4
	2	98.5	98.5	51.4	90.0	88.5	42.8	88.5
	3	92.7	95.7	80.0	88.5	97.1	58.5	78.5

algoritmo de paradigma estatístico, sessão 3.2.1, realizado nessa pesquisa. O comportamento de ambos foram bem semelhantes, pois eles seguem o padrão de valores os quais não se alteram muito a cada iteração.

Tabela 8 – Resultado de 4(*quatro*) iterações do algoritmo CART; Legenda dos Atributos: (A)area, (B)perimetro, (C)compactness, (D)Lkernel, (E)Wkernel, (F)asymetry, (G)lkgroove

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

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

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

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

O resultado do algoritmo CART na base de dados **Seeds** tem como rótulos:

- $r_{c_1} = \{(area, [12.78 \sim 16.14]), (perimetro, [13.73 \sim 15.18])\}$
- $r_{c_2} = \{(area, [16.14 \sim 21.18]), (perimetro, [15.18 \sim 17.25])\}$
- $r_{c_3} = \{(perimetro, [12.41 \sim 13.73]), (wkernet, [2.63 \sim 3.049])\}$

3.3 Iris - Identificação de Tipos de Plantas

Essa base de dados utilizada neste trabalho, extraída do UCI⁴ Machine Learning, também já é uma base conhecida em outras pesquisas⁵. Possui 150 registros de amostra de plantas com um total de 4 atributos definindo as características das plantas e mais 1 atributo classe. Este último atributo classifica o tipo de planta em 3 tipos, segundo (RUNKLER, 2016) :

⁴ <https://archive.ics.uci.edu/ml/>

⁵ (LOPES; MACHADO; RABELO,), (KOTSIANTIS; PINTELAS, 2005), (FILHO, 2015) e outros

- 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.

Foi aplicado na configuração de implementação o método EFD⁶ de discretização com divisão em três faixas, $R = 3$ para todos os atributos, e inserido o valor de variação $V = 3\%$. Mais uma vez, o valor V é subjetivo do pesquisador e influenciado pelos valores de correlação dos atributos nos grupos, tabela 10.

Seguindo a análise, semelhante da base de dados anterior, será realizado testes utilizando dois algoritmos⁷ e cada resultado será exibido em tabelas. Portando as colunas são formadas por **Clusters**, **Rótulos**, **Relevância** e **Elem fora da Faixa** representando os valores que não estão dentro da faixa escolhida como rótulo. Também foi posto nas tabelas 12 e 10 os resultados das correlações entre os atributos de cada grupo, servindo de informação para decisão do valor de V . E também apresentado os resultados de outras iterações de cada algoritmo, para mostrar o comportamento dos atributos entre eles no grupo.

3.3.1 Naive Bayes

Segue a tabela 9 com os resultados da rotulação 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 obteve os mesmo atributos do cluster 1, mas as faixas de valores são diferentes. E no cluster 3 somente um atributo foi selecionado, petalwidth, conseguindo ter uma relevância entre os outros atributos de 90%.

Tabela 9 – Resultado da aplicação do algoritmo Naive Bayes

Cluster	Rótulos		Relevância(%)	Elem fora da Faixa
	Atributos	Faixa		
1	petallength	[1.0 ~ 3.7]	100%	0
	petalwidth	[0.1 ~ 1.0]	100%	0
2	petallength] 3.7 ~ 5.1]	84%	7
	petalwidth] 1.0 ~ 1.7]	82%	8
3	petalwidth] 1.7 ~ 2.5]	90%	5

Os valores na coluna de relevância não podem ser analisados isoladamente. Para isso a tabela 10 possui os valores de todos os atributos no momento que ele são classes.

⁶ sessão 1.2.2

⁷ sessões 3.2.1, 3.2.2

Os valores são em porcentagem para melhor análise do grau de relacionamento entre os outros atributos.

Tabela 10 – Resultado de 4(*quatro*) execuções do algoritmo Naive Bayes; Legenda dos Atributos: (SL)sepalength,(SW)sepalwidth,(PL)petallength,(PW)petalwidth

1a. Execução		Atributos			
		SL	SW	PL	PW
Clusters	1	80	68	100	100
	2	72	76	84	82
	3	76	74	68	90

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

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

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

Nessa tabela 10 foram inseridas quatro tabelas com os resultados de cada execução. Foi escolhida na tabela 10 a 1a. execução para montar a tabela de rótulos, tabela 9. A partir dela o pesquisador poderá arbitrar sobre o valor de V para melhor adaptá-lo a base. Das várias execuções expostas na tabela 10, percebe-se que não há muita diferença entre esses valores em cada execução. Isso mostra um padrão de valores de acordo com a base. No caso da 1a. execução os valores escolhidos como rótulo estão destacados em cada cluster.

Se a tabela escolhida fosse a da 4a. execução, os valores de rótulos seriam modificados, em virtude da diferença do atributo de maior valor com a variável V . O atributo de maior valor, no cluster 2, é o PL com 86%. Então a diferença desse valor com $V = 3$ chega em 83%. Tanto SL, SW e PW possuem valores inferiores a 83%, com isso o atributo escolhido para compor o rótulo no cluster 2 seria só o PL, diferente das outras tabelas que são o PL e PW.

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

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

3.3.2 CART

A aplicação do algoritmo CART na base de dados Iris gerou a tabela 11 como resultado, e ao examinar pode-se observar uma semelhante com a sessão anterior 3.3.1 onde foi aplicado o Naive Bayes.

Tabela 11 – Resultado da aplicação do algoritmo CART

Cluster	Rótulos		Relevância(%)	Elem fora da Faixa
	Atributos	Faixa		
1	petallength	[1.0 ~ 3.7]	100%	0
	petalwidth	[0.1 ~ 1.0]	100%	0
2	petallength] 3.7 ~ 5.1]	88%	7
	petalwidth] 1.0 ~ 1.7]	90%	8
3	petalwidth] 1.7 ~ 2.5]	90%	5

Ao observar a tabela, 11, percebe-se que o resultado de rotulação é idêntico ao do algoritmo anterior, mas na coluna de **Relevância** existe uma diferença no cluster 2, contudo essa diferença não chega a modificar a rotulação desta base de dados. Iris Fazendo uma análise dessa diferença, na coluna de **Relevância**, valor esse, adquirido conforme a 1a. Execução da tabela 12, e já sabendo que o a escolha do rótulo tem a influência do valor de relevância, e também do valor de V . É constatado que os valores podem até ser maiores, nessa caso aqui específico, e mesmo assim a rotulação seria a mesma. Então mesmo alterando o valor de V até 12 , os rótulos continuariam os mesmo.

Com o valor de $V = 3$ ou até com $V = 12$, como exemplo, e analisando as execuções da tabela 12, pode-se perceber que a diferença dos números dos rótulos são altos em relação aos outros atributos. Então a influência dos atributos escolhidos são bem fortes em relação aos que não são atributos. Com estes resultados percebe-se que os atributos são bem correlacionado com as classes, gerando uma boa distinção dos rótulos.

Tabela 12 – Resultado de 4(*quatro*) iterações do algoritmo CART; Legenda dos Atributos: (SL)sepalength,(SW)sepalwidth,(PL)petallength,(PW)petalwidth

1a. Execução		Atributos			
Clusters		SL	SW	PL	PW
	1	80	68	100	100
	2	74	76	88	90
	3	68	68	74	90

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

3a. Execução		Atributos			
Clusters		SL	SW	PL	PW
	1	80	68	100	100
	2	74	76	86	90
	3	70	66	78	90

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

Segue abaixo os rótulos na base de dados **Iris** aplicado no algoritmo CART:

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

Conclusões

Aqui serão abordados as conclusões dessa proposta de mestrado referentes aos resultados do capítulo 3, bem como uma sessão de Trabalhos Futuros e Cronograma. Em Trabalhos Futuros a pretensão é expor melhorias e expansões de tudo que fora realizado nesta pesquisa, e mostrar que existe uma continuidade para todo esse estudo aqui elaborado. Já no Cronograma, será criada uma tabela dividida em meses definindo os passos a serem seguidos até a conclusão da dissertação.

Conclusões

O que se pretende fazer aqui nesta sessão é comentar as conclusões aqui realizadas neste trabalho, capítulo 3, onde nesse capítulo mostrou a aplicação dos algoritmos supervisionados em cima de algumas bases de dados, e de fato relatar se o objetivo foi satisfatório, ou não.

Uma vez conhecido o problema, foi realizado a execução de dois algoritmos supervisionados, servindo de amostra para provar que era possível fazer rotulação de dados com quaisquer algoritmos supervisionados, tema deste trabalho. E já identificando alguns trabalhos que já haviam feito rotulação, (LOPES; MACHADO; RABELO,), utilizando algoritmos supervisionados, o intuito era executar outros algoritmos com paradigmas diferentes aos que foram realizados nessas pesquisas anteriores.

Dos dois algoritmos apresentados, um pertencente ao paradigma simbólico e o outro estatístico. Ademais cada execução desses algoritmos resultarão respostas satisfatórias no âmbito da rotulação de dados. Embora cada um tenham suas peculiaridades.

No modelo de resolução proposto foi inicialmente utilizado Naive Bayes na base de dados Seeds⁸. Logo o resultado mostrou-se bem confiável pois o método consegue mostrar através da tabela 5, os valores de correlação entre os atributos, também exibido na coluna **Relevância** da tabela 3. Dessa forma fica fácil identificar quais os atributos podem ser os rótulos dos clusters. Embora essa decisão possa ser modificada de acordo com o valor da variável V . Variável essa criada para melhor escolher os atributos do rótulo, e podendo assumir valores diferentes dependendo do comportamento da base de dados.

Continuando com a base Seeds, 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 discretização⁹ e independente da primeira etapa. O método é

⁸ Sessão 3.2.1

⁹ sessão 1.2

capaz de gerar a faixa de maior repetição de valores de qualquer atributo, mas aqui neste trabalho escolhemos o 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 3 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.

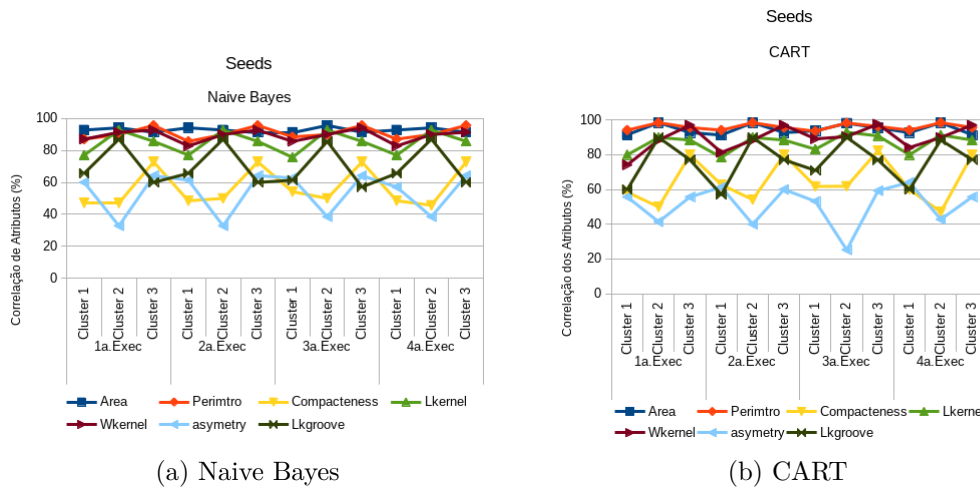


Figura 13 – Gráfico de Execuções dos algoritmos supervisionados na base de dados SEEDS.

No caso 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 5 e 8 podem ser observadas nos gráficos da figura 13. Como já comentado anteriormente o comportamento dos valores do correlacionamento dos atributos ao longo das execuções mostra-se equilibrada, figura 13b. O gráfico do CART tem um movimento semelhante ao do aplicado do Naive Bayes 13a, embora a variável **asymetry** saia um pouco do padrão, mas como seus valores são baixos, nada alterou nos rótulos, 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 acima pode-se dizer nesta análise, que o Naive Bayes acabou tendo resultados um pouco melhores, 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, comparando com os resultados do Naive Bayes. Quer dizer, o rótulo deixa de representar mais elementos usando o CART ao invés do Naive Bayes, ou em outras palavras, o Naive Bayes representou mais elementos que o CART.

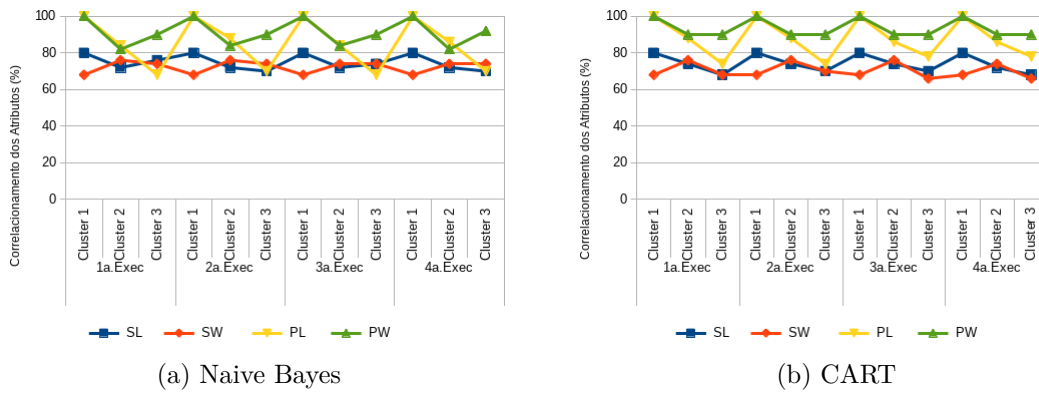


Figura 14 – Gráfico de Execuções dos algoritmos supervisionados na base de dados IRIS.

Ja na base de dados Iris, os dois algoritmos supervisionados testados apresentaram os mesmos rótulos. Mantendo as mesmas configurações, $R = 3$, $V = 3\%$ e EFD na discretização. Nos gráficos da figura 14 pode-se acompanhar como os valores dos atributos se comportam em seus clusters nas 4 iterações.

Nesta base Iris os algoritmos tem resultados nos gráficos, bastantes semelhantes, e logo se percebe que essa base contém características que possuem mais coerência com a classe em relação ao da base SEEDS, pois nenhum atributo possui valor abaixo da linha 65(%) de relacionamento entre eles. Embora no gráfico as linhas referentes aos comportamentos dos atributos não sejam totalmente iguais em cada algoritmo executado, não chegou a um valor que diferenciase para modificar o resultado dos rótulos como resposta.

Na rotulação encontrada pelo dois algoritmos nos resultados da tabela 9 e 11 o rótulo escolhido no cluster 1 teve dois atributos, **petallength** e **petalwidth** e cada um deles definiram uma faixa onde foi possível abranger 100% de elementos dentro das faixas escolhidas por cada um dos atributos. Já no cluster 2 os mesmos atributos são escolhidos mas com faixas de valores diferentes. Embora não tivesse fechado os mesmos valores do cluster 1, obteve um total, de 7 elementos do **petallength** e 8 do **petalwidth**, igual a 15 elementos que não estão dentro da faixa delimitada pelos rótulos. O cluster 2 possui um total de 50 elementos e 15 deles não são representados pelo rótulo do cluster. E no cluster 3 o atributo escolhido para compor o rótulo é mais uma vez o **petalwidth**. Logo se percebe a importância do atributo nos clusters, mas em nenhum deles a faixa é a mesma. Isso define bem o rótulo do cluster 3, pois o rótulo representa 45 elementos dentro do cluster, possuindo somente 5 elementos fora dessa faixa representada pelo atributo.

A repetição do atributo **petalwidth** em todos os rótulos, acaba mostrando o grau de relevância desse atributo na base de dados. Para um especialista é interessante saber que esse atributo possui um grande referencial na base de dados. Nos rótulos esse atributo assumiu faixas diferentes conseguindo assim o método da um significado ao cluster.

Com uma breve análise já se constata que os resultados nos dois algoritmos supervisionados foram bem satisfatórios nas bases utilizadas, e provando que é possível a rotulação de dados. Além de conseguir representar bem os clusters através dos rótulos encontrados. E uma observação técnica dos algoritmos utilizados é que o CART se mostrou bem mais rápido em relação ao Naive Bayes para gerar os resultados.

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 e com isso traçar uma estratégia caso seja necessário a utilização da idéia de rotulação por algum fim. Caso algum órgão/setor/empresa precise utilizar a rotulação em seu meio, seria interessante o analista de dados, saber com quais algoritmos supervisionados ele obteria melhores resultados. Embora se saiba nesse estudo que quaisquer algoritmos supervisionados são capazes de realizar a rotulação de dados, também foi provado que em algumas bases um algoritmo se sobressai a outro. Por consequencia disso ter um maior número de base com características diferentes ajudaria em uma tomada de decisão.

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

Cronograma

Tabela 13 – Cronograma de atividades

Atividades	Meses					
	Março	Abril	Maió	Junho	Julho	Agosto
Testes com Novas Bases de Dados						
Modificar Números de Faixa (R)						
Testar com outros Métodos de Discretização						
Testar com outros Algoritmos com Paradigmas Diferentes						
Preparar Artigo						
Escrita da Dissertação						

Referências

- BARBER, D. *Bayesian Reasoning and Machine Learning*. [s.n.], 2011. ISSN 9780521518147. ISBN 9780511804779. Disponível em: <<http://ebooks.cambridge.org/ref/id/CBO9780511804779>>. Citado 3 vezes nas páginas 1, 6 e 10.
- BREIMAN, L. *Classification and Regression Trees*. [S.l.: s.n.], 1984. (Chapman e Hall/CRC). Citado na página 7.
- CATLETT, J. Into Ordered Discrete Attributes. v. 3, n. 1989, p. 2006, 2006. Citado 2 vezes nas páginas 10 e 18.
- DOUGHERTY, J.; KOHAVI, R.; SAHAMI, M. Supervised and Unsupervised Discretization of Continuous Features. *Machine Learning Proceedings 1995*, v. 0, p. 194–202, 1995. ISSN 0717-6163. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/B9781558603776500323>>. Citado na página 10.
- FILHO, V. P. R. *Rotulacao de grupos utilizando conjuntos fuzzy*. Tese (Doutorado) — Universidade Federal do Piauí, 2015. Citado 4 vezes nas páginas 15, 13, 14 e 29.
- 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 10 e 18.
- 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 10.
- KOTSIANTIS, S. B.; PINTELAS, P. E. Logitboost of simple bayesian classifier. *Informatica (Slovenia)*, v. 29, p. 53, 2005. Citado na página 29.
- LIMA, B. V. A. Método Semissupervisionado de Rotulação e Classificação Utilizando Agrupamento por Sementes e Classificadores. 2015. Citado na página 13.
- LOPES, L. A.; MACHADO, V. P.; RABELO, R. D. A. L. Automatic Labeling of Groupings through Supervised Machine Learning. Citado 10 vezes nas páginas 15, 2, 11, 12, 13, 16, 17, 18, 29 e 33.
- LUCCA, G. et al. Uma implementação do algoritmo Naïve Bayes para classificação de texto. *Centro de Ciências Computacionais - Universidade Federal do Rio Grande (FURG) Rio Grande - RS - Brasil*, p. 1–4, 2013. 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 1 e 5.
- QUINLAN, J. R. Induction of Decision Trees. *Machine Learning*, v. 1, n. 1, p. 81–106, 1986. ISSN 15730565. Citado na página 7.
- RAIMUNDO, L. R.; MATTOS, M. C. D.; WALESKA, P. O Algoritmo de Classificação CART em uma Ferramenta de Data Mining. 2008. Citado na página 8.
- RUNKLER, T. A. *Models and Algorithms for Intelligent Data Analysis*. 2. ed. [S.l.]: Springer Vieweg, 2016. (Data Analytics). ISBN 978-3-658-14075-5. Citado na página 29.
- RUSSEL, S.; NORVIG, P. *Inteligência Artificial*. 3^a. ed. Rio de Janeiro: [s.n.], 2013. ISBN 9780136042594. Citado 2 vezes nas páginas 5 e 6.
- 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.
- 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 8.