Universidade Federal de Itajubá

Instituto de Matemática e Computação

Aprendizado de máquina fuzzy
potenciais contribuições da teoria fuzzy na concepção de técnicas de aprendizado de máquina para a tarefa de classificação

RELATÓRIO FINAL
PROGRAMA (PIVIC)

CICLO 2019/2020

Aluno: Isabela Corsi
Matrícula: 2018016354
Curso: Ciência da Computação
Orientador: Isabela Neves Drummond

Fase/Período:2019/2020

RESUMO

As árvores de decisão (AD) são empregadas na solução de problemas que envolvem a tarefa de classificação, uma tarefa supervisionada de aprendizado de máquina. Uma de suas características principais é a utilização do método de dividir para conquistar, ou seja, transforma problemas complexos em problemas mais simples, recebendo como entrada um conjunto composto por atributos de entrada e saída. Esta saída é a classificação das instâncias que compõem o conjunto. Vários algoritmos baseados na AD são apresentados na literatura, sendo os mais conhecidos: ID3, CART e o C4.5. Entretanto os modelos clássicos de árvore de decisão não abrangem o tratamento de problemas que envolvem imprecisão. Para tanto foram desenvolvidas as chamadas Árvore de Decisão Fuzzy (ADF), que atribui a capacidade de suportar variáveis fuzzy (ou difusas, em português), ou seja, as variáveis que são tratadas com incerteza. Diante deste contexto, este trabalho de iniciação científica possui como foco principal analisar e comparar o modelo clássico de AD C4.5 e o ADF. Foram realizados testes com 6 bases de dados disponíveis no repositório de dados UCI: Pima Indians Diabetes, Glass, Heart Disease, Iris, Wine e Haberman's Survival, e a comparação foi realizada através da análise da matriz de confusão gerada a partir da classificação, e a taxa de acerto e erro, em função do total de instâncias classificadas. Para gerar a AD (baseada no modelo C4.5), foi utilizado o WEKA, uma plataforma de mineração de dados implementada em linguagem Java, sendo possível gerar uma AD utilizando-se o algoritmo J48 (implementação baseada no modelo C4.5). O modelo tem como entrada um conjunto de dados e como saída as medidas selecionadas para comparação. Para gerar uma ADF, primeiramente, foi utilizado o algoritmo FuzzyDT proposto por Marco E. Cintra, que recebe como entrada um conjunto de dados, no formato numérico, e transforma estes valores em variáveis fuzzy. Após a etapa de transformação da entrada, os novos valores formam a entrada para o algoritmo C4.5, gerando uma AD, porém com valores fuzzy. Este processo pode ser executado empregando a plataforma WEKA, o que permitiu a geração das mesmas medidas para comparação. Após análise dos resultados para cada base empregando os dois algoritmos foi possível observar que os resultados foram muito similares, exceto pela classificação com taxa de acerto mais alta para o modelo ADF na base de dados Haberman's Survival.

Palavras-chave: classificação, Árvore de Decisão, Lógica *Fuzzy*, Árvore de Decisão *Fuzzy*

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AD Árvore de Decisão

AF Árvore Fuzzy

AM Aprendizado de Máquina

IA Inteligência Artifical

SUMÁRIO

1 INTRODUÇÃO

O termo Aprendizado de Máquina ou, do inglês *Machine Learning*, diz respeito aos sistemas computacionais que buscam desenvolver métodos de "aprender", obter as respostas e conhecimentos automaticamente, partindo de experiências acumuladas durante as operações (ALPAYDIN, 2010). Witten, Frank e Hall (WITTEN; FRANK; HALL, 1999) também definiram aprendizado de máquina como *"As máquinas aprendem quando mudam o seu comportamento de forma que as fazem ter um melhor desempenho no futuro"*. Ou seja, o aprendizado estuda métodos de adquirir boas habilidades buscando melhorias em seu desempenho futuro nos algoritmos, com base em experiências (MITCHELL, 1997).

Ademais, o aprendizado de máquina é dividido em Aprendizado Não-Supervisionado e o Aprendizado Supervisionado. O primeiro é quando o conjunto não possui exemplos rotulados. Ele busca agrupar e analisar elementos similares. Já o Aprendizado Supervisionado consiste em um conjunto de dados com exemplos já previamente rotulados, ou seja, os conjuntos possuem entradas e saídas já esperadas. Seu objetivo é gerar saídas corretas para diferentes conjuntos de novos dados (RUSSELL; NORVIG., 1995). Este é divido em classificação e regressão. A classificação, realiza por meio de algoritmos indutivos, a previsão de saídas corretas dada a entrada de dados, constituída por atributos que juntos determinam uma classe. A classe, é o rótulo determinado no final de uma classificação (QUINLAN, 1992).

Em meio aos algoritmos que desempenham a tarefa de classificação, estão definidos diversos modelos de algoritmos indutivos e para as tomadas de decisões, entre eles a chamada árvore de decisão, sendo comumente empregada para a solução de problemas de classificação. As árvores de decisão recebem como entrada instâncias que compõem um conjunto de atributos e a saída consiste em definir a qual classe cada instância é pertencente. Este modelo de árvore de decisão é muito utilizado em aprendizados de máquina devido ao seu fácil entendimento e por ser muito intuitivo. Vários algoritmos foram criados com base na árvore de decisão, entre eles o ID3 (QUINLAN, 1986), CART (BREIMAN et al., 1984) e C4.5 (QUINLAN, 1996) que são considerados os principais.

Apesar das árvores de decisão serem úteis para a resolução de grande parte dos problemas, elas não abrangem os valores que são considerados imprecisos. Para a representação de tais dados utiliza-se a chamada lógica *fuzzy* (no português, lógica difusa ou nebulosa) proposta por Zadeh (ZADEH, 1965), em 1965. Essa lógica consiste em capturar dados mais vagos, em um intervalo de 0 e 1, ao contrário da lógica *clássica*, que é representada apenas por 0 e 1, associando graus de pertinência aos elementos dos conjuntos. A lógica *fuzzy* trabalha com conjuntos *fuzzy* e regras *fuzzy*. A partir da lógica *fuzzy*, torna-se possível a utilização de árvores de decisão para classificação dos dados considerados imprecisos ou incertos, chamadas de Árvore de Decisão *Fuzzy* (RIBEIRO; CAMARGO; CINTRA, 2013).

Neste trabalho de iniciação científica, foi abordado o tema de aplicação e comparação dos modelos de árvore de decisão indutiva, com base no algoritmo, C4.5 e árvore de decisão *fuzzy* para problemas de classificação. Para tanto foram empregados seis bancos de dados disponíveis no repositório de dados UCI (DUA; GRAFF, 2017), que foram classificados a partir das árvores de decisão, sendo os resultados obtidos comparados a partir das taxa de acerto, da matriz de confusão de cada modelo, e pelo número de regras que cada modelo gerou.

Este relatório está organizado da seguinte maneira:

- Este Capítulo com uma breve introdução do trabalho;
- o Capítulo 2 define os objetivos;
- no Capítulo 3 são apresentados os conceitos preliminares da pesquisa como aprendizado de máquina, Lógica *Fuzzy*, árvores de decisão indutivas e árvores de decisão *fuzzy*.
- o Capítulo 4 detalha a metodologia de desenvolvimento da pesquisa;
- no Capítulo 5 estão os resultados alcançados e uma análise comparativa;
- e, por fim, o Capítulo 6 traz as principais contribuições deste trabalho e sugestões para trabalhos futuros.

2 OBJETIVOS PROPOSTOS

Com base no que foi apresentado no Capítulo 1 e com o intuito de analisar os modelos de classificação do tipo Árvore de Decisão comparando as versões clássica e *fuzzy*, os objetivos desta pesquisa podem ser descritos como se segue:

- Identificar os algoritmos definidos na literatura que empregam a Árvore de Decisão com a lógica *Fuzzy*, identificando as características principais de cada modelo;
- Seleção dos modelos para investigação a partir de um conjunto de experimentos em diferentes bases de dados;
- Seleção das bases de dados para definição dos experimentos;
- Definição da metodologia de testes para análise e comparações do algoritmo de uma Árvore de Decisão *Fuzzy* com um modelo clássico;
- Execução dos experimentos e avaliação a partir do número de instâncias classificadas corretamente para cada conjunto de dados.

Pretende-se ainda avaliar o emprego da lógica *fuzzy* na tarefa de classificação, atentando-se para a adequação da lógica *fuzzy* na representação de dados que envolve imprecisão e incerteza.

3 REFERENCIAL TEÓRICO

3.1 INTELIGENCIA ARTIFICIAL E APRENDIZADO DE MÁQUINA

Segundo (FERNANDES, 2003) Inteligencia Artificial (IA) vem do latim inter e legere, que significam, respectivamente, entre e escolher, ou seja, é algo que o homem é capaz de escolher entre uma coisa ou outra. O termo foi denominado por John McCarthy (MCCARTHY, 1990) e ele a definiu como a ciência e engenharia de produzir máquinas inteligentes. E é também considerada uma área da computação que busca métodos similares ao do raciocínio do homem. A IA pode ser divida nas seguintes principais áreas (GROOVER; NAGEL; ODREY, 1989):

- Aprendizado de Máquina
- Robótica
- Automação de Raciocínio
- Representação do Conhecimento
- Processamento de Linguagem Natural
- Visão Computacional

Este trabalho de iniciação científica concentra-se no campo do Aprendizado de Máquina (AM). Sendo considerado um subcampo da Inteligência Artificial, o objetivo do AM é construir modelos computacionais que podem adaptar-se e aprender a partir da experiência (DIETTERICH, 2009). Ou seja, o AM pode ser definido como um aprendizado por repetitivas execuções de tarefas, fazendo com que o algoritmo retorne respostas com base nas experiências anteriores. Assim, o AM está relacionado a capacidade dos computadores desenvolverem um conjunto de regras, ou padrões, e gerar certas continuidades e resoluções a essas específicas tarefas. Busca-se por um modelo generalizado, quando este modelo é exibido a dados diferentes daqueles empregados no processo de aprendizado, existe capacidade de adaptação, ou generalização, permitindo que a máquina apresente resultados adequados.

A Figura 1 apresenta um ramo do AM que é o chamado Aprendizado Supervisionado, que é divido em Classificação e Regressão.

Figura 1 – Diagrama da divisão de tipos de aprendizado de máquina

O Aprendizado de Máquina Supervisionado consiste em um modelo onde tem-se um conjunto de dados rotulados, e seu objetivo é encontrar parâmetros a partir do treinamento para ajustar um modelo e prever rótulos desconhecidos em um conjunto de dados de teste (RUSSELL; NORVIG., 1995). Este trabalho tem como foco principal a tarefa de classificação, que é detalhada na Seção 3.2.

3.2 CLASSIFICAÇÃO

Em um contexto de aprendizado de máquina, o processo de classificação ocorre por meio de um algoritmo indutivo e seu objetivo é prever o rótulo de dados novos de entrada com base em dados já conhecidos e rotulados previamente.

Um conjunto de dados de entrada possui vários exemplos e cada um deles é constituído por atributos que juntos determinam uma classe, ou seja, o rótulo. Logo, a classe é o que determina o fim de uma classificação (QUINLAN, 1992).

A entrada de um algoritmo indutivo é o o conjunto de treinamento empregado com o objetivo de alcançar as associações dos atributos com suas respectivas classes. Esse processo resulta-se em um modelo classificador.

3.3 LÓGICA *FUZZY*

O conceito da lógica *fuzzy* (definida no portugûes como difusa ou nebulosa) foi introduzido no ano de 1965 pelo professor Lofti Zadeh (ZADEH, 1965).

A lógica *fuzzy* admite multi-valores, capturando dados mais vagos variados entre 0 (falso) e 1 (verdadeiro), sendo esta a principal diferença da lógica clássica, logo que ela busca se aproximar do mundo real em que não há apenas respostas certeiras. Além da possibilidade de apresentar um meio termo, ainda por meio desta lógica é possível mensurar um grau de aproximação com a solução correta.

A saída de um modelo que usa a lógica *fuzzy*, pode ser manipulada através do conjunto *fuzzy* e das regras *fuzzy*, que englobam o sistema *fuzzy*. Na Figura 2 podem ser observados os

componentes de um sistema *fuzzy*.

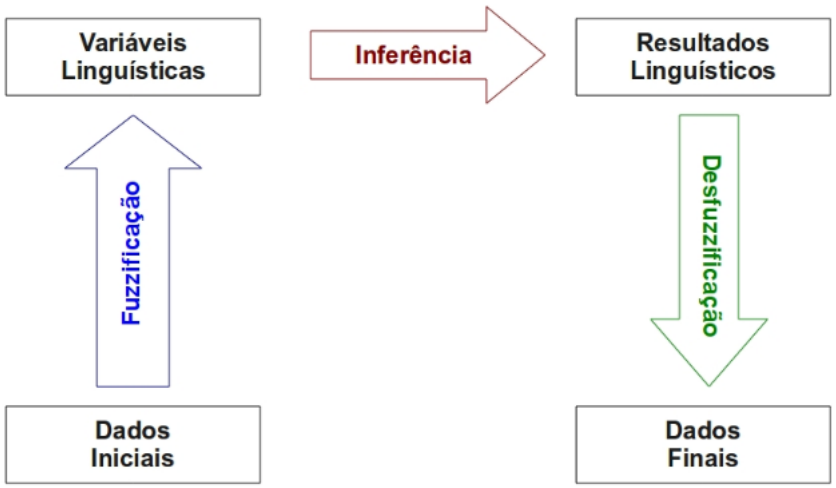

Figura 2 – Operações do sistema *Fuzzy*. Adaptada de: (COX, 1994)

(ZADEH, 1965) definiu que os sistemas *fuzzy* possuem pelo menos uma variável representada pela teoria dos conjuntos *fuzzy*, logo que ela se torna útil nos trabalhos com dados que envolvem imprecisão e incerteza.

3.3.1 CONJUNTO *FUZZY*

Segundo a teoria clássica dos conjuntos, um elemento pertence ou não a um determinado conjunto, porém em certos casos, um elemento pode pertencer parcialmente ao conjunto. Esse fato evidencia a necessidade de uma representação diferente, em que conjuntos podem conter o mesmo elemento, com graus de pertinência diferentes. Dessa forma, torna-se necessária a pertinência parcial de um elemento, que é representada através de uma função onde é associado o grau de pertinência de um elemento do conjunto. Estas funções são denominadas funções de pertinência e o grau que retornam para a pertinência dos elementos nos conjuntos está, normalmente, no intervalo [0,1] (PEDRYCZ; GOMIDE, 1998).

As funções comumente empregadas são as funções triangulares, trapezoidais e gaussianas, representadas na Figura 3.

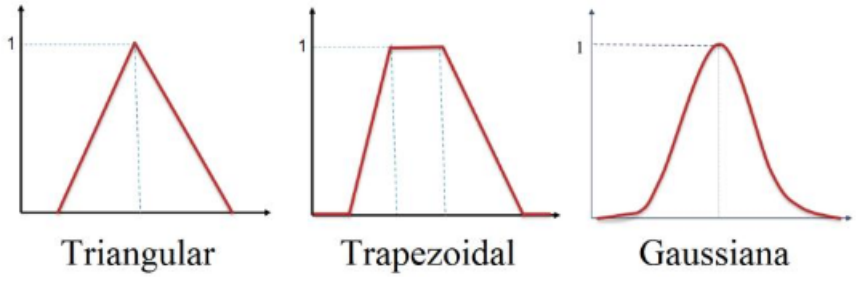

Figura 3 – Gráficos Função Permanência. Fonte: (PINHO, 2016)

Além disso, o conjunto *fuzzy*, possui as variáveis linguísticas, visto como o primeiro passo na Figura 2. Segundo (PEDRYCZ; GOMIDE, 1998), uma variável linguística é uma variável cujos valores são nomes de conjuntos *fuzzy*, e através destas variáveis é possível caracterizar os conceitos empregados. Por exemplo, uma variável do tipo temperatura pode ter valores baixos, médios ou altos, e as variáveis linguísticas empregam os conjuntos fuzzy para expressar com maior adequação o significa do "baixo", "médio"e "alto". Este conceito está ilustrado no gráfico da Figura 4, onde podem ser obtidos os valores de pertinência para cada valor da variável.

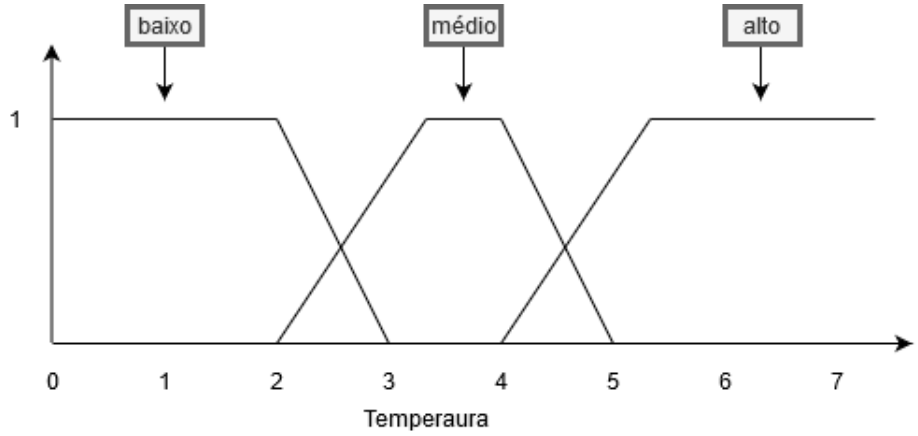

Figura 4 – Variável Linguística da Temperatura

Num sistema *fuzzy*, o processo de inferência (Figura 2), é onde são formadas as regras baseadas nas variáveis linguísticas.

3.3.2 REGRAS *FUZZY*

A etapa de criação de regras *fuzzy* pode ser dividida de acordo com a Figura 5.

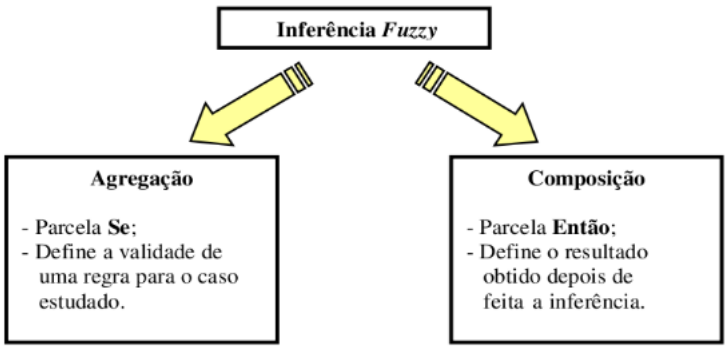

Figura 5 – Etapas da Inferência *Fuzzy*. Fonte: (JANé, 2004)

Assim, a lógica *fuzzy* apresenta o seguinte tipo de regra, apresentando o conhecimento de forma mais clara (KLIR; YUAN, 1995) :

SE antecedente, **ENTÃO** consequente

Expressando uma relação de condição, ou seja, a veracidade do antecedente implica no consequente. Essa expressão também pode ser dada de forma composta, como por exemplo:

SE antecedente 1 e **SE** antecedente 2 **ENTÃO** consequente

3.4 ÁRVORE DE DECISÃO

As árvores de decisão são modelos estatísticos que utilizam um treinamento supervisionado para a classificação e previsão de dados (QUINLAN, 1986). Elas empregam o método de divisão e conquista, ou seja, dividir um problema complexo em um simplificado. Estas também, constituem uma técnica bastante popular na realização da tarefa de classificação devido a certas características (KOTHARI; DONG, 2001) como por exemplo: são geradas de forma rápida, são facilmente aplicadas em domínios numéricos, e suas decisões são facilmente compreendidas.

Uma árvore de decisão, é considerada um grafo acíclico direcionado a um nó de divisão ou um nó folha. Um nó folha, já é rotulado com uma função, e nele são considerados apenas os valores da variável, ou seja, nele inclui a classe que identifica as características definitivas. Cada nó da árvore compreende um teste condicional de acordo com os valores de um atributo específico (FACELI, 2011). Apesar do teste a partir de um nó comparar o valor do atributo com uma constante, pode haver atributos diferentes sendo comparados ou uma função que combina dois ou mais atributos é utilizada (WITTEN; FRANK; HALL, 1999). Seguem exemplos de teste condicionais:

- Altura $> 1{,}70m$
- Camiseta $\in \{azul, verde\}$

As propriedades das árvores de decisão envolvem:

- com exceção do nó raiz, todos os nós possuem um único pai;
- com exceção do nó folha, todos os nós possuem sucessores.

A Figura 6 mostra um exemplo genérico de uma árvore de decisão.

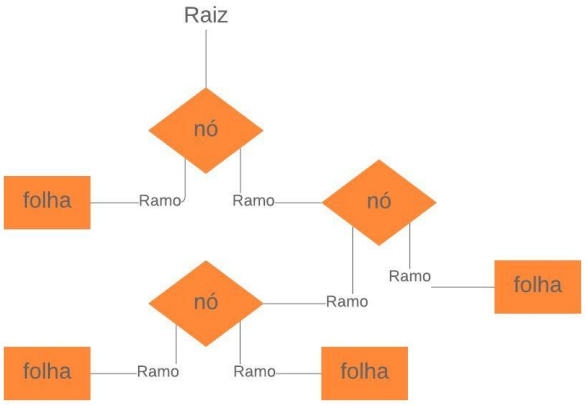

Figura 6 – Árvore de decisão genérica

O algoritmo 1 foi apresentado por (WITTEN; FRANK; HALL, 1999) e descreve os passos para a formação de uma árvore de decisão. A entrada consiste em um conjunto de dados **D**. Logo em seguida, é avaliado o critério de parada, e caso necessário, haverá mais divisões onde é escolhido o atributo que maximiza a medida de impureza (passo 5), como por exemplo, o ganho de informação. As medidas de impureza medem a homogeneidade dos subconjuntos gerados, caso a divisão ocorra, os mais homogêneos apresentam maior grau de pureza. E, por fim, no passo 7, a função GeraÁrvore é aplicada de forma recursiva em cada subconjunto de **D**.

Algoritmo 1: Algoritmo geração de árvore de decisão

Entrada: Conjunto de treinamento D

1 Função GeraÁrvore(D);
2 **se** *critério de parada (**D**) = Verdadeiro* **então**
3 **Retorna:** um nó folha rotulada com a constante que minimiza a função perda;
4 **fim**
5 Escolha um atributo que maximiza o critério de divisão em **D**;
6 **para cada** *partição dos exemplos* $\mathbf{D}_i$ *baseado nos valores do atributo escolhido* **faça**
7 Induz uma subárvore $Árvore_i = GeraÁrvore(\mathbf{D}_i)$;
8 **fim**
9 **Retorna:** *Árvore* contendo um nó de decisão baseado no atributo escolhido, e descendente $Árvore_i$

Certos métodos de indução, podem gerar um problema conhecido como *overfitting*. Quando isso ocorre, a árvore de decisão indutiva busca classificar corretamente os dados apresentados no treinamento e, porém, quando dados não rotulados são apresentados para o modelo de classificação gerado o desempenho é ruim, ou seja o modelo não tem a capacidade de dar respostas corretas.

A solução encontrada para esse problema, é conhecido como técnicas de poda, e é considerada uma parte importante na construção de uma árvore de decisão, logo que ela tira as partes que não contribuem para a precisa classificação, o desempenho final tende a melhorar.

Dessa forma, foram definidos algoritmos para a criação da árvore de decisão, sendo os mais conhecidos: ID3, C4.5 e CART (detalhados na Seção 3.4.1).

3.4.1 ALGORITMOS

Primeiramente, o algoritmo ID3 que foi desenvolvido por Quilan em 1986 (QUINLAN, 1986), é o primeiro algoritmo feito para árvore de decisão. Este é um algoritmo recursivo, procurando no conjunto de atributos aquele que melhor divide os exemplos de treinamento. A limitação desse tipo de algoritmo, é que ele lida apenas com dados numéricos.

Em seguida veio o algoritmo CART (*Classification and Regression Trees*), desenvolvido por Leo Breiman (BREIMAN et al., 1984). Este modelo consiste em uma técnica não-parametrizada que é capaz de induzir dois tipos de árvores: de regressão e de classificação. As árvores que são geradas por esse tipo de algoritmo são do tipo binária, se baseando na questão do "sim"ou "não", simplesmente. Suas características principais são:

- A capacidade de dividir os nós com base em um conjunto de regras (Índice de Gini com base na entropia);
- Decisão de quando uma árvore está completa;
- Associação um nó terminal a uma classe.

E, por fim, o modelo C4.5, proposto no ano de 1992 por Quinlan (QUINLAN, 1996), emprega a técnica de dividir para conquistar. Utilizando um conjunto de treinamento $D = (x_i, y_i)$, a árvore é induzida seguindo os passos do Algoritmo 1. Ele consiste em ignorar valores desconhecidos, utilizando a razão de ganho para selecionar melhor o atributo que divide os exemplos (o critério de divisão citado no passo 5 do Algoritmo 1). Assim, o atributo que possuir a maior razão de ganho passa a ser considerado o nó raiz e define os próximos testes a serem realizados. Como principais características do algoritmo C4.5 destacam-se:

1. A utilização da entropia e a razão de ganho como critérios de seleção;
2. Tratamento de atributos discretos e contínuos;
3. Utilização do método da pós-poda para solução do problema de *overfitting*.

A razão de ganho de um dado atributo informa quanta informação pode ser adquirida ao particionar um conjunto de exemplos segundo os valores deste atributo. Quanto mais homogêneos forem os subconjuntos resultantes, melhor. Este cálculo é definido pela normalização do ganho de informação, que é calculado pela redução em entropia (QUINLAN, 1992).

Assim, dado um conjunto (S) que pode ter n classes distintas, a entropia é dada pela Equação 3.1:

$$Entropia(S) = -\sum_{i=1}^{n} p(C_n,S) log_2(p(C_n,S)) \tag{3.1}$$

Onde,

$$p(C_n,S) = \frac{freq(C_n,S)}{|S|} \tag{3.2}$$

Em que $freq(C_n,S)$ representa os exemplos em S com a classe C_n e $|S|$ os exemplos totais do conjunto (S).

E o ganho de informação de um atributo A em um conjunto S resulta na medida da diminuição esperada da entropia ao utilizar o atributo A para a divisão. Este é dado pela Equação 3.3:

$$Ganho(S,A) = Entropia(S) - \sum_{x \in P(A)}^{n} |S_x| \cdot |S|^{-1} \tag{3.3}$$

Onde:

- $P(A)$ é um conjunto de valores que A pode assumir;
- x é um elemento do conjunto;
- S_x é um subconjunto de S.

Dadas as Equações 3.1 e 3.3 é possível obter o ganho de informação que é definido pela Equação 3.4:

$$RazaoDeGanho(A) = \frac{Entropia(S)}{Ganho(S,A)} \tag{3.4}$$

Dessa forma, a razão de ganho define o atributo que será atribuído a cada nó.

3.5 ÁRVORE DE DECISÃO *FUZZY*

A partir dos conceitos apresentados acerca da árvore de decisão, foram desenvolvidos estudos, e uma das extensões é o emprego da lógica *fuzzy*, constituindo um modelo de árvore de decisão *fuzzy* para a representação de dados imprecisos.

Um dos algoritmos que foi desenvolvido neste formato é o FuzzyDT (CINTRA, 2016), proposto por Marco E. Cintra. Este modelo é baseado no C4.5 (QUINLAN, 1996), porém, os conjuntos *fuzzy* são previamente definidos, ou seja, antes da entrada de dados no algoritmo C4.5. Este é o modelo selecionado para estudo neste trabalho. O emprego da técnica é detalhada na Seção 4.2

Porém, assim como as árvores de decisão clássicas, uma árvore *fuzzy* (AF) também apresenta certos problemas, como na definição dos conjuntos *Fuzzy*. Este fato se deve a esta definição ser muito abrangente, um conjunto *fuzzy* não possui um valor exato, e também não há um consenso de quantas variáveis ele deve possuir.

Outro problema a ser citado é visto na publicação *"A comparative analysis of pruning strategies for fuzzy decision trees"* (RIBEIRO; CAMARGO; CINTRA, 2013), o número de regras de uma AF varia para cada base de dados que é analisada. Isso acontece devido ao primeiro problema que foi descrito, ou seja, pela quantidade de conjuntos *fuzzy*. Quando um atributo é selecionado para um nó divisão, a quantidade de ramos que será gerada, isto é, as regras que serão geradas, é equivalente ao número de conjuntos associados a ele.

4 DESCRIÇÃO DAS ATIVIDADES DESENVOLVIDAS

Para o desenvolvimento desta pesquisa de iniciação cientifica, foi utilizado o pacote de *software* WEKA (*Waikato Environment for Knowledge Analysis*), implementado em Java, formado por um conjunto de implementações de algoritmos de diversas técnicas de classificação e agrupamento de dados (UNIVERSITY OF WAIKATO, 2010).

Dentro do WEKA, foi utilizado o algoritmo de classificação J48, que consiste em gerar uma árvore de decisão. A partir de um conjunto de dados de entrada, são gerados como saída a matriz de confusão, a taxa de acerto do classificador, e o número de nós folha e tamanho da árvore obtida, conforme fluxograma da Figura 7. Essas três saídas constituem a forma de avaliação para cada modelo gerado neste trabalho, para que os resultados alcançados possam ser comparados, permitindo a comparação da árvore de decisão indutiva e a árvore *fuzzy*.

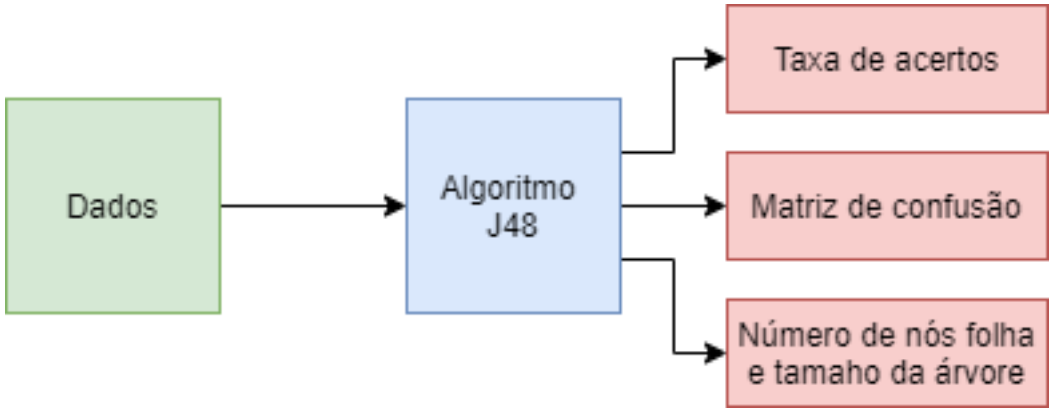

Figura 7 – Fluxograma da entrada e saída dos dados

Neste capítulo estão descritas as entradas e saídas de dados dos dois modelos: árvore de decisão indutiva e a árvore de decisão *fuzzy*, assim como suas aplicações em seis bases de dados de aprendizado de máquina selecionadas para as avaliações e comparações dos modelos, sendo todos elas fornecidas pelo repositório de dados UCI - *Machine Learning Repository*(DUA; GRAFF, 2017):

1. Pima Indians Diabetes;
2. Glass;
3. Haberman's Survival;
4. Heart Disease;
5. Iris;
6. Wine.

4.1 DESENVOLVIMENTO DA ÁRVORE DE DECISÃO C4.5

Para a geração das árvores de decisão indutivas foi utilizado o algoritmo de classificação J48 fornecido pelo WEKA. O J48 é baseado no algoritmo C4.5 de Quinlan, no qual consiste em utilizar a entropia e a razão de ganho para a seleção de atributos, sendo que o atributo com a maior razão de ganho, torna-se o nó raiz e é a base para os próximos testes.

A forma de entrada dos dados utilizada foi a validação cruzada, ou seja, todos os dados são separados em conjuntos de treinamento e teste. Em seguida, os dados são embaralhados e divididos em um total de 10 grupos, dessa forma, são treinados 10 classificadores variando a entrada, onde para cada modelo 9 pastas constituem o conjunto de treinamento e 1 pasta é o conjunto de teste (Figura 8):

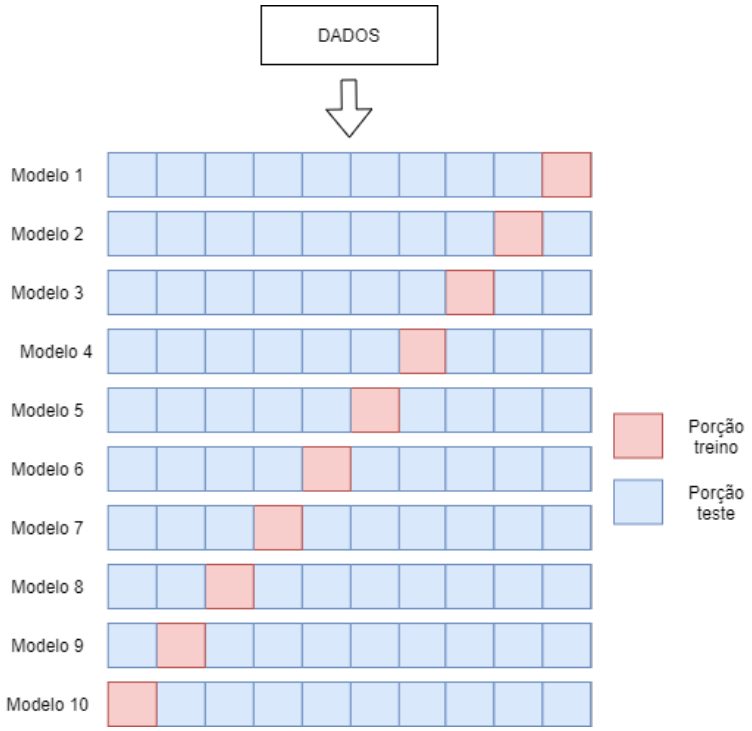

Figura 8 – Validação Cruzada com 10 pastas

Para cada modelo é gerado como saída uma AD e as medidas de desempenho (taxa de acerto, matriz de confusão, e número de nós folha e o tamanho da árvore). Para uma ilustração do que foi descrito, considere o conjunto *Container Crane Controller* (FERREIRA et al., 2016) fornecido pela UCI (DUA; GRAFF, 2017). Os contêineres são transportados de um ponto a outro por meio de cabos. Esses cabos formam um ângulo de abertura que interfere a operação em altas velocidades, o que pode causar um acidente. Este conjunto descrito possui dois atributos de entrada, velocidade e ângulo, e um atributo de saída, a potência. Seus atributos e instâncias são apresentados na Tabela 1.

Tabela 1 – Tabela dos atributos do conjunto Container Crane Controller

Velocidade	Ângulo	Potência
1	-5	0,3
2	5	0,3
3	-2	0,5
1	2	0,5
2	0	0,7
6	-5	0,5
7	5	0,5
6	-2	0,3
7	2	0,3
6	0	0,7
8	-5	0,5
9	5	0,5
10	-2	0,3
8	2	0,3
9	0	0,5

A saída para este conjunto conta com a árvore de decisão apresentada na Figura 9.

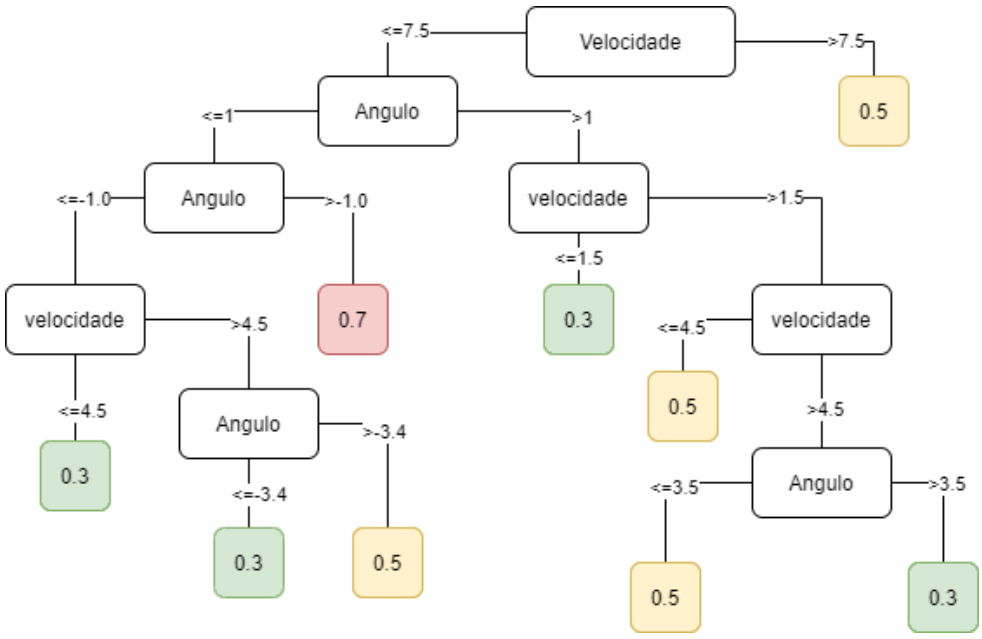

Figura 9 – Árvore de Decisão do conjunto Container Crane Controller

4.2 DESENVOLVIMENTO DA ÁRVORE DE DECISÃO *FUZZY*

Para o desenvolvimento deste modelo, foi utilizado como base o algoritmo FuzzyDT (CINTRA; CAMARGO, 2010). Esse algoritmo transforma os dados de entrada em valores *fuzzy*.

O algoritmo FuzzyDT foi elaborado por (CINTRA; CAMARGO, 2010). Ele utiliza as mesmas características do algoritmo C4.5, a entropia e o ganho de informação, para a seleção dos atributos, além de utilizar também a estratégia de indução para a partição recursiva gerando

ramificações até que uma classe seja atribuída a determinada ramificação. A diferença do algoritmo FuzzyDT para o C4.5 é que os atributos contínuos são transformados em conjuntos *fuzzy* antes que ocorra a indução da árvore. A Figura 10 apresenta o fluxograma para o FuzzyDT.

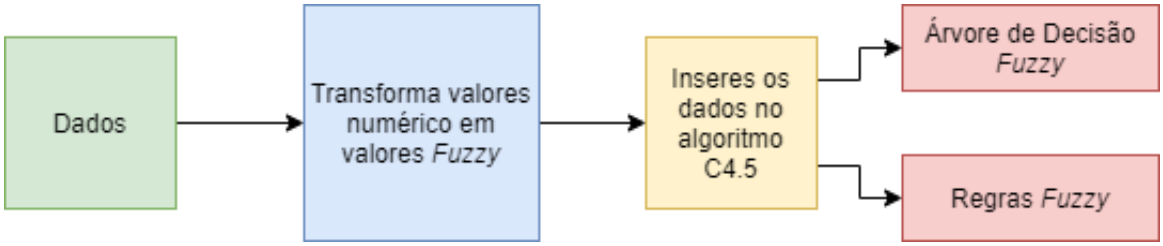

Figura 10 – Fluxograma de entrada e saída de dados do FuzzyDT

E o algoritmo 2 descreve o modelo FuzzyDT.

Algoritmo 2: Algoritmo fuzzyDT por Cintra e Camargo (CINTRA, 2016)

Entrada: Conjunto de treinamento

1 Determinar o conjunto de dados *fuzzy* em que cada atributo contínuo será particionado;
2 Substituir os valores contínuos por termos linguísticos do conjunto *fuzzy* com a maior compatibilidade dos dados de entrada;
3 Calcula a entropia e o ganho de informação para cada atributo e dividi o conjunto de treinamento definindo os testes que serão realizados nos nós, até que todos os atributos seja utilizado;
4 Aplicar o processo de poda utilizando 25% de confiança.

O modelo empregado está implementado em linguagem de programação Java, com base no algoritmo 2, onde os parâmetros recebidos são: a base de dados, o método de partição e a taxa confiança.

Para todas as bases de dados, o método de partição escolhido foi o de (WANG; MENDEL, 1992). Esse método consiste em 3 passos:

1. Dividir o espaço de entrada em regiões *fuzzy*;
2. Gerar as regras *fuzzy* a partir dos pares de dados fornecidos;
3. Remover as regras conflitantes.

E para a confiança, foi utilizado uma taxa de 25%. A saída do algoritmo fornece a árvore de decisão *fuzzy* além de fornecer também as regras *fuzzy*.

Para uma demonstração deste algoritmo, considere o conjunto *Container Crane Controller*, descrito na seção 4.1. Conforme os passos do algoritmo FuzzyDT, é necessário, primeiramente, definir um conjunto de dados *Fuzzy* em que os atributos são particionados. O próprio algoritmo é capaz de realizar essa tarefa, e para esta base de dados foram considerados três conjuntos

fuzzy para os atributos. A geração desses conjuntos resultou em três funções de pertinências triangulares, conforme as Figura 11, 12 e 13.

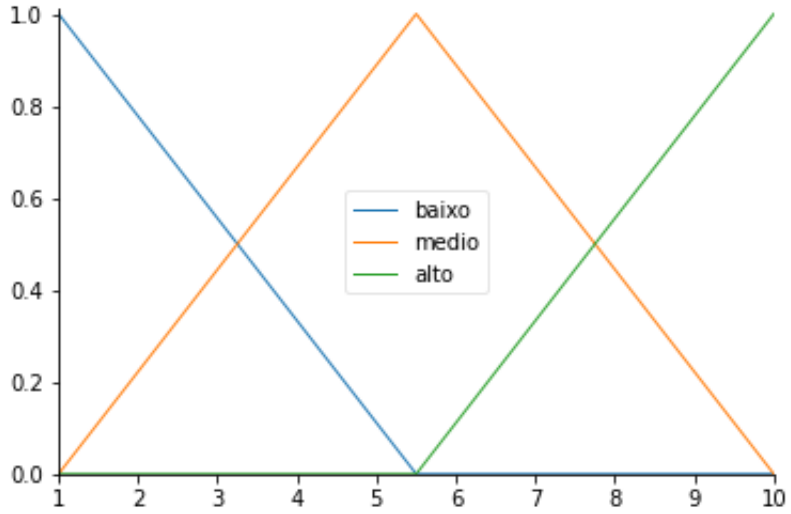

Figura 11 – Função pertinência que representa o atributo velocidade

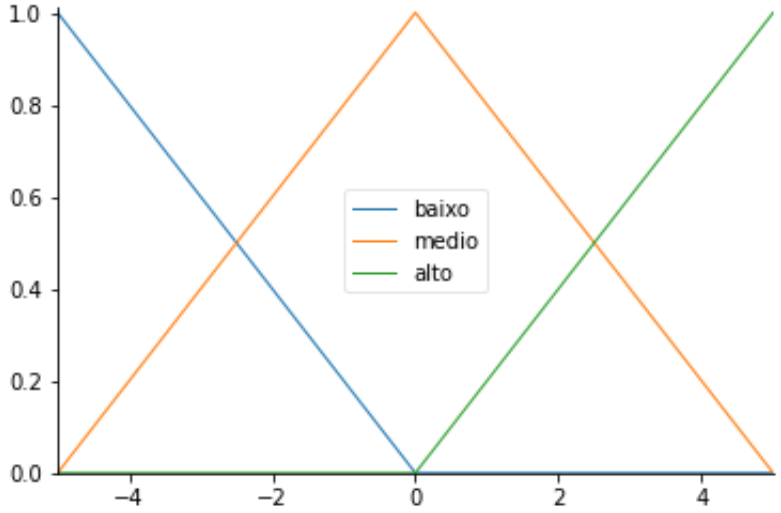

Figura 12 – Função pertinência que representa o atributo ângulo

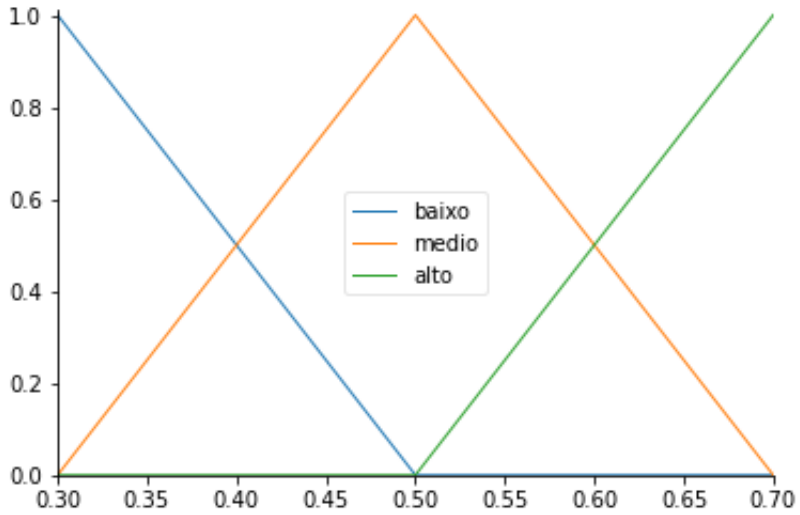

Figura 13 – Função pertinência que representa o atributo potência

Assim, os valores contínuos serão substituídos pelos termos linguísticos definidos pelos conjuntos *fuzzy*, conforme na Tabela 2.

Tabela 2 – Tabela de variáveis *fuzzy* do conjunto Container Crane Controller

Velocidade	**Ângulo**	**Potência**
baixo	baixo	baixa
baixo	alto	baixa
baixo	médio	média
baixo	médio	média
baixo	médio	alta
médio	baixo	média
médio	alto	média
médio	médio	baixa
médio	médio	baixa
médio	médio	alta
médio	baixo	média
alto	alto	média
alto	médio	baixa
médio	médio	baixa
alto	médio	média

A Tabela 2 passa a ser a entrada de dados no algoritmo C4.5 com os atributos na forma nominal.

Assim, a saída do FuzzyDT é a geração das regras *fuzzy*, a árvore *fuzzy*, além de um arquivo ARFF. Este arquivo permite a visualização dos dados na forma nominal, assim como também torna-se possível a inserção e análise da saída destes novos dados no algoritmo J48 do WEKA, assim como mostrado no fluxograma da Figura 7. A AF gerada é demonstrada na Figura 14.

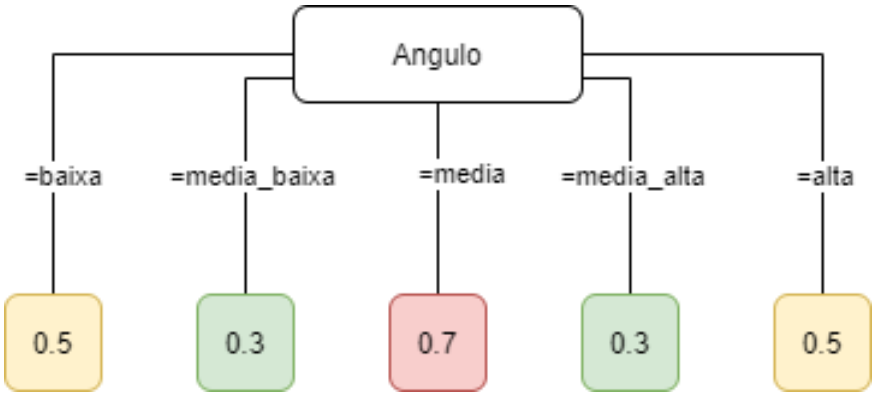

Figura 14 – Árvore de conjunto Container Crane Controller

4.3 AVALIAÇÃO E COMPARAÇÃO DA CLASSIFICAÇÃO OBTIDA

Para a avaliação das classificações obtidas, são empregados três critérios: a taxa de acerto do classificador, a matriz de confusão e, também, através do número de nós folha e do tamanho da árvore.

As taxas de acertos e erros são definidas a partir do número total de instâncias que foram classificadas corretamente, assim como o número total de erros de classificação, dentro de um conjunto onde foi utilizada a validação cruzada como meio de entrada.

A matriz de confusão representa os erros e acertos de cada classe através de uma matriz, onde cada linha é uma classe do problema e cada coluna é a contabilização da classe predita pelo classificador para um elemento do conjunto. Assim é possível identificar que a soma da diagonal principal representa os acertos do classificador, além de ser possível identificar onde o erro acontece. A Figura 15 representa a matriz de confusão generalizada, onde:

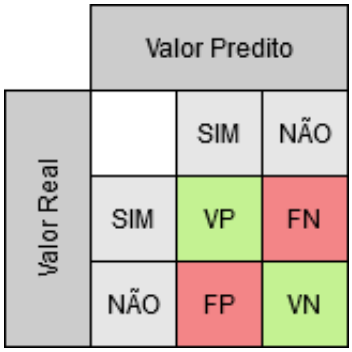

Figura 15 – Matriz de confusão generalizada

- Verdadeiro positivo (VP): ocorre quando houve a classificação correta da classe positiva;
- Verdadeiro negativo (VN): ocorre quando houve a classificação correta da classe negativo;
- Falso positivo (FP): erro no qual o modelo previu uma classe positiva, quando era para ser negativa;

- Falso negativo (FN): erro no qual o modelo previu uma classe negativa, quando era para ser positiva.

Para exemplificar, considere novamente a base de dados *Container Crane Controller* (FERREIRA et al., 2016). A matriz de confusão do conjunto é definida na Figura 16.

		Valor Predito		
		A	B	C
Valor Real	A	2	4	0
	B	6	0	0
	C	2	0	0

Figura 16 – Matriz de confusão Container Crane Controller

Dessa forma, tem-se que:

- O modelo classificou corretamente 2 instâncias da classes A;
- O modelo classificou corretamente 0 instâncias da classe B;
- O modelo classificou corretamente 0 instâncias da classe C;
- O modelo classificou incorretamente 8 instâncias da classe A;
- O modelo classificou incorretamente 4 instâncias da classe B;
- O modelo classificou incorretamente 0 instâncias da classe C.

E, como último método de análise, da Figura 14, pode-se observar que esta árvore gerou 5 nós folha e a árvore tem um tamanho igual a 6.

5 RESULTADOS OBTIDOS E ANÁLISE

Para a avaliação das árvores de decisão clássica (árvore indutiva) e *fuzzy* foram analisados os resultados obtidos a partir da execução dos modelos empregando seis diferentes conjuntos de dados. Todos os conjuntos foram obtidos a partir do repositório de dados UCI (DUA; GRAFF, 2017), sendo bases de dados numéricas. As seções que se seguem apresentam uma breve descrição de cada conjunto, bem como os resultados alcançados e a comparação dos modelos.

5.1 IRIS

O banco de dados Iris foi apresentado em 1963 no artigo *"The Use of Multiple Measurements in Taxonomic Problems"* (FISHER, 1936) e consta na classificação de três espécies da flor Iris. Nesse banco de dados há 50 amostras de cada espécie da flor e os seguintes atributos: comprimento da sépala, largura da sépala, comprimento da pétala e largura da pétala. A partir dos dados numéricos desses atributos é possível distinguir três tipos de flor Iris: Setosa, Virgínica e Versicolor. Assim, há um total de 150 instâncias classificadas.

Na Tabela 3 são apresentadas as taxas de acertos e na Figura 17 observa-se a matriz de confusão para o conjunto Iris.

Tabela 3 – Taxa de acertos do conjunto Iris

	C4.5	FuzzyDT
Taxa de acertos	142 (94.6667 %)	138 (92 %)

Figura 17 – Matriz Confusão do conjunto Iris

Esta é a base de dados que apresentou maior número de acertos em instâncias classificadas corretamente, com aproximadamente 95% de acerto. Este fato é melhor notado na Figura 17, em que é observado que, por exemplo, para a classe A, que seria a Setosa, no modelo C4.5,

houve apenas uma instância que foi classificada como Versicolor, enquanto no FuzzyDT, não teve nenhuma instância classificada como incorreta. Para a classe B, a Virginica, o melhor resultado também foi dado pelo modelo FuzzyDT, onde a diferença é dada por apenas uma instância. Já para a última classe, a Versicolor, a diferença é um pouco maior, com 8 instâncias classificadas corretamente a mais no modelo C4.5, e 3 classificadas como Virginica.

A Figura 18 mostra a árvore de decisão desta base de dados gerada a partir do modelo clássico.

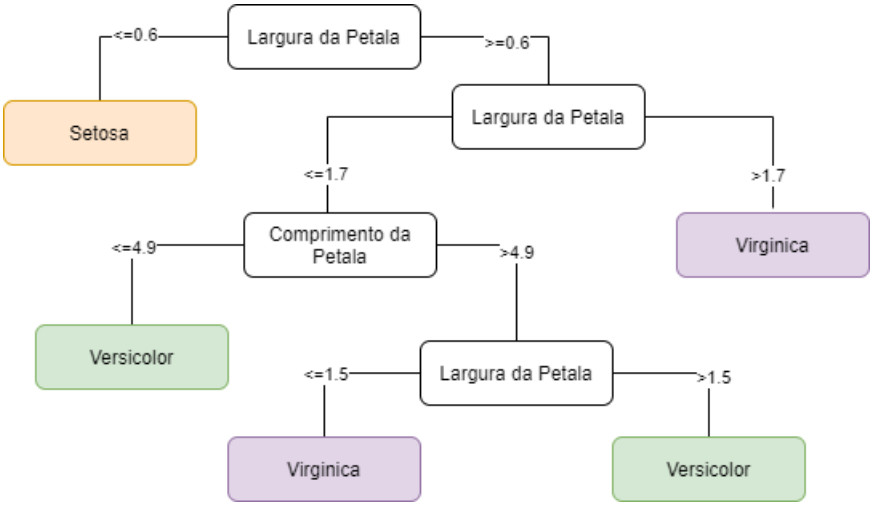

Figura 18 – Árvore de Decisão do conjunto Iris

E a Figura 19 apresenta a árvore *fuzzy*.

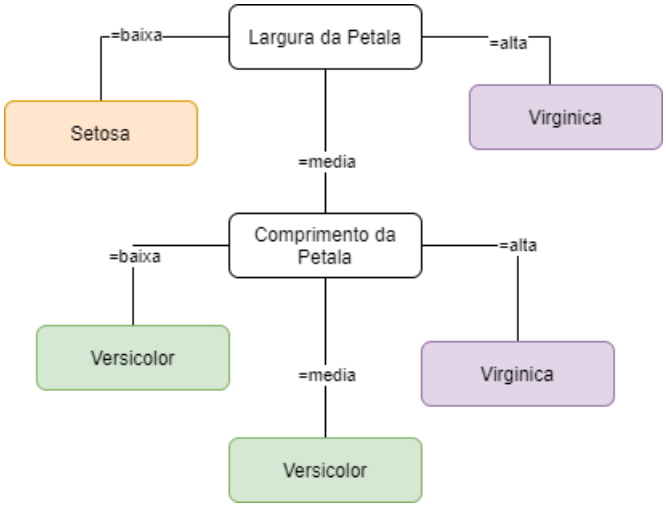

Figura 19 – Árvore *Fuzzy* do conjunto Iris

Tanto na Figura 18 como na Figura 19, os atributos são representados pelos nós e seus ramos contém as regras da classificação que está associada ao nó folha, ou seja, o rótulo de classe. Essas regras podem ser expressar no formato "se..., então ...".

Por exemplo, na Figura 18 tem-se a seguinte regra:

SE Largura da Petala <= 0.6, **ENTÃO** classe = Setosa

E na Figura 19 a seguinte regra pode ser gerada:

SE Largura da Petala = baixa, **ENTÃO** classe = Setosa

Logo, a Árvore *Fuzzy* para este conjunto apresenta um número menor de nós folhas, ou seja, um menor número de regras. Assim como um tamanho de árvore menor.

5.2 PIMA INDIANS DIABETES

Essa base de dados é originalmente do *National Institute of Diabetes and Digestive and Kidney Diseases* (Instituto Nacional de Diabetes e Doenças Digestivas e Renais), e seu objetivo é gerar um modelo para prever, considerando uma pessoa com um conjunto de valores para os atributos, se possui ou não diabetes. Este conjunto apresenta restrições como: todos os pacientes são mulheres e com, no mínimo, 21 anos de herança indígena Pima.

Este conjunto possui 767 instâncias e 9 atributos de entrada: número de vezes que ficou grávida, concentração de glicose, pressão arterial diastólica (mmHg), espessura da dobra da pele do tríceps(mm), insulina sérica de 2 horas (mu U / ml), índice de massa corporal $(peso/(altura)^2)$, função que avalia a probabilidade de diabetes com base no histórico familiar e a idade. E o atributo de saída é 0 ou 1, indicando não tem diabetes ou tem diabetes, respectivamente.

A Tabela 4 apresenta a taxa de acerto para os dois modelos, C4.5 e o FuzzyDT.

Tabela 4 – Taxa de acertos do conjunto Pima Indians Diabetes

	C4.5	FuzzyDT
Taxa de acertos	567 (73,8281%)	564 (73,5332%)

A partir da Tabela 4, é notório que o modelo C4.5 obteve uma melhor classificação, classificando 3 instâncias a mais corretamente e com uma diferença de 0,2949%, em comparação ao FuzzyDT.

A Figura 20 demonstra a matriz de confusão obtida para os dois modelos.

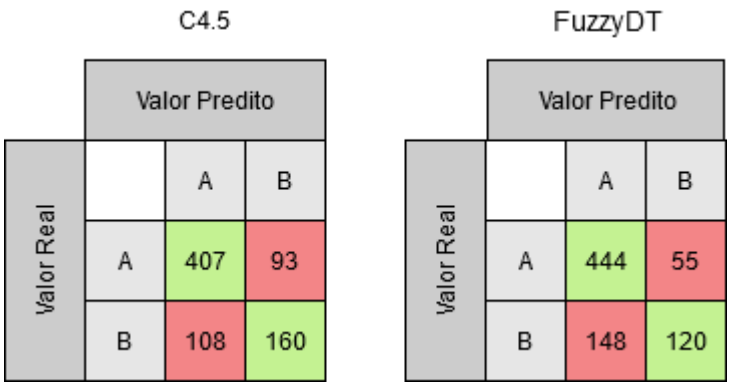

Figura 20 – Matriz confusão do conjunto Pima Indians Diabetes

A partir dessa matriz da Figura 20 é possível observar que a classe A obteve melhor desempenho no FuzzyDT com 444 instâncias classificadas adequadamente, enquanto o C4.5 obteve 407. O contrário ocorreu na classe B, onde o C4.5 obteve vantagem, com 160 classificações corretas e 120 corretas para o FuzzyDT.

Para este conjunto, a árvore apresentou, no modelo C4.5, um total de 20 nós folhas e o tamanho da árvore foi igual a 39 nós. Já para a árvore *fuzzy*, foram 21 nós folhas e a árvore no tamanho de 29. Logo, o C4.5 apresentou uma regra a menos do que o FuzzyDT.

5.3 GLASS

A base de dados *Glass*, define diferentes tipos de vidro a partir dos seguintes atributos: índice de refração, sódio, magnésio, alumínio, silício, potássio, cálcio, bário e ferro. E o tipo de vidro, que é a saída desse conjunto, possui 7 classes, que são:

1. *building windows float processed* (vidros de veículos flutuantes processados);
2. *building windows non float processed (vidros de veículos não flutuantes processados)*;
3. *vehicle windows float processed* (vidros de veículos flutuantes processados);
4. *vehicle windows non float processed* (vidros de veículos não flutuantes processados);
5. *containers* (recipiente);
6. *tableware* (talheres);
7. *headlamps* (faróis).

Este banco de dados possui um total de 9 atributos e 214 instâncias.

Observando a Tabela 5, analisa-se que a diferença entre os modelos foi mínima. O modelo C4.5 obteve apenas uma instância a mais classificada corretamente em comparação ao modelo FuzzyDT.

Tabela 5 – Taxa de acertos do conjunto Glass

	C4.5	FuzzyDT
Taxa de acertos	141 (65.8879 %)	140 (65,4206 %)

C4.5

Valor Real \ Valor Predito	1	2	3	4	5	6	7
1	50	14	4	0	0	1	1
2	16	43	8	0	5	2	2
3	7	5	5	0	0	0	0
4	0	0	0	0	0	1	1
5	0	1	0	0	11	0	1
6	1	0	0	0	0	8	0
7	2	3	0	0	0	0	24

FuzzyDT

Valor Real \ Valor Predito	1	2	3	4	5	6	7
1	53	16	0	0	0	0	1
2	19	50	0	0	4	3	0
3	12	5	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	4	0	0	8	1	0
6	0	2	0	0	3	4	0
7	0	2	0	0	1	1	25

Figura 21 – Matriz de confusão do conjunto Glass

Fazendo a análise classe a classe a partir das matrizes de confuão na Figura 21, primeiramente para a classe 1, o FuzzyDT obteve um vantagem com 53 instâncias classificadas de forma correta, ou seja, 3 a mais do que o C4.5. Ele também teve sucesso com a classe 2, com 7 instâncias corretas a mais do que seu concorrente. Para a classe 3, o C4.5 apresenta melhor desempenho, isto visto que o FuzzyDT não teve nenhuma instância classificada nesta classe. Já para a classe 4, ambos os modelos não classificaram nenhuma instância corretamente. Na classe 5, o C4.5 obteve melhor desempenho com 3 instâncias classificadas adequadamente a mais do que o modelo *fuzzy*, assim como para a classe 6, porém com 4 instâncias a mais. E, por último, a classe 7, que teve melhor desempenho no modelo FuzzyDT, com apenas uma instâncias classificada a mais do que o C4.5. Assim, foram 3 classes com melhor desempenho no C4.5, 3 com melhor desempenho e uma classe com o mesmo desempenho.

Nesta base de dados, o modelo C4.5 apresentou 30 nós folhas e o tamanho da árvore foi de 59 nós, e no FuzzyDT, foram 29 nós folhas, ou seja, 1 regra a menos do que o modelo clássico, e 40 nós que resultam no tamanho da árvore.

5.4 HABERMAN'S SURVIVAL

O conjunto *Haberman's Survival* possui casos de um estudo feito entre os anos de 1958 a 1970 no Hospital Billings em Chicago sobre a sobrevivência de pacientes que realizaram uma cirurgia para o câncer de mama. Seus atributos são: idade do paciente no dia que a cirurgia foi

realizada, ano da cirurgia e número de nódulos axilares positivos detectados. E a saída é 1 ou 2, sendo 1 caso o paciente viveu mais que 5 anos e 2 caso o paciente morreu em 5 anos. Assim, há 3 atributos e 306 instâncias.

Tabela 6 – Taxa de acertos do conjunto Haberman's Survival

	C4.5	FuzzyDT
Taxa de acertos	220 (71.8954 %)	225 (73.5294 %)

Figura 22 – Matriz confusão do conjunto Haberman's Survival

A Tabela 6, permite observar que o modelo FuzzyDT, teve 5 instâncias classificadas a mais corretamente comparado ao C4.5, tendo a diferença dada em 1,634%. Um fato é que este conjunto especifico, foi o único a apresentar o modelo FuzzyDT como melhor.

A partir da Figura 22, observa-se que no modelo FuzzyDT nenhuma das instâncias foi classificada na classe B, ou seja, os casos em que o paciente morreu em 5 anos. Diferente do modelo C4.5, que classificou 24 instâncias corretas e 29 incorretas na classe B. Já em relação a classe A, o modelo C4.5 previu 196 instâncias corretas e 57 incorretas, enquanto o FuzzyDT previu 225 instâncias corretas e 81 incorretas, apresentando assim melhor desempenho para esta classe.

A árvore para esta base de dados, para o C4.5, apresenta 3 nós folhas e tamanho da árvore igual a 5. Enquanto o FuzzyDT apresentou um nó folha, ou seja, há apenas uma classe, isto porque este modelo não conseguiu classificar corretamente nenhuma instância de uma das classe do problema.

5.5 HEART DISEASE

Heart Disease refere-se a uma base de dados definida em 1988 e consiste em 4 bancos: Cleveland, Hungria, Suíça e Long Beach. Possui 76 atributos porém todos os experimento publicados referem-se ao uso de um subconjunto de 14 atributos apenas, sendo ele o de Creveland. O objeto do conjunto é identificar a presença de alguma doença cardíaca no paciente. Seus atributos são os seguintes: idade, sexo, tipo de dor no peito (4 tipos), pressão sanguínea em

repouso, colesterol sérico em mg / dl, açúcar no sangue em jejum > 120 mg / dl, resultados eletrocardiográficos de repouso (valores 0,1,2), frequência cardíaca máxima alcançada, angina induzida por exercício, depressão de ST induzida por exercício em relação ao repouso, a inclinação do segmento ST de pico do exercício e número de vasos principais (0-3) coloridos por fluorosopia. E o atributo de saída é 0 ou 1, sendo 0 normal e 1 é defeito corrigido. Logo, o conjunto apresenta 14 atributos e 303 instâncias.

Tabela 7 – Taxa de acertos do conjunto Heart Disease

	C4.5	FuzzyDT
Taxa de acertos	238 (78.5479 %)	232 (76.5677 %)

Figura 23 – Matriz confusão do conjunto Heart Disease

Para esta base de dados, nota-se na Tabela 7, que novamente o modelo de Quinlan, o C4.5, teve sucesso, onde 6 instâncias foram classificadas a mais corretamente comparadas ao modelo que envolves a lógica *Fuzzy*. Além disso, observando a Figura 23, é possível ver que ambos modelos classificaram um total de instâncias iguais para a classe A, que no caso é normal. Já para a classe B, ou seja, o defeito corrigido, o modelo FuzzyDT classificou 35 instâncias incorretas para a classe A, enquanto o C4.5 classificou 29, logo, obtendo melhor desempenho para esta classe.

A árvore deste conjunto foi gerada com 26 nós folha e tamanho da árvore igual a 51 nós, para o modelo C4.5, e de 24 nós folha e tamanho da árvore de 34 nós para o FuzzyDT. Logo, o fuzzyDT se mostrou melhor para os dois parâmetros, com 2 regras e 17 nós a menos.

5.6 WINE

O conjunto *Wine* (CORTEZ et al., 2009) possui o objetivo de classificar a qualidade do "Vinho Verde"português por meio de 11 atributos: acidez fixa, acidez volátil, ácido cítrico, açúcar residual, cloretos, dióxido de enxofre livre, dióxido de enxofre total, densidade, pH, sulfatos e álcool. Dessa forma, a qualidade do vinho, que é a saída, pode ser classificada entre 1 e 9. Este conjunto é o maior entre os que foram analisados, contendo 1599 instâncias no total.

Tabela 8 – Taxa de acertos do conjunto Wine

	C4.5	FuzzyDT
Taxa de acertos	978 (61.1632 %)	924 (57.7861 %)

C4.5

Valor Real \ Valor Predito	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	1	3	2	3	1	0	0
4	0	0	4	8	21	17	3	0	0
5	0	0	3	23	478	160	17	0	0
6	0	0	5	17	169	392	50	5	0
7	0	0	2	4	23	65	99	6	0
8	0	0	0	0	0	11	7	0	0
9	0	0	0	0	0	0	0	0	0

FuzzyDT

Valor Real \ Valor Predito	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	2	8	3	1	0	0
4	0	0	1	5	29	16	2	0	0
5	0	0	4	4	473	191	9	0	0
6	0	0	0	5	194	419	19	1	0
7	0	0	0	0	13	159	27	0	0
8	0	0	0	0	2	14	2	0	0
9	0	0	0	0	0	0	0	0	0

Figura 24 – Matriz confusão do conjunto Wine

Este é o maior banco de dados analisado, e as taxas apresentadas na Tabela 8, permitem analisar que este foi o conjunto com maior diferença entre os modelos, de 3,3771% a favor do modelo C4.5.

Analisando, agora, a Figura 24, observa-se que nenhum dos modelos classificou alguma instância nas classes 1, 2 e 9. Na classe 3, o C4.5 obteve sucesso com apenas uma instância e o FuzzyDT nenhuma foi classificada corretamente. Para a classe 4, o modelo de Quinlan também obteve melhor desempenho, com 3 instâncias a mais do que o FuzzyDT classificadas de forma correta. Em ambos, a classe com o maior número de instâncias classificadas corretamente foi a classe 5, com 478 no C4.5 e 473 para o FuzzyDT. A classe 6 teve o segundo maior número de instâncias classificadas adequadamente, porém nesta o FuzzyDT uma obteve uma vantagem de 27 instâncias sobre o C4.5, diferente do que ocorre na classe 7, onde o C4.5 teve melhor resultado, com 72 instâncias a mais classificadas de forma correta. E, por fim, na classe 8, nenhuma instância para ambos foi classificada apropriadamente.

O Wine apresentou a maior árvore entre os conjuntos apresentados, sendo que para o C4.5 o número de nós folha foi de 227, um total de 453 nós que resultam no tamanho da árvore. Para o FuzzyDT, os números são muito reduzidos, com 82 nós folha e com tamanho da árvore igual a 117. Assim, o FuzzyDT mostrou 145 regras a menos, a maior diferença observada dentre as bases de dados analisadas.

5.7 CONSIDERAÇÕES FINAIS

Nesta Seção foram descritos os conjuntos de dados utilizados para a avaliação e comparação dos modelos selecionados para estudo. A análise foi feita através dos valores de taxa de acerto e matrizes de confusão geradas para cada classificador.

No gráfico da Figura 25 pode ser observado a diferença das taxas de acerto para todas as bases de dados considerando os dois modelos de classificação estudados.

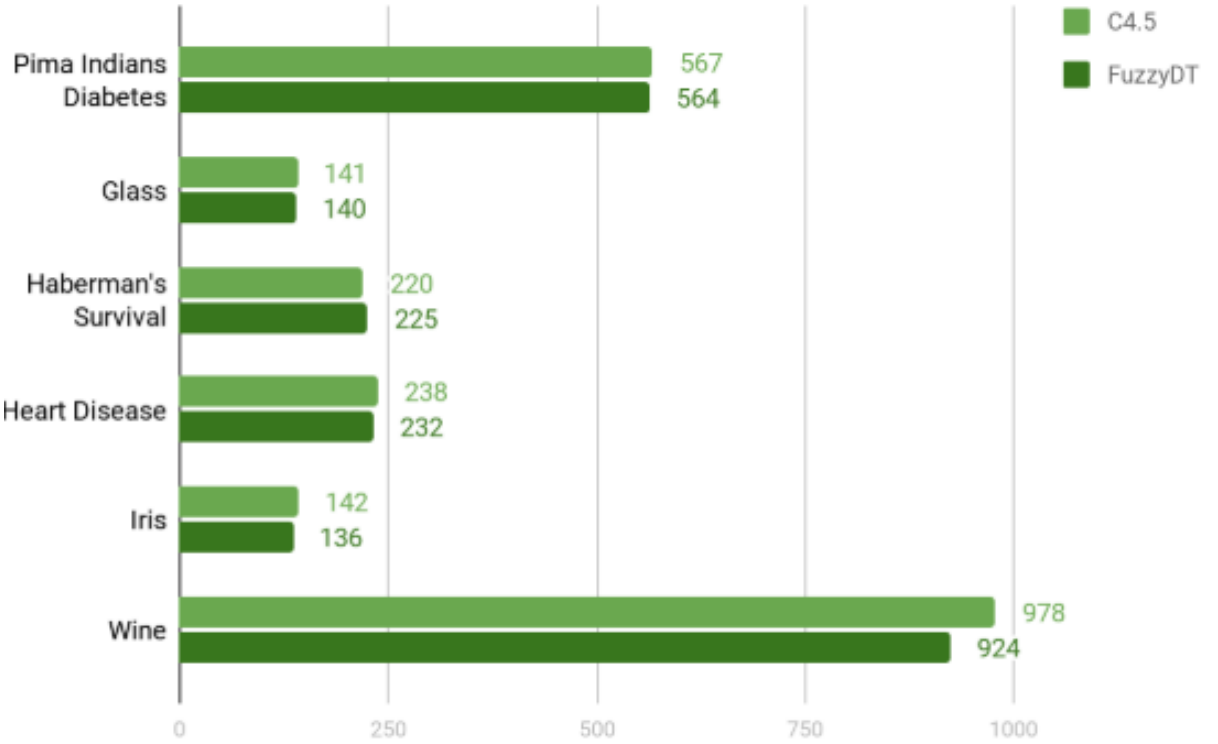

Figura 25 – Gráfico com a quantidade de acertos dos conjunto

A partir da Figura 25, considerando a taxa de acerto do classificador, observa-se que o modelo FuzzyDT de Cintra foi melhor apenas para o a base de dados *Haberman's Survival*. Além disso, é notório que os modelos obtiveram uma diferença mínima, sendo que a maior diferença vista foi a do *Wine*, com apenas 3% aproximadamente de diferença.

A Tabela 9 apresenta o número de classes que cada modelo obteve melhor desempenho para cada um dos conjuntos de dados apresentados.

Tabela 9 – Desempenho dos modelos baseado no número de classes classificadas corretamente

Banco de Dados	Total de Classes	Número de classes com melhor desempenho no C4.5	Número de classes com melhor desempenho no FuzzyDT	Número de classes com o mesmo número de instâncias classificadas corretas para ambos modelos
Pima Indian Diabetes	2	1	1	0
Glass	7	3	3	1
Haberman's Survival	2	1	1	0
Heart Disease	2	1	0	1
Iris	3	1	2	0
Wine	9	4	1	4

Dessa forma, da Tabela 9 observa-se que em dois banco de dados, os modelos obtiveram o mesmo número de classes com melhor desempenho, o Pima Indian Diabetes, o Haberman's Survival e o Glass , sendo que o último apresentou uma classe onde os dois classificaram o mesmo número de instâncias corretas. Já para os outros conjuntos, o C4.5 mostrou um maior número de classes no Heart Disease (com uma classe com o valor igual de número de instâncias corretas) e no Wine (com 4 classes com o valores iguais de números de instâncias corretas), e o FuzzyDT, mostrou vantagem apenas no banco Iris, com 2 classes apresentando melhor desempenho.

A Tabela 10 apresenta a diferença entre os dois modelos utilizando os parâmetros de número de nós folha e do tamanho das árvores geradas.

Tabela 10 – Comparação entre o número de folhas e do tamanho da árvore em cada modelo

	C4.5		FuzzyDT	
Banco de Dados	Número de nós folha	Tamanho da Árvore	Número de nós folha	Tamanho da Árvore
Pima Indian Diabetes	20	39	21	29
Glass	30	59	29	40
Haberman's Survival	3	5	1	1
Heart Disease	26	51	24	34
Iris	5	9	5	7
Wine	227	453	82	117

Assim, para uma análise final, observa-se na Tabela 10 que em todos as bases de dados, exceto para a base Pima Indians, o modelo fuzzyDT mostrou ser melhor em relação ao número de nós folha e no tamanho da árvore gerada, com valores menores, quando comparado ao C4.5. Este fato, mostra que o modelo FuzzyDT apresentou um número menor de regras, e que para algumas aplicações pode ter desempenho melhor que os modelos clássicos.

6 CONCLUSÃO

Dentre os modelos de Aprendizado de Máquina para a tarefa de classificação, as Árvores de Decisão (AD) são largamente utilizadas. Este fato se deve a sua característica de fácil interpretação e intuitividade, além de que ela pode ser expressa graficamente, possibilitando visualização adequada para análise pelo usuário do modelo. Dessa forma, variados algoritmos para sua representação são vistos na literatura, sendo os principais o ID3 (QUINLAN, 1986), o CART (BREIMAN, 2001) e o C4.5 (QUINLAN, 1996). Esta pesquisa realizou teste a partir do modelo C4.5 que faz uso do ganho de informação e do cálculo da entropia para a seleção de atributos, formando os ramos. Eles podem ser vistos como uma regra cujas condições são feitas pelos atributos do conjunto em análise. Isto é feito de forma recursiva até que uma classe seja atribuída ao nó folha da árvore, ou seja, o final.

As ADs são fortemente úteis quando se é trabalhada com valores tanto discretos como contínuos, porém quando o valor é incerto, ou impreciso, é necessário fazer uso de alguma teoria matemática que possa representar a imprecisão. Para tanto a lógica *Fuzzy* pode ser empregada. Este trabalho compara o modelo FuzzyDT (CINTRA, 2016), que transforma os valores contínuos, de entrada, em valores *fuzzy*, com um modelo clássico. As análises e comparações dos dois modelos de Árvores de Decisão, o C4.5 e o FuzzyDT, foram relizadas empregando seis bases de dados da UCI (DUA; GRAFF, 2017), sendo avaliados a partir da taxa de acerto, da matriz de confusão, além de comparar as árvores geradas em tamanho e número de nós folha. Todos os testes foram realizados na plataforma WEKA.

Os resultados obtidos dos experimentos, demonstrou que 5 dos 6 bancos de dados analisados, obtiveram maiores números de instâncias classificadas corretamente pelo modelo C4.5, sendo eles o Pima Indians Diabetes, Glass, Heart Disease, Iris e Wine, e o único que teve vantagem do FuzzyDT foi o Haberman's Survival. Apesar disso, um fato notável é que, as diferenças entre eles foram minímas, sendo que a maior diferença apresentada foi de aproximadamente 3%. Ressalta-se também que apenas um dos conjuntos obteve uma classificação correta de mais de 90% para ambos os modelos, que no caso foi o Iris, os outros obtiveram uma média de acerto de aproximadamente 70%. Já em relação a análise realizada classe a classe, para três bases de dados apresentaram o mesmo resultado de desempenho para ambos os modelos (Pima Indians Diabetes, Glass e Haberman's Survival), duas bases de dados apresentaram resultado favorável ao C4.5 (Heart Disease e Wine) e uma base de dados foi favorável ao FuzzyDT (Iris).

Ademais, destaca-se a análise realizada baseando-se no número de nós folha gerado, que indica o número de regras que podem ser extraídas de uma árvore de decisão, e de tamanho de árvore para cada modelo. O FuzzyDT mostrou apresentar um menor número de regras do que o C4.5 para todas as bases de dados analisadas, com exceção do Pima Indians Diabetes, em que C4.5 teve uma regra a menos. E o FuzzyDT obteve vantagem em todos os conjuntos no

tamanho da árvore, seus resultados sempre são muito menores quando comparado ao modelo clássico, o C4.5. Estes resultados podem sugerir a adequação da fuzzificação das variáveis para as bases selecionadas para estudo. Não há como afirmar que um modelo é melhor do que outro, mas indicar que os modelos podem ser melhores para classes de problemas similares. Coincidentemete, a maioria das árvores menores obtidas a partir do modelo *fuzzy* são menores e indicam uma vantagem deste modelo, se para estes conjuntos a aplicação pede interpretação de regras.

Dessa forma, observa-se a necessidade, de estudar em trabalhos futuros, diferentes modelos de AD, assim como também o estudo de diferentes algoritmos envolvendo a Lógica *Fuzzy*. Além disso, também envolver investigações em bases de dados com um maior número de atributos e instâncias, e analisar o número de regras que os modelos utilizantes da Lógica *Fuzzy* obtém comparado um modelo clássico de AD.

REFERÊNCIAS BIBLIOGRÁFICAS

ALPAYDIN, E. Introduction to machine learning. mit press. 2010. Citado na página 7.

BREIMAN, L. Random forests. *Machine Learning*, v. 45, 10 2001. Citado na página 37.

BREIMAN, L. et al. Classificação e regressão Árvores. 1984. Citado 2 vezes nas páginas 7 e 16.

CINTRA, M. E. Fuzzydt. 2016. Disponível em: <https://www.researchgate.net/publication/341069396_FuzzyDT>. Citado 3 vezes nas páginas 17, 22 e 37.

CINTRA, M. E.; CAMARGO, H. A. *Feature Subset Selection for Fuzzy Classification Methods*. [S.l.]: Information Processing and Management of Uncertainty in Knowledge-Based Systems - IPMU, 2010. v. 80. 918-327 p. Citado na página 21.

CORTEZ, P. et al. Modeling wine preferences by data mining from physicochemical properties. p. 547–553, 2009. Citado na página 33.

COX, E. A. The fuzzy systems handbook: a practioners guide to building, using and maintaining fuzzy systemsuma introdução ao estudo da lógica fuzzy. New York: Academic Press, 1994. Citado 2 vezes nas páginas 3 e 12.

DIETTERICH, T. Interacting meaningfully with machine learning systems: Three experiments. International Journal of Human-Computer Studies, p. 639–662, 2009. Citado na página 10.

DUA, D.; GRAFF, C. *UCI Machine Learning Repository*. 2017. Disponível em: <http://archive.ics.uci.edu/ml>. Citado 5 vezes nas páginas 8, 19, 20, 27 e 37.

FACELI, K. Inteligência artificial: Uma abordagem de aprendizado de máquina. 2011. Citado na página 14.

FERNANDES, A. M. da R. Inteligência artificial: noções gerais. Visual Books, 2003. Citado na página 10.

FERREIRA, R. et al. Container crane controller with the use of a neurofuzzy network. p. 122–129, 2016. Citado 2 vezes nas páginas 20 e 26.

FISHER, R. A. The use of multiple measurements in taxonomic problems. Annals Eugen, 1936. Citado na página 27.

GROOVER, M.; NAGEL, R. N.; ODREY, N. G. Robótica: Tecnologia e programação. 1989. Citado na página 10.

JANé, D. D. A. Uma introdução ao estudo da lógica fuzzy. Revista de Humanidades e Ciências Sociais Aplicadas, p. 1–16, 2004. Citado 2 vezes nas páginas 3 e 13.

KLIR, G. J.; YUAN, B. Fuzzy sets and fuzzy logic - theory and applications. *Prentice Hall*, 1995. Citado na página 13.

KOTHARI, R.; DONG, M. Decision trees for classification: A review and some new results. p. 241–252, 2001. Citado na página 14.

MCCARTHY, J. Generality in artificial intelligence. p. 226–236, 1990. Citado na página 10.

MITCHELL, T. Machine learning. 1997. Citado na página 7.

PEDRYCZ, W.; GOMIDE, F. *Sistema de Apoio à Decisão para avaliação técnica de jogadores de futebol: impAn Introduction to Fuzzy Sets*. [S.l.]: MIT Press, 1998. Citado 2 vezes nas páginas 12 e 13.

PINHO, L. Sistema de apoio à decisão para avaliação técnica de jogadores de futebol: implementação de ferramenta de etl e modelagem conceitual baseada em lógica fuzzy. 2016. Citado 2 vezes nas páginas 3 e 12.

QUINLAN, J. R. Induction of decision trees. v. 1, p. 81–106, 1986. Citado 4 vezes nas páginas 7, 14, 16 e 37.

QUINLAN, J. R. C4.5: Programs for machine learning. 1992. Citado 3 vezes nas páginas 7, 11 e 16.

QUINLAN, J. R. Bagging, boosting e c4.5. p. 725–730, 1996. Citado 4 vezes nas páginas 7, 16, 17 e 37.

RIBEIRO, M. V.; CAMARGO, H. A.; CINTRA, M. E. A comparative analysis of pruning strategies for fuzzy decision trees. In: *2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)*. [S.l.: s.n.], 2013. p. 709–714. Citado 2 vezes nas páginas 7 e 18.

RUSSELL, S.; NORVIG., P. Artificial intelligence: A modern approach. 1995. Citado 2 vezes nas páginas 7 e 11.

UNIVERSITY OF WAIKATO. Weka 3 – machine learning software in java. 2010. Disponível em: <http://www.cs.waikato.ac.nz/ml/weka>. Citado na página 19.

WANG, L. X.; MENDEL, J. M. Generating fuzzy rules by learning from examples. p. 141–1427, 1992. Citado na página 22.

WITTEN, I. H.; FRANK, E.; HALL, M. A. Data mining: Practical machine learning tools and techniques. 1999. Citado 3 vezes nas páginas 7, 14 e 15.

ZADEH, L. A. Fuzzy sets. *Information and control*, Elsevier, v. 8, n. 3, p. 338–353, 1965. Citado 3 vezes nas páginas 7, 11 e 12.

Isabela Corsi

Isabela Nunes Drummond.