



**UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

ALESSANDRO JORGE RODRIGUES DA SILVA

**UMA DISCUSSÃO ENTRE AS TÉCNICAS DE CLASSIFICAÇÃO:
NAIVE BAYES E ÁRVORE DE DECISÃO**

São Luís
2016

ALESSANDRO JORGE RODRIGUES DA SILVA

**UMA DISCUSSÃO ENTRE AS TÉCNICAS DE CLASSIFICAÇÃO:
NAIVE BAYES E ÁRVORE DE DECISÃO**

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Ivo José da Cunha Serra

São Luís

2016

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Núcleo Integrado de Bibliotecas/UFMA

da Silva, Alessandro Jorge Rodrigues.

UMA DISCUSSÃO ENTRE AS TÉCNICAS DE CLASSIFICAÇÃO: NAIVE BAYES E ÁRVORE DE DECISÃO / Alessandro Jorge Rodrigues da Silva. - 2016.

52 p.

Orientador(a): Ivo José da Cunha Serra.

Monografia (Graduação) - Curso de Ciência da Computação, Universidade Federal do Maranhão, São Luis, 2016.

1. Árvore de Decisão. 2. Mineração de Dados. 3. Naive Bayes. I. Serra, Ivo José da Cunha. II. Título.

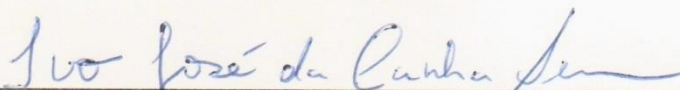
ALESSANDRO JORGE RODRIGUES DA SILVA

**UMA DISCUSSÃO ENTRE AS TÉCNICAS DE CLASSIFICAÇÃO:
NAIVE BAYES E ÁRVORE DE DECISÃO**

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

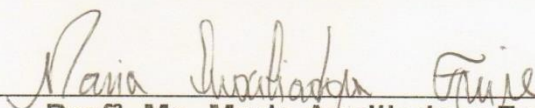
Aprovada em: 14/09/2016

BANCA EXAMINADORA



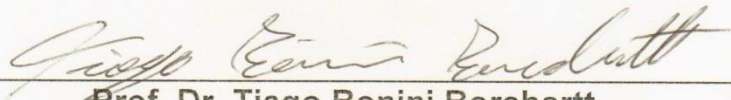
Prof. Dr. Ivo José da Cunha Serra (Orientador)

Doutor em Ciência da Computação
Universidade Federal do Maranhão



Profª. Ma. Maria Auxiliadora Freire

Mestre em Ciência de Engenharia
Universidade Federal do Maranhão



Prof. Dr. Tiago Bonini Borchardt

Doutor em Ciência da Computação
Universidade Federal do Maranhão

A minha família e em especial a Deus e
aos meus pais que me acompanharam
durante toda a minha caminhada.

AGRADECIMENTOS

Ao professor Ivo Serra que gentilmente aceitou ser meu orientador, disponibilizando seu tempo e demonstrando sempre muita paciência e sabedoria na condução deste trabalho.

A minha namorada Keila Sousa sempre muito prestativa, paciente, com suas palavras de carinho e incentivo, me dando forças nos momentos difíceis.

A todos os professores que se esforçaram para me passar um pouco de seu conhecimento durante a graduação e que depois da aula ainda disponibilizavam um pouco de seu tempo para tirar algumas dúvidas.

Aos amigos de curso que estiveram comigo durante toda essa caminhada até aqui e que de uma maneira ou de outra colaboraram para que esse momento fosse possível. Amigos esses que dividimos conhecimento, preocupações e ansiedades devido às dificuldades que foram muitas.

“A imaginação é mais importante que a ciência, porque a ciência é limitada, ao passo que a imaginação abrange o mundo inteiro.”

(Albert Einstein)

RESUMO

Com o avanço da tecnologia da informação tornou-se possível armazenar grandes volumes de dados. A KDD (*Knowledge Discovery in Databases*) surgiu nesse cenário como uma alternativa para extrair informações úteis dessas bases de dados. A mineração de dados é uma das etapas da KDD na qual ocorre a busca efetiva por conhecimentos novos e úteis a partir dos dados. Dentre as tarefas da área de mineração de dados, a classificação é uma das tarefas mais importantes. Este trabalho analisa duas das mais importantes técnicas de classificação que são a Árvore de Decisão e *Naive Bayes*. Uma discussão foi realizada levando em consideração a forma como cada classificador trata os atributos no momento da construção do modelo de classificação. Nessa fase a entropia e o ganho de informação foram utilizados para auxiliar na escolha dos atributos considerados mais importantes. Os atributos redundantes, que repetem informações de outros atributos, e os atributos irrelevantes, que agregam pouca ou quase nenhuma informação para a tarefa de classificação, também fazem parte da discussão. A pesquisa foi baseada em literaturas renomadas na área da mineração de dados das quais podemos citar: Goldschmidt e Passos (2005), Russel e Norvig (2004), Tan, Steinbach e Kumar (2009), Carvalho (2002), Ham e Kamber (2006), Garcia (2000) e Gama (2000). A pesquisa permitiu observar que cada técnica apresenta vantagens e limitações, considerando o tipo de dados em que são aplicadas. A Árvore de Decisão tende a obter melhores resultados quando na base de dados não estão presentes os atributos irrelevantes. Já o *Naive Bayes* obtém resultados mais desejáveis na ausência de atributos redundantes.

Palavras-chave: Mineração de dados, *Naive Bayes*, Árvore de Decisão.

ABSTRACT

With the advancement of information technology has made it possible to store large volumes of data. The KDD (Knowledge Discovery in Databases) emerged in this scenario as an alternative to extract useful information from these databases. Data mining is one of the stages of KDD in which there is an effective search for new and useful knowledge from the data. Among the tasks of data mining area, the classification is one of the most important tasks. This paper examines two of the most important classification techniques that are Decision Tree and Naive Bayes. A discussion was conducted taking into account how each classifier is the attributes at the time of construction of the classification model. In this phase the entropy and information gain were used to assist in choosing the attributes considered most important. The redundant attributes, which repeat information from other attributes, and irrelevant attributes that add little or no information for the classification task, are also part of the discussion. The research was based on literature renowned in the field of data mining which include: Goldschmidt and Passos (2005), Russell and Norvig (2004), Tan, Steinbach and Kumar (2009), Carvalho (2002), Ham and Kamber (2006), Garcia (2000) and Gama (2000). The research allowed to observe that each technique has advantages and limitations, considering the type of data they are applied. Decision tree tends to get better results when the database are not present irrelevant attributes. But the Naive Bayes get more desirable results in the absence of redundant attributes.

Keywords: Data mining, Naive Bayes, Decision Tree.

LISTA DE FIGURAS

| | |
|---|----|
| Figura 2.1 – Classificação como a tarefa de mapear um conjunto de atributos x no seu rótulo de classe y | 18 |
| Figura 3.1 – Árvore de decisão induzida do conjunto de dados da tabela 2.1 | 24 |
| Figura 3.2 – Taxas de erros de testes e de treinamento. | 28 |
| Figura 4.1 – Divisão dos exemplos por meio de testes em atributos. | 42 |
| Figura 4.2 – Árvore de decisão induzida do conjunto de dados da tabela 4.1. | 44 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 2.1 – Informações sobre condições climáticas | 18 |
| Tabela 2.2 – Matriz de Confusão para um problema de duas classes..... | 20 |
| Tabela 2.3 - Exemplo de matriz de confusão para o modelo M | 21 |
| Tabela 3.1 – Informações sobre condições do tempo..... | 37 |
| Tabela 4.1 – Conjunto de dados weather..... | 40 |
| Tabela 4.2 – Probabilidade condicional para os atributos da tabela 4.1 | 43 |
| Tabela 4.3 – Características da Árvore de Decisão e <i>Naive Bayes</i> | 46 |

LISTA DE SIGLAS

| | |
|-------|--|
| CART | Classification and Regression Trees |
| CHAID | Chi Square Automatic Interaction Detection |
| ID3 | Iterative Dichotomiser 3 |
| KDD | Knowledge Discovery in Databases |
| KNN | K-Nearest Neighbors |
| NFL | No Free Lunch Theorem |
| UFMA | Universidade Federal do Maranhão |
| WEKA | Waikato Environment for Knowledge Analysis |

SUMÁRIO

| | |
|--|-----------|
| 1 INTRODUÇÃO | 14 |
| 1.1 Motivação..... | 15 |
| 1.2 Objetivos do trabalho..... | 15 |
| 1.3 Organização do trabalho | 16 |
| 2 CLASSIFICAÇÃO | 17 |
| 2.1 Definição | 17 |
| 2.2 Métricas para avaliação de modelos de classificação | 19 |
| 2.2.1 Acurácia | 19 |
| 2.2.2 Taxa de erro | 20 |
| 2.3 Matriz de confusão | 20 |
| 3 TÉCNICAS DE CLASSIFICAÇÃO..... | 22 |
| 3.1 Árvores de decisão | 22 |
| 3.1.1 Definição | 22 |
| 3.1.2 Funcionamento de uma árvore de decisão | 23 |
| 3.1.3 Indução de uma árvore de decisão | 24 |
| 3.1.4 Métricas para escolha de atributos..... | 25 |
| 3.1.4.1 Entropia | 26 |
| 3.1.4.2 Ganho de informação | 26 |
| 3.1.5 <i>Overfitting e underfitting</i> de modelos de classificação..... | 27 |
| 3.1.6 Algoritmos de Indução de árvores de decisão..... | 28 |
| 3.1.6.1 ID3..... | 29 |
| 3.1.6.2 C4.5..... | 30 |
| 3.1.7 Características de indução de Árvore de Decisão:..... | 31 |
| 3.2 Naive bayes..... | 32 |
| 3.2.1 Definição | 32 |
| 3.2.2 Probabilidade incondicional..... | 33 |
| 3.2.3 Probabilidade condicional..... | 33 |
| 3.2.4 Independência condicional | 33 |
| 3.2.5 Teorema de bayes | 34 |
| 3.2.6 Usando o teorema de bayes para a classificação | 35 |

| | |
|--|-----------|
| 3.2.5 Como funciona um classificador de <i>Naive Bayes</i> | 36 |
| 3.2.6 Características de classificadores <i>Naive Bayes</i> | 36 |
| 4 UMA DISCUSSÃO ENTRE OS CLASSIFICADORES: NAIVE BAYES E ÁRVORE DE DECISÃO. | 39 |
| 4.1 Atributos mais promissores do conjunto de treinamento..... | 39 |
| 4.2 Atributos irrelevantes e redundantes presentes no conjunto de treinamento | 44 |
| 5. CONCLUSÃO | 47 |
| REFERÊNCIAS..... | 50 |

1 INTRODUÇÃO

O avanço da tecnologia possibilitou o surgimento de hardwares com capacidade cada vez maior de armazenamento e com baixo custo de aquisição, o que possibilitou o acúmulo de dados nas grandes corporações que tinham origem tanto nas operações do dia a dia, na internet e em outras fontes de dados, formando grandes bancos de dados. Pouca ou quase nenhuma informação relevante era identificada em todo esse emaranhado de dados, pois técnicas tradicionais de análise de dados eram utilizadas sem muito êxito para esse propósito. As informações estavam presentes, mas ao mesmo tempo estavam “escondidas” o que foi relatado por Ham e Kamber (2006), ricos em dados, mas pobre de informações.

A quantidade de informações disponíveis ultrapassou a capacidade humana de compreensão. Não é viável, sem o auxílio de ferramentas computacionais apropriadas, a análise de grandes quantidades de dados pelo homem. Portanto, torna-se imprescindível o desenvolvimento de ferramentas que auxiliem o homem, de forma automática e inteligente, na tarefa de analisar, interpretar e relacionar esses dados para que se possa desenvolver e selecionar estratégias de ação em cada contexto de aplicação (GOLDSCHMIDT; PASSOS, 2005).

É nesse cenário rico em dados, mas que estavam sendo mal aproveitados pelas grandes corporações, que novos processos de classificação e exploração de dados foram desenvolvidos, tanto para processamento automático quanto orientado manualmente por um operador. Esse conjunto de técnicas é denominada Descoberta de Conhecimento em Bases de Dados (Knowledge Discovery in Databases - KDD), que se refere ao processo de descobrir informações úteis em grandes bases de dados com o intuito de auxiliar nas tomadas de decisões de forma inteligente. Segundo Fayyad et al. (1996), a KDD é uma tentativa de solucionar o problema causado pela chamada "era da informação": a sobrecarga de dados.

A expressão Mineração de Dados é mais conhecida, mas é na verdade, uma das etapas da Descoberta de Conhecimento em Bases de Dados. Afirma também que a Mineração de Dados é a principal etapa do processo de KDD, é nessa etapa que ocorre a busca efetiva por conhecimentos novos e úteis a partir dos dados (GOLDSCHMIDT; PASSOS, 2005). A Mineração de Dados é composta por muitas

tarefas, a classificação é uma delas, cujo objetivo é encontrar uma função que possa atribuir rótulos de classe a um determinado exemplo.

As técnicas de classificação utilizam um conjunto de dados de treinamento para construir um modelo de classificação que define um mapeamento de instâncias para classes. No momento da classificação, este modelo é utilizado para predizer a qual classe pertence uma nova instância. Entre as técnicas de classificação mais conhecidas estão a Árvore de Decisão e *Naive Bayes*.

Este trabalho propõe-se a fazer uma discussão envolvendo as técnicas de classificação Árvore de Decisão e *Naive Bayes* tomando como base para essa discussão a importância que cada atributo tem para o algoritmo de aprendizagem, as métricas entropia e ganho de informação vão orientar nessa tarefa e também o comportamento apresentado por cada classificador quando no conjunto de dados estiverem presentes atributos do tipo irrelevantes e atributos redundantes.

1.1 Motivação

A tarefa de classificar, ou seja, de definir objetos em categorias pré-definidas desperta o interesse de pesquisadores das mais diversas áreas de negócios como a Medicina, Ciências, Marketing, Economia, Astronomia, Geologia. Para as pessoas envolvidas nesses negócios a classificação está no centro das atenções uma vez que a mesma ajuda na tomada de decisões mais racionais a médio e em longo prazo.

Dito isso, a motivação para realizar este trabalho nasceu da necessidade de promover uma discussão sobre dois dos mais usuais métodos de classificação que são o *Naive Bayes* e Árvore de Decisão. Apresentar mais uma discussão como outras já existentes, só que sem o uso de software que auxiliassem nessa tarefa como o Weka, uma vez que o texto é mais uma discussão que gira em torno do comportamento dos dois classificadores no trato dos atributos no momento da concepção do modelo de conhecimento e também de como se “comportam” na presença de atributos irrelevantes e atributos redundantes.

1.2 Objetivos do trabalho

O objetivo principal deste trabalho é realizar uma discussão entre as técnicas de classificação Árvore de Decisão e *Naive Bayes* levando em consideração a

escolha dos atributos mais promissores do conjunto de dados, os atributos redundantes e irrelevantes.

Os objetivos específicos do trabalho são:

- a) Avaliar a importância das métricas para avaliar modelos de classificação como a Acurácia, Taxa de Erro e a Matriz de Confusão.
- b) Investigar a importância da Entropia e do Ganho de Informação, para a escolha dos atributos mais promissores para dividir o conjunto de treinamento.
- c) Analisar as consequências do problema de *Overfitting* e *Underfitting* sobre o poder de generalização dos modelos de classificação.

1.3 Organização do trabalho

Com o propósito de organizar o trabalho de forma a facilitar o entendimento do tema abordado, o conteúdo presente foi dividido nos seguintes tópicos:

O Capítulo 2 descreve o que é classificação no contexto de mineração de dados. Apresenta também métodos para avaliar um modelo de classificação: acurácia e taxa de erros e ainda a matriz de confusão que tabula o número de erros e acertos em suas linha e colunas.

O capítulo 3 fornece a base para o entendimento dos classificadores de Árvore de Decisão e *Naive Bayes*. Introduce outras questões importantes para a classificação como: as métricas para escolher a melhor forma de dividir o conjunto de treinamento que é a entropia e o ganho de informação; *overfitting* e *underfitting* de modelos de classificação; os algoritmos de indução de Árvore de Decisão ID3 e C4.5 e as características de um classificador de Árvore de Decisão. Continua com a descrição de conceitos importantes para um bom entendimento do classificador *Naive Bayes* como: probabilidade condicional e incondicional, independência condicional e o importante teorema de *Bayes*.

O capítulo 4 apresenta uma discussão entre os classificadores *Naive Bayes* e Árvore de Decisão, tomando como base a seleção do melhor atributo para dividir o conjunto de treinamento, os atributos irrelevantes e redundantes.

E por fim, no capítulo 5 serão apresentadas algumas conclusões obtidas da discussão realizada no capítulo 4, além de sugestões para a realização de trabalhos futuros.

2 CLASSIFICAÇÃO

O ser humano está sempre classificando o que percebe a sua volta, como exemplo: criando classes de relações humanas diferentes e dando a cada classe uma forma diferente de tratamento; formando classes de comportamento em diferentes ambientes; definindo classes sociais; estabelecendo preconceitos e tratando as pessoas segundo estes estereótipos, entre outras formas de classificação (CARVALHO, 2002).

Entre as inúmeras aplicações que fazem com que a tarefa de classificação tenha um papel importante na vida das pessoas, podemos citar: detecção de fraudes, detecção de mensagens de spam em e-mails, a categorização de células como malignas ou benignas, análise de créditos, análise de riscos em seguros, diagnósticos precisos de doenças e prescrição de tratamento, análise de defeitos em equipamentos, classificação de galáxias, identificação de perfis para determinados produtos, identificar segmentos de mercado e identificar padrões de rotatividade.

Este capítulo abordará sobre a tarefa de classificação no contexto da mineração de dados. A seção 2.1 descreve os propósitos de um modelo de classificação; a seção 2.2 discorre sobre as métricas para avaliar modelos de classificação e a seção 2.3 descreve a matriz de confusão.

2.1 Definição

De acordo com Goldschmidt e Passos, classificação consiste basicamente em descobrir uma função que seja capaz de mapear um conjunto de registros em um conjunto de rótulos categóricos predefinidos, que são as classes. Assim que descoberta, tal função pode ser aplicada a novos registros para prever a classe em que tais registros se enquadram.

É a tarefa que se ocupa em organizar objetos em uma entre diversas categorias pré-definidas, é um problema universal que envolve diversas aplicações (TAN; STEINBACH; KUMAR, 2009).

Ainda segundo Tan, Steinbach e Kumar (2009), classificação é a tarefa de assimilar uma função alvo f que mapeie cada conjunto de atributos x para um dos rótulos de classes y pré-definidos, onde essa função é também conhecida informalmente como modelo de classificação e que é útil para os seguintes propósitos:

Modelagem Descritiva: O modelo, nesse caso, serve como ferramenta explicativa para distinguir objetos e classes diferentes. Por exemplo, seria útil aos jogadores de golfe e as pessoas envolvida em algum evento dessa natureza, ter em mãos um modelo descritivo que resuma os dados da tabela 2.1, de tal forma que ficasse claro quais características definem um dia como bom ou ruim para se jogar golfe.

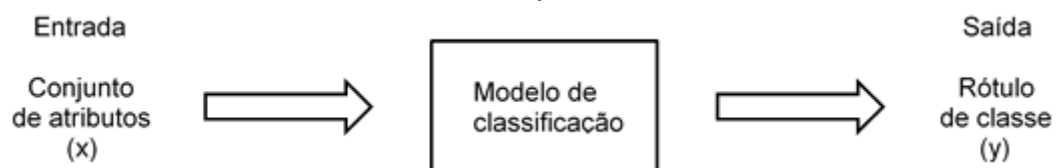
Tabela 2.1 – Informações sobre condições climáticas

| Atributos | | | | |
|------------------|-------------------------|--------------------|--------------|--------------------|
| Aparência | Temperatura (°F) | Umidade (%) | Vento | Jogar Golfe |
| Ensolarada | 75 | 70 | Sim | Sim |
| Ensolarada | 80 | 90 | Sim | Não |
| Ensolarada | 85 | 85 | Não | Não |
| Ensolarada | 72 | 95 | Não | Não |
| Ensolarada | 69 | 70 | Não | Sim |
| Nublada | 72 | 90 | Sim | Sim |
| Nublada | 83 | 78 | Não | Sim |
| Nublada | 64 | 65 | Sim | Sim |
| Nublada | 81 | 75 | Não | Sim |
| Chuvosa | 71 | 80 | Sim | Não |
| Chuvosa | 65 | 70 | Sim | Não |
| Chuvosa | 75 | 80 | Não | Sim |
| Chuvosa | 68 | 80 | Não | Sim |
| Chuvosa | 70 | 96 | Não | Sim |

Fonte: Goldschmidt e Passos (2005).

Modelagem Preditiva: O modelo de classificação também pode ser utilizado para prever o rótulo de classe de registros não conhecidos. A figura 2.1 descreve o comportamento de um modelo de classificação que ao receber um conjunto de atributos de um registro desconhecido como entrada gera como saída o rótulo de classe.

Figura 2.1 – Classificação como a tarefa de mapear um conjunto de atributos x no seu rótulo de classe y .



Fonte: Tan, Steinbach e Kumar (2009).

2.2 Métricas para avaliação de modelos de classificação

Esta seção descreve as métricas para avaliar modelos de classificação baseado no número de erros e acertos cometido pelo modelo em um conjunto de dados.

2.2.1 Acurácia

Conhecida como taxa de acerto ou precisão, é uma das principais métricas utilizadas para avaliar modelos em problemas de classificação. Denotada pela equação 2.1:

$$Acc (\%) = \frac{|S_c|}{|S|} \times 100 \quad (2.1)$$

Onde $|S_c|$ é a quantidade de registros classificados corretamente em um conjunto qualquer S e $|S|$ é quantidade de registros no conjunto S.

Para ilustrar o conceito de acurácia considere a tabela 2.1 com informações das condições climáticas. A partir desse conjunto de dados podemos gerar um modelo de classificação M que classificou corretamente doze das quatorze instâncias desse conjunto, portanto a sua acurácia é calculada por meio da equação 2.1:

$$Acc (\%) = \frac{12}{14} \times 100 \cong 85,71\%$$

A acurácia calculada a partir do conjunto de dados utilizado para gerar o classificador é denominada acurácia de treinamento, denotada por: Acc_{trein} .

Agora vamos supor que queiramos testar o desempenho do modelo M em um conjunto de dados também com quatorze instâncias, mas que não é igual aos dados da tabela 2.1.

Suponha, também, que o modelo M tenha classificado corretamente 10 das quatorze instâncias, logo a sua acurácia será:

$$Acc (\%) = \frac{10}{14} \times 100 \cong 71,42\%$$

A acurácia calculada a partir de um conjunto de testes é denominada de acurácia de teste, denotada por: Acc_{teste} .

Perceba que o classificador teve um desempenho razoável no conjunto de testes. Em geral, alta acurácia de treinamento não implica, necessariamente, que o

modelo é bom, apenas significa que ele é um espelho dos dados (síntese). Por outro lado, baixa acurácia de treinamento não implica que o modelo é ruim. Afirmações quanto a classificar um modelo como bom ou ruim só podem ser feitas a partir da sua acurácia de teste que é uma boa estimativa para acurácia de execução (taxa de previsão). Em suma, um bom modelo de classificação deve ter altas taxas de acurácia.

2.2.2 Taxa de erro

Outra medida muito conhecida, também denominada de taxa de classificação incorreta, é a taxa de erro. Esta métrica é complementar a acurácia podendo, portanto, ser expressa em termos dela e vice-versa. Comumente denotada pela equação 2.2:

$$Err (\%) = \frac{|S_e|}{|S|} \times 100 \quad (2.2)$$

Onde $|S_e|$ é a quantidade de registros classificados incorretamente em um conjunto qualquer S e $|S|$ é quantidade de registros no conjunto S.

Como a acurácia e a taxa de erro são medidas complementares, isso implica dizer matematicamente que:

$$Acc = 1 - Err \quad (2.3)$$

2.3 Matriz de confusão

A avaliação de desempenho de um modelo de classificação é realizada a partir das contagens de registros de testes previstos correta e incorretamente pelo modelo. Estas contagens são registradas em uma tabela conhecida como matriz de confusão (TAN; STEINBACH; KUMAR, 2009).

Tabela 2.2 – Matriz de Confusão para um problema de duas classes

| | | Classe prevista | |
|-------------|----------|-----------------|----------|
| | | Classe=1 | Classe=0 |
| Classe Real | Classe=1 | f_{11} | f_{10} |
| | Classe=0 | f_{01} | f_{00} |

Fonte: Tan; Steinbach e Kumar (2009).

A tabela 2.2 mostra uma matriz de confusão para um problema com duas classes, portanto com duas dimensões, uma que se refere às classes verdadeiras (a classe que de fato os registros pertencem) e a outra referente às classes preditas (a classe indicada pelo modelo de classificação). Para ilustrar, a tabela 2.3 demonstra o desempenho do modelo M no conjunto de teste. Perceba que o modelo classificou corretamente dez registros (somatório dos elementos da diagonal principal) e classificou quatro registros de maneira incorreta (diagonal secundária).

Tabela 2.3 - Exemplo de matriz de confusão para o modelo M

| Classes | Jogar = Sim | Jogar = Não |
|-------------|-------------|-------------|
| Jogar = Sim | 6 | 3 |
| Jogar = Não | 1 | 4 |

Uma visão geral da matriz de confusão de um classificador possui as seguintes características:

- O número de acertos, para cada classe, se localiza na diagonal principal $M(f_{11}, f_{00})$ da matriz;
- Os demais elementos $M(f_{10}, f_{01})$, para $i \neq j$, representam erros na classificação;
- A matriz de confusão de um classificador ideal possui todos os elementos da diagonal secundária iguais a zero, já que não comete erros.

Apesar de a matriz de confusão fornecer as informações necessárias para determinar o quão bem um modelo de classificação é executado, mais apropriado do que resumir estas informações com um único número, seria comparar o desempenho de modelos diferentes. Isto pode ser feito usando a métrica de desempenho conhecida como precisão (TAN; STEINBACH; KUMAR, 2009).

3 TÉCNICAS DE CLASSIFICAÇÃO

Uma técnica de classificação é uma abordagem sistemática que consiste na construção de modelos de classificação a partir de um conjunto de dados de entrada. Cada técnica emprega um algoritmo de aprendizagem para identificar um modelo que melhor se ajuste ao relacionamento entre o conjunto de atributos e a categoria rotulada dos dados de entrada, com o objetivo de prever corretamente os rótulos da categoria de novos exemplos (TAN; STEINBACH; KUMAR, 2009).

Este capítulo irá abordar os principais conceitos sobre duas das mais conhecidas técnicas de classificação e que são objetos de estudo desse trabalho: *Árvore de Decisão* e *Naive Bayes*. Será conduzida uma discussão mais detalhada desses dois classificadores no capítulo 4.

3.1 Árvores de decisão

Esta seção descreve uma técnica de classificação conhecida como *Árvore de Decisão*, em seguida abordará conceitos importantes como métricas para selecionar a melhor divisão do conjunto de treinamento e por fim os algoritmos de indução de *Árvore de Decisão* o ID3 e C4. 5.

3.1.1 Definição

Uma *Árvore de Decisão* é um modelo de conhecimento em que cada nó interno da árvore representa uma decisão sobre um atributo que determina como os dados estão particionados pelos seus nós filhos. No início, todos os exemplos com todas as classes misturadas, fazem parte da raiz da árvore. Denominado ponto de separação, um predicado é escolhido como sendo a condição que melhor divide ou discrimina as classes. Este predicado é um dos atributos do problema e divide a base dados em dois ou mais conjuntos, que são associados cada um a um nó filho. Cada novo nó abrange, uma partição da base de dados que é recursivamente separada até que o conjunto associado a cada nó folha tenha somente registros, ou a maioria de registros de uma mesma classe (GOLDSCHMIDT; PASSOS, 2005).

As *Árvores de Decisão* são representações simples do conhecimento e um meio eficiente de construir classificadores que predizem classes baseadas nos valores de atributos de um conjunto de dados (GARCIA, 2000). A *Árvore de Decisão* utiliza a estratégia de dividir-para-conquistar, um problema complexo é dividido em

subproblemas mais simples. A mesma estratégia é aplicada recursivamente a cada um dos subproblemas (GAMA, 2000).

Entre as várias técnicas de classificação as Árvores de Decisão são as únicas que representam os seus resultados de maneira hierárquica, ou seja, nodos em níveis superiores representam atributos de maior importância. Em contrapartida, nodos inferiores representam atributos menos importantes. Assim, a raiz da árvore representa o atributo de maior importância para o problema em questão.

3.1.2 Funcionamento de uma árvore de decisão

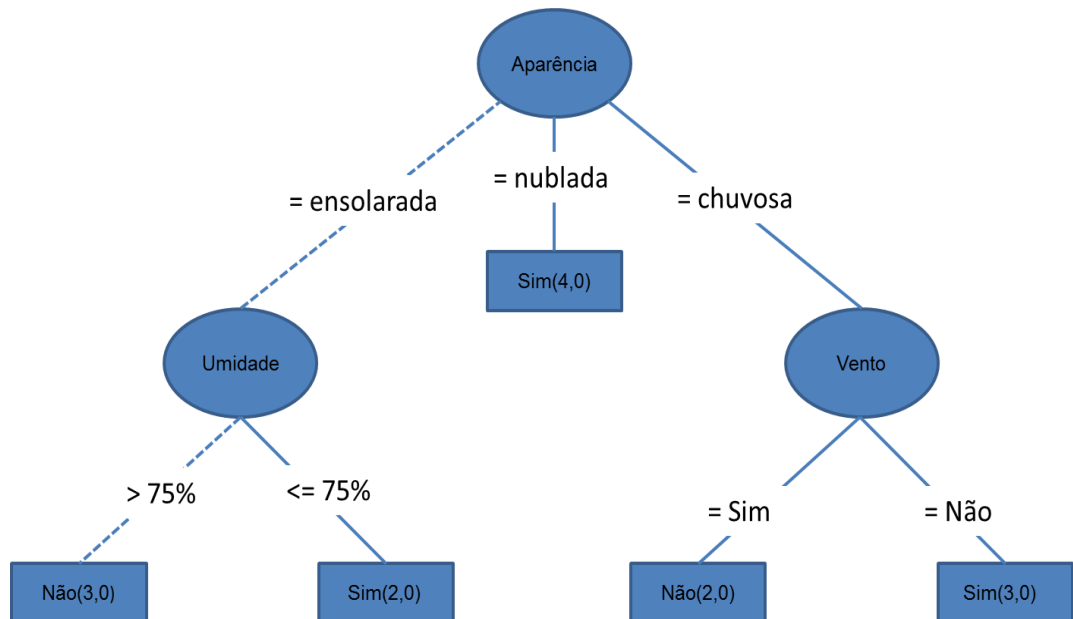
Para exemplificar como a classificação com Árvore de Decisão funciona, considere um problema simples para classificar os dias, conforme eles são satisfatórios ou não, para jogar golfe. Os registros são formados pelos seguintes atributos: Aparência, Temperatura, Umidade, Vento e o atributo classe Jogar Golfe. Os dias serão classificados em duas classes: Jogar Golfe= Sim e Jogar Golfe = Não. Os dados estão expostos na Tabela 2.1.

De posse da instância (Aparência = Ensolarada, Temperatura = 80° F, Umidade = 90%, Vento = sim) como concluir que é um bom dia para jogar golfe? Uma forma de fazer isso é realizar uma série de questões (testes de atributos) sobre as características de cada dia. A primeira questão é qual é a aparência do dia. Se a resposta for nublada, definitivamente é um bom dia para jogar golfe. No caso da instância acima a resposta para o primeiro questionamento é ensolarada. O próximo passo é saber qual é a umidade do dia, obtemos como resposta 90% aonde se chegou à classe Não, ou seja, não é um bom dia para jogar golfe.

A classificação da instância dada como exemplo, foi possível após uma série de questões cuidadosamente organizadas sobre os atributos do registro. A cada resposta alcançada, uma nova pergunta é realizada até alcançar o nó folha que indica a classe do registro.

De maneira bem direta, a Árvore de Decisão classifica novas instâncias partindo do nodo raiz (Aparência) até alcançar um dos nodos filhos. A figura 3.1 demonstra como a série de questões realizadas para se chegar a uma conclusão sobre qual classe o registro pertence, pode ser organizada em forma de uma árvore hierárquica com nós e arestas direcionais.

Figura 3.1 – Árvore de decisão induzida do conjunto de dados da tabela 2.1



Fonte: Goldschmidt e Passos (2005)

3.1.3 Indução de uma árvore de decisão

A tarefa de construir uma Árvore de Decisão é comumente chamada de indução, e se dá por meio de um conjunto de treinamento que deva conter registros previamente rotulados. Na maioria dos algoritmos de indução de Árvore de Decisão corresponde a um procedimento guloso que recursivamente constrói a árvore de cima para baixo, ou seja, do nó raiz em direção aos nós terminais. A cada interação, a partir do conjunto de treinamento, os algoritmos procuram pelo atributo que melhor divide as classes para realizarem a ramificação da árvore, e recursivamente processam os subproblemas resultantes das ramificações.

Conforme Goldschmidt e Passos (2005), a fase de construção de Árvores de Decisão é composta por duas operações: (a) a avaliação e identificação de qual atributo divide melhor o conjunto de treinamento; e (b) a criação das partições usando o melhor atributo identificado para os casos pertencentes a cada nó.

Como definir o atributo que melhor divide o conjunto de treinamento? Certamente, testar todas as possibilidades não seria uma boa ideia e dependendo

do espaço de busca, que algumas vezes é bastante grande, isso se tornaria inviável e demasiadamente custoso. Portanto, para que a Árvore de Decisão tenha uma acurácia aceitável, deve-se definir medidas que auxiliem na escolha do atributo que melhor divide o conjunto de treinamento.

3.1.4 Métricas para escolha de atributos

Existem várias medidas que podem ser usadas para determinar a melhor forma de dividir os registros de um conjunto de dados. Tais medidas são definidas em função da distribuição de classes dos registros antes e depois da divisão (TAN; STEINBACH; KUMAR, 2009).

Segundo Tan, Steinbach e Kumar (2009), as métricas desenvolvidas para selecionar a melhor divisão, são frequentemente baseadas no grau de impureza dos nodos filhos. Quanto menor o grau de impureza, mais desigual é a distribuição das classes. Exemplificando, um nodo com distribuição de classes (0,1) tem impureza zero, em contrapartida um nodo com distribuição de classes uniforme possui a maior impureza. Ainda segundo Tan, Steinbach e Kumar (2009), são exemplos de métricas de impureza:

$$\text{Entropia}(t) = - \sum_{i=1}^c p(i / t) \log_2 p(i / t) \quad (3.1)$$

$$\text{Gini}(t) = 1 - \sum_{i=1}^c [p(i / t)]^2 \quad (3.2)$$

$$\text{Erro de Classificação}(t) = 1 - \max[p(i / t)] \quad (3.3)$$

Onde c é o número de classes, $p(i / t)$ é a razão entre o número de registros que pertencem a classes i sobre o número do total de elementos do conjunto de dados em questão.

Das três métricas de números 3.1, 3.2 e 3.3 iremos dar ênfase apenas a entropia, uma vez que é a métrica mais utilizada pelos algoritmos de indução de Árvore de Decisão junto com outra medida que é o ganho de informação, que por sua vez avalia a qualidade de cada condição de teste.

3.1.4.1 Entropia

Entropia é a medida que indica a homogeneidade dos exemplos contidos em um conjunto de dados. Permite caracterizar a "pureza" (e impureza) de uma coleção arbitrária de exemplos (OSÓRIO, 2000).

Conforme Goldschmidt e Passos (2005), em um conjunto de atributos, o grau de entropia expressa o grau de complexidade da informação contida no referido conjunto. Portanto, quanto menor a entropia, menor a quantidade de informação codificada em um ou mais atributos. Em contrapartida, quanto maior a entropia de um conjunto de atributos, maior a relevância destes atributos na descrição do conjunto de dados.

As métricas citadas acima conforme Tan, Steinbach e Kumar (2009), alcançam seus valores máximos quando a distribuição de classe é uniforme (p igual 0.5). E valores mínimos quando todos os registros fazem parte de uma mesma classe (p é igual a 0 ou 1), onde p é a fração de registro que pertence a cada classe.

Tomando o exemplo ilustrado na tabela 2.1, que é uma coleção de 14 exemplos com 9 exemplos positivos (Jogar Golfe = Sim) e 5 negativos (Jogar Golfe = Não), a entropia no conjunto de exemplos pode ser calculada através da equação 3.1:

$$\text{Entropia(Tabela 2.1)} = -\left(\frac{9}{14}\right) \log_2 \left(\frac{9}{14}\right) - \left(\frac{5}{14}\right) \log_2 \left(\frac{5}{14}\right) = 0.94$$

Para determinarmos qual atributo melhor divide o conjunto de dados, temos que comparar o grau de impureza do nodo pai (antes da divisão) com o grau de impureza dos nodos filhos (após a divisão). Quanto maior for a diferença melhor é o atributo. A diferença entre a entropia do nodo pai e do nodo filho recebe o nome de Ganho de Informação (TAN; STEINBACH; KUMAR, 2009).

3.1.4.2 Ganho de informação

O Ganho de Informação é baseado no índice de entropia para medição da homogeneidade de cada nó. O algoritmo ID3 (QUINLAN, 1986), foi pioneiro em indução de árvores de decisão, utiliza essa medida para determinar o quão boa é uma condição de teste realizada. Depois de feita a comparação do nodo pai com os

nodos filhos, o atributo que gerar uma maior diferença será escolhido como condição de teste.

O ganho de informação pode ser definido como a medida da efetividade de um atributo para classificar os dados de treinamento (GUARDA 2000). Ou seja, é a redução esperada na entropia do conjunto causada pelo particionamento por este atributo.

O ganho é dado formalmente pela Equação (3.4), na forma:

$$\text{Ganho}(A) = \text{Entropia}(t) - \sum_{i=1}^k \left[\left(\frac{|t_i|}{|t|} \right) \text{Entropia}(t_i) \right] \quad 3.4$$

Sendo que A é o atributo associado à partição t, por sua vez t é a partição associada ao nodo em questão, k é a quantidade de valores possíveis que o atributo A pode assumir e t_i é partição associada a cada um desses valores.

No capítulo 4 os conceitos de entropia e ganho de informação serão postos em prática.

3.1.5 *Overfitting* e *underfitting* de modelos de classificação

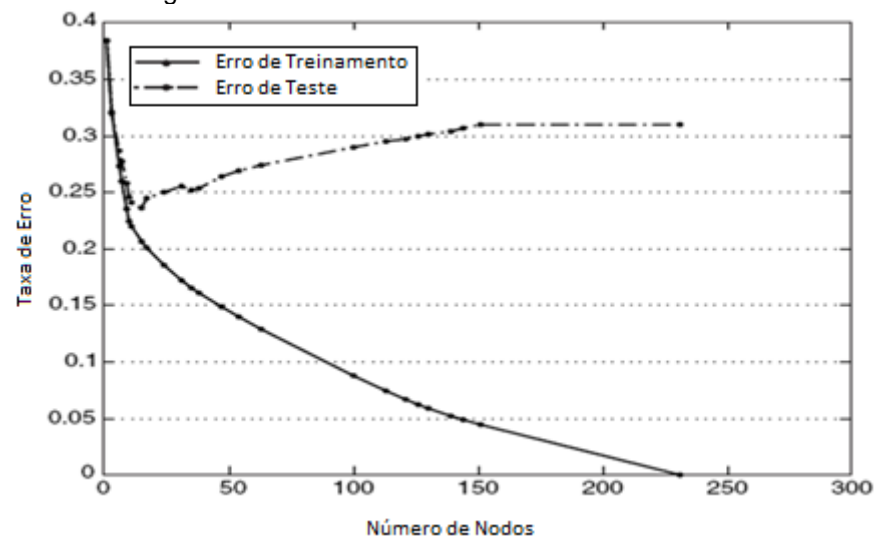
De acordo com Goldschmidt e Passos (2005), uma vez induzida uma hipótese (modelo de classificação), esta pode ser muito específica para o conjunto de treinamento utilizado. Caso este conjunto não seja suficientemente representativo, o modelo de classificação pode ter bom desempenho no conjunto de treinamento, mas não no conjunto de teste. Diz-se, neste caso, que o modelo ajustou-se em excesso ao conjunto de treinamento, ocorrendo um fenômeno denominado *overfitting*. Em contrapartida, quando o modelo ajusta-se muito pouco ao conjunto de treinamento, diz-se que ocorre um *underfitting*. Este fenômeno costuma ocorrer em função de parametrizações inadequadas do algoritmo de aprendizado. Por exemplo, um número alto de registros por nodo em uma árvore de decisão, ou uma tolerância de erro excessivamente alta. Já segundo Tan, Steinbach e Kumar (2009), *overfitting* e *underfitting* são duas patologias relacionadas à complexidade dos modelos.

Analisemos a figura 3.2 na qual temos, inicialmente, uma árvore simples (poucos nós) com elevadas taxas de erros de treinamento e teste, caracterizando o que chamamos de *underfitting*. Isto ocorre, pois, a árvore ainda não conhece bem a estrutura dos dados e, conseqüentemente, tem fraco desempenho nos dois

conjuntos, o que nos leva a pensar da seguinte maneira: “Já que o modelo ainda não conhece a estrutura dos dados devido à sua simplicidade, basta aumentarmos a sua complexidade (número de nodos) para que as taxas de erro caiam solucionando, assim, o problema do *underfitting*”.

Esta estratégia realmente soluciona o problema, pois, as taxas de erros caem. Entretanto, percebemos que, à medida que continuamos a aumentar o número de nodos da árvore, a taxa de erro treinamento continua a cair, mas a taxa de erro de teste aumenta. Tal situação é o que chamamos de *overfitting*.

Figura 3.2 – Taxas de erros de testes e de treinamento.



Fonte: Tan, Steinbach e Kumar (2009).

3.1.6 Algoritmos de Indução de árvores de decisão

Os algoritmos de indução de Árvores de Decisão geralmente empregam uma estratégia que cresce uma Árvore de Decisão tomando uma série de decisões locais consideradas ótimas sobre qual atributo usar para particionar os dados. Não se pode determinar qual é a melhor. Dependendo do problema, um algoritmo pode ser mais eficiente que outro.

Em 1983, o professor Ross Quinlan da universidade de Sidney deu sua enorme contribuição para os estudos que permitiram o aparecimento das Árvores de Decisão. Elaborou um algoritmo ID3 e logo em seguida surgiu o C4.5, esses algoritmos serão descritos com mais detalhes nesse trabalho.

3.1.6.1 ID3

O professor Ross Quinlan desenvolveu a tecnologia que permitiu o aparecimento das Árvores de Decisão. Muitas pessoas na indústria de *Data Mining* consideram Quinlan como o "pai das Árvores de Decisão". A contribuição de Quinlan foi a elaboração de um novo algoritmo chamado ID3, desenvolvido em 1983. O ID3 e suas evoluções (ID4, ID6, C 4.5, See 5) são muito bem adaptadas para usar em conjunto com as Árvores de Decisão, na medida em que eles produzem regras ordenadas pela importância. Essas regras são, então, usadas para produzir um modelo de Árvore de Decisão dos fatos que afetam os itens de saída (DWBRASIL, 2002).

O algoritmo ID3 foi um dos primeiros algoritmos de Árvore de Decisão, tendo sua elaboração baseada em sistemas de inferência e em conceitos de sistemas de aprendizagem. Logo após foram elaborados diversos algoritmos, sendo os mais conhecidos: C4.5, CART (Classification and Regression Trees), CHAID (Chi Square Automatic Interaction Detection), entre outros (GARCIA, 2000).

O ID3 constrói a árvore de cima para baixo (top down) com o intuito de escolher sempre o melhor atributo para cada nó de decisão da árvore. O ID3 faz uma avaliação da informação contida em cada atributo baseado no ganho de informação. Sendo que o atributo mais importante, ou seja, aquele com maior valor de ganho de informação é colocado na raiz e, de forma *top down*, a árvore é construída recursivamente. O algoritmo é aplicado recursivamente a cada nó descendente, até que algum critério de parada seja atingido. Isto gera uma Árvore de Decisão aceitável, na qual o algoritmo nunca retrocede para reconsiderar escolhas feitas anteriormente.

Uma das vantagens do ID3 é sua simplicidade, o seu processo de construção torna relativamente simples a compreensão de seu funcionamento.

Por vez, a maior desvantagem do ID3 é o fato de que a árvore induzida é imutável, sendo assim não se pode eficientemente reutilizar a árvore sem a reconstruir.

Abaixo uma lista de vantagens e desvantagens do algoritmo ID3:

Vantagens do ID3:

- Obtém regras a partir da experiência;

- As regras geradas estão livres de inconsistências, eliminando o processo de verificação e ratificação no que diz respeito à redundância e conflito de regras;
- Simplicidade de implementação e execução.

Desvantagens do ID3:

- Aplicado apenas a domínios onde tanto as classes como os valores dos atributos são mutuamente exclusivos;
- Não contorna domínios que contenha ruídos;
- Gera regras ideais e completas (Acurácia 100%), o que o torna mais suscetível a *overfitting*;

O C4.5 é desenvolvido com o intuito de sanar a deficiência do algoritmo ID3 que é a possibilidade de gerar árvores demasiadamente ajustadas aos dados de treinamento (*overfitting*).

3.1.6.2 C4.5

O C4.5 é um dos mais tradicionais algoritmos de classificação (QUINLAN, 1993). Inspirado no algoritmo ID3, o método C4.5 procura abstrair Árvores de Decisão a partir de uma abordagem recursiva de particionamento das bases de dados (GOLDSCHMIDT, PASSOS, 2005).

Já segundo Souza (2007) a principal diferença entre o ID3 e o C4.5 é a capacidade deste trabalhar com atributos cujo domínio seja contínuo. Afirma ainda que no caso de atributos discretos, a própria natureza dos valores pode conter informações com alto grau de correlação em relação à classificação das ocorrências, ou seja, alguns valores para este tipo de atributo podem ser um fator determinante para a escolha da classe.

O ID3 cria uma sub-árvore para cada valor possível do atributo (discreto). Entretanto, esta abordagem não pode ser empregada para atributos contínuos, pois se cada valor deste atributo originar uma sub-árvore, é provável que cada uma delas tenha apenas um único exemplo, isso acabaria, portanto, anulando o poder preditivo da árvore, já que valores desconhecidos jamais seriam classificados.

Para solucionar este problema Quinlan (QUINLAN, 1993) propõe uma divisão binária para atributos contínuos, ou seja, propõe testes dicotômicos da forma $A \leq k$ (A

$< k$ ou $A \geq k$), onde A é um atributo qualquer e k é número real denominado limiar. Isto resulta em árvores estritamente binárias para bases contínuas. Consequentemente, a escolha do melhor atributo deve levar em conta o melhor limiar de particionamento para classificar os dados. Quinlan nos fornece o arcabouço necessário para encontrar o melhor limiar dentre os limiares candidatos. Em (QUINLAN, 1986), ele sugere que os limiares candidatos de um atributo A sejam escolhidos da seguinte forma:

1. Ordenam-se os valores distintos de A para obter a sequência v_1, \dots, v_n .
2. Os limiares candidatos são todos os pontos médios entre v_i e v_{i+1} , onde $i \in (1, 2, \dots, n-1)$:

Após a seleção dos limiares candidatos, utiliza-se a mesma estratégia do ID3 para a seleção do melhor limiar: o ganho de informação. Assim, todos os limiares são testados e é escolhido aquele que apresenta maior ganho de informação. Esse procedimento é feito para todos os limiares de todos os atributos e repetido para a construção de cada sub-árvore (SOUZA, 2007).

Após a construção de uma Árvore de Decisão é importante avaliá-la. Esta avaliação é realizada através da utilização de dados que não tenham sido usados no treinamento. Esta estratégia permite estimar como a árvore generaliza os dados e se adapta a novas situações, podendo, também, se estimar a proporção de erros e acertos ocorridos na construção da árvore (Brazdil et al., 2003).

3.1.7 Características de indução de Árvore de Decisão:

Algumas características importantes de algoritmos Árvores de Decisão de acordo com Tan, Steinbach e Kumar, (2009):

1. Não assume nenhuma distribuição particular para os dados para construir modelos de classificação. Em palavras mais simples, a Árvore de Decisão não requer quaisquer suposições a priori quanto ao tipo de distribuição de probabilidades satisfeita pela classe e outros atributos.
2. Encontrar a Árvore de Decisão ótima é um problema NP-Completo. Muitos algoritmos empregam alguma abordagem baseada em heurísticas para guiá-lo pelo vasto espaço de hipóteses. Por exemplo, os algoritmos

apresentados utilizam estratégias gulosas de particionamento recursivo para construção da árvore.

3. As técnicas desenvolvidas para construção de Árvores de Decisão são computacionalmente menos custosas, tornando possível construir, rapidamente, modelos mesmo quando um conjunto de treinamento muito grande é utilizado. Além disso, uma vez que a Árvore de Decisão tenha sido construída, classificar um novo registro é extremamente rápido, tendo no pior caso uma complexidade $O(w)$, onde w é a máxima altura da árvore.
4. Árvores de Decisão, especialmente as de menor tamanho, são relativamente fáceis de interpretar. As acurácias das Árvores de Decisão são comparáveis as de outras técnicas para muitos conjuntos de dados simples.
5. Árvores de Decisão são bastante robustas á presença de ruído, principalmente, quando métodos para evitar o *overfitting* são empregados.

3.2 Naive bayes

Esta seção introduz uma a técnica conhecida como *Naive Bayes* que é baseada no Teorema de *Bayes*, que por sua vez é um princípio estatístico que combina conhecimento prévio das classes com novas evidências colhidas dos dados. A seção inicia com alguns conceitos importantes para um bom entendimento da mesma. Descreve o teorema de *Bayes* e o seu uso no processo de classificação seguido pelo classificador de *Naive Bayes*.

3.2.1 Definição

Um classificador de *Naive Bayes* baseia-se no Teorema de *Bayes*, que por sua vez está relacionado ao cálculo das probabilidades condicionais. É utilizado em tarefas de classificação, como o próprio nome sugere (GOLDSCHMIDT; PASSOS, 2005). De acordo com Tan, Steinbach e Kumar (2009), O classificador de *Naive Bayes* analisa a probabilidade condicional de classe supondo que os atributos são condicionalmente independentes, dado o rótulo de classe. Possivelmente, o modelo de rede bayesiana mais utilizado na aprendizagem de máquina é o modelo de *Naive Bayes* (RUSSEL; NORVIG, 2004).

3.2.2 Probabilidade incondicional

A probabilidade incondicional, que também é chamada de probabilidade a priori ou marginal, pode ser representada pela notação $P(A)$, significando a probabilidade que a proposição A seja verdadeira na ausência de qualquer outra informação relevante (RUSSEL; NORVIG, 2004). É importante salientar que $P(A)$ somente pode ser usada quando não há outra informação relacionada. Logo, quando uma nova informação relevante B torna-se conhecida, é necessário raciocinar com a probabilidade condicional de A dada B , ao invés de raciocinar com a probabilidade a priori (RUSSEL; NORVIG, 2004).

3.2.3 Probabilidade condicional

A probabilidade de um evento A ocorrer, dado que outro evento B ocorreu, é chamada probabilidade condicional do evento A dado B .

Podemos citar como exemplo, a probabilidade de que uma pessoa venha a ter problemas do coração dado que é sedentária ou ainda, dessa pessoa contrair diabetes dado que seus pais são diabéticos.

Uma probabilidade condicional de acordo com Tan, Steinbach e Kumar (2009), é a de que uma variável aleatória receba um determinado valor dado que o resultado de outra variável aleatória seja conhecido. Exemplificando, a probabilidade condicional $P(Y=y/X=x)$ refere-se à probabilidade da variável Y receber o valor y , dado que o valor da variável X seja x .

Quando o especialista do domínio obtém alguma evidência anteriormente desconhecida, as probabilidades a priori devem ser revistas, pois podem perder a sua utilidade. Ao invés delas, devem ser usadas probabilidades condicionais (também denominadas probabilidades a posteriori), com a notação $P(X/Y)$. Essa expressão é lida como “a probabilidade de X , dado que tudo o que sabemos é Y ” (RUSSEL; NORVIG, 2004).

3.2.4 Independência condicional

Segundo (TAN; STEINBACH; KUMAR, 2009), três variáveis X , Y e Z ao denotarem conjuntos de variáveis aleatórias, as variáveis de X são ditas condicionalmente independentes de Y dado Z , se a condição seguinte for respeitada:

$$P(X/Y, Z) = P(X/Z) \quad (3.7)$$

As relações de independência condicional são usadas para diminuir a dimensionalidade e o número de declarações de probabilidade condicional, simplificando assim os cálculos.

Sendo que a suposição de independência condicional entre X e Y pode ser declarada formalmente da seguinte maneira:

$$P(X/Y = y) = \prod_{i=1}^d P(X_i/Y = Y) \quad (3.8)$$

Onde cada conjunto de atributos $X = \{X_1, X_2, X_3, \dots, X_d\}$ consiste de d atributos.

3.2.5 Teorema de bayes

É um princípio estatístico que combina o conhecimento prévio com novas evidências colhidas dos dados (TAN; STEINBACH; KUMAR, 2009).

Sejam X e Y variáveis aleatórias, podemos definir a probabilidade junta, $P(X=x, Y=y)$, lê-se: probabilidade da variável X receber o valor x e variável Y receber o valor y. Como já foi dito na seção 3.2.3, a probabilidade condicional é quando a variável X recebe um valor dado que o valor de Y é conhecido $P(Y = y | X = x)$.

Relacionado as probabilidades condicionais e juntas temos a seguinte forma:

$$P(X,Y) = P(Y/X) \times P(X) = P(X/Y) \times P(Y) \quad (3.9)$$

Igualando as duas últimas expressões na equação 3.10, leva a seguinte fórmula, conhecida como teorema de Bayes:

$$P(Y/X) = \frac{P(X,Y)}{P(X)} \quad (3.10)$$

3.2.6 Usando o teorema de *bayes* para a classificação

Suponhamos um conjunto de atributos denotado por X e que Y por sua vez denote a variável de classe. Se esta tiver um relacionamento não determinístico com os atributos, então podemos tratar X e Y com variáveis aleatórias e capturar seu relacionamento usando probabilisticamente $P(Y|X)$, esta probabilidade também é conhecida como probabilidade posterior de Y , em oposição à sua probabilidade anterior, $P(Y)$ (TAN; STEINBACH; KUMAR, 2009).

Ainda segundo Tan, Steinbach e Kumar (2009), durante a fase de treinamento deve-se descobrir as probabilidades posteriores $P(Y|X)$ para cada uma das combinações de X e Y com base nas informações coletadas a partir dos dados de treinamento. Conhecendo estas probabilidades, um registro de teste X' pode ser classificado localizando-se a classe Y' que maximiza a probabilidade posterior, $P(Y'|X')$.

Para exemplificar considere o exemplo retirado da tabela 2.1 que possui o seguinte conjunto de atributos: $X = (\text{Aparência} = \text{Ensolarada}, \text{Temperatura} = 72^\circ \text{F}, \text{Umidade} = 95\%, \text{Vento} = \text{Não})$. Para classificar o exemplo precisamos calcular as probabilidades posteriores para $P(\text{Jogar} = \text{Sim}|X)$ e $P(\text{Jogar} = \text{Não}|X)$. Os cálculos terão como base as informações da tabela 2.1. Se $P(\text{Jogar} = \text{Sim}|X) > P(\text{Jogar} = \text{Não}|X)$ o exemplo será classificado como Sim, se for o contrário será classificado como Não.

Ainda que para um número moderado de atributos, é difícil avaliar as probabilidades posteriores com precisão para cada combinação possível de rótulo de classe, isso por que requer um conjunto de treinamento muito grande. A utilidade do teorema de *Bayes* está principalmente em permite que se calcule a probabilidade posterior em termos da probabilidade anterior $P(Y)$, da probabilidade condicional de classe $P(X|Y)$ e da evidência $P(X)$, o que é demonstrado na equação 3.11:

$$P(Y/X) = \frac{P(X/Y) \times P(Y)}{P(X)} \quad (3.11)$$

O termo $P(X)$ pode ser ignorado, tendo em vista que o mesmo é constante para diferentes valores de Y . A probabilidade anterior $P(Y)$ é facilmente verificada a

partir do conjunto de treinamento calculando-se a fração de registro de treinamento que pertence a cada classe.

De um modo mais geral e para facilitar o entendimento, a expressão 3.11 pode ser escrita assim como descrito na equação 3.12:

$$\text{Posterior} = \frac{\text{verossimilhança x conhecimento prévio}}{\text{evidência}} \quad (3.12)$$

A probabilidade de classe $P(X|Y)$, será avaliada com uma das mais simples, porém não menos importante implementação de método de classificação Bayesiano, o método de *Naive Bayes*.

3.2.5 Como funciona um classificador de *Naive Bayes*

Como o classificador de *Naive Bayes* avalia a probabilidade condicional de classe supondo a independência condicional entre os atributos, estima-se a probabilidade condicional de cada X_i , dado $Y(\text{classe})$, em vez de calcular a probabilidade condicional de classe para cada combinação de X .

Consequentemente uma boa estimativa da probabilidade pode ser alcançada com menos esforços, tendo em vista que não é necessário um grande conjunto de treinamento.

O classificador de *Naive Bayes* calcula a probabilidade posterior para cada classe Y para que possa classificar um determinado registro, equação 3.13. Sendo que o objetivo é escolher a classe que maximize o numerador, $P(Y) \prod_{i=1}^d P(X_i|Y)$, já que $P(X)$ é fixo.

$$P(Y|X) = \frac{P(Y) \prod_{i=1}^d P(X_i|Y)}{P(X)} \quad (3.13)$$

Para exemplificar o funcionamento do classificador *Naive Bayes*, considere uma instância que é representada pelo seguinte conjunto de atributos $X = (\text{Visual} = \text{Sol}, \text{Temperatura} = \text{Agradável}, \text{Umidade} = \text{Alta}, \text{Vento} = \text{Forte})$.

Tabela 3.1 – Informações sobre condições do tempo

| Visual | Temperatura | Umidade | Vento | Jogar |
|---------|-------------|---------|-------|------------|
| Sol | Quente | Alta | Fraco | Não |
| Sol | Quente | Alta | Forte | Não |
| Nublado | Quente | Alta | Fraco | Sim |
| Chuva | Agradável | Alta | Fraco | Sim |
| Chuva | Frio | Normal | Fraco | Sim |
| Chuva | Frio | Normal | Forte | Não |
| Nublado | Frio | Normal | Forte | Sim |
| Sol | Agradável | Alta | Fraco | Não |
| Sol | Frio | Normal | Fraco | Sim |
| Chuva | Agradável | Normal | Fraco | Sim |

Primeiro calcula-se a probabilidade anterior de classe para as classes Sim e Não, com base nas informações da tabela 3.1 temos:

$$P(\text{Sim} | X) = 6/10 \text{ e } P(\text{Não} | X) = 4/10.$$

O próximo passo é calcular a probabilidade condicional para cada atributo X_i :

$$P(X|\text{Sim}) = P(\text{Visual} = \text{Sol} | \text{Sim}) \times P(\text{Temperatura} = \text{Agradável} | \text{Sim}) \times P(\text{Umidade} = \text{Alta} | \text{Sim}) \times P(\text{Vento} = \text{Forte} | \text{Sim}) = 1/6 \times 2/6 \times 2/6 \times 1/6 = 0,0030.$$

$$P(X|\text{Sim}) = P(\text{Visual} = \text{Sol} | \text{Não}) \times P(\text{Temperatura} = \text{Agradável} | \text{Não}) \times P(\text{Umidade} = \text{Alta} | \text{Não}) \times P(\text{Vento} = \text{Forte} | \text{Não}) = 3/4 \times 1/4 \times 3/4 \times 2/4 = 0,0703.$$

Finalmente, calculando as probabilidades posteriores para a classe Sim e Não temos:

$$P(\text{Sim}|X) = 6/10 \times 0,0030 = 0,0018.$$

$$P(\text{Não}|X) = 4/10 \times 0,0703 = 0,0281.$$

Uma vez que $P(\text{Não}|X) > P(\text{Sim}|X)$, o registro é classificado como Não.

3.2.6 Características de classificadores *Naive Bayes*

Características relevantes inerentes aos classificadores *Naive Bayes* segundo Tan, Steinbach; Kumar, (2009):

1. São robustos para pontos de ruídos isolados porque calculam a média de tais pontos ao avaliar probabilidades condicionais a partir de dados.
2. São robustos para atributos irrelevantes. Se X_i for um atributo irrelevante, então $P(X|Y)$ se torna quase que uniformemente distribuído. A

probabilidade condicional de classe para X_i não tem impacto no cálculo geral da probabilidade posterior.

3. Atributos correlacionados podem degradar o desempenho de classificadores *Naive Bayes* porque a suposição de independência condicional não é mais verdadeira para tais atributos.

4 UMA DISCUSSÃO ENTRE OS CLASSIFICADORES: *NAIVE BAYES* E ÁRVORE DE DECISÃO.

O capítulo 3 descreveu os conceitos de entropia, ganho de informação, probabilidade condicional e focou principalmente em duas das mais populares técnicas de classificação: *Naive Bayes* e Árvore de Decisão. Este capítulo introduz uma base de dados frequentemente utilizada na literatura a weather tabela 4.1, que está presente no software WEKA (Waikato Environment for Knowledge Analysis), que é um software de domínio público, desenvolvido no meio acadêmico da Universidade de Waikato, na Nova Zelândia, em 1999 e que está disponível no endereço eletrônico: <https://sourceforge.net/projects/weka/files/weka-3-6/3.6.9/>. Em seguida será realizada uma discussão entre os classificadores *Naive Bayes* e Árvore de Decisão, utilizando os conceitos de entropia e ganho de informação. A seção 4.1 guia essa discussão levando em consideração a escolha dos atributos mais promissores do conjunto de dados; já a seção 4.2 toma como parâmetros para essa discussão os atributos irrelevantes e redundantes.

4.1 Atributos mais promissores do conjunto de treinamento.

Nesta seção é demonstrado como os atributos são tratados pelos classificadores durante a concepção do modelo de classificação. Nesse contexto, as métricas entropia e ganho de informação desempenham um importante papel.

Um modelo de classificação é gerado a partir do momento em que um conjunto de treinamento, com um número de N exemplos, é submetido ao algoritmo de aprendizagem de uma técnica de classificação. Esse é um procedimento comum aos classificadores *Naive Bayes* e Árvore de Decisão. A diferença está no esquema utilizado pela Árvore de Decisão para selecionar os melhores atributos com o objetivo de minimizar a profundidade da árvore, obter uma classificação correta com um pequeno número de testes. Esse procedimento não ocorre no *Naive Bayes*.

Segundo Russell e Norvig (2004), o algoritmo de aprendizagem de Árvore de Decisão testa primeiro o atributo “mais importante”, que é aquele que faz maior diferença na classificação de um exemplo. Para um exemplo mais concreto e com o intuito de demonstrar essa questão da importância de cada atributo, considere o conjunto de dados da tabela 4.1 que tem informações sobre as condições climáticas

e recomendações que podem ajudar a determinar um bom dia para jogar tênis. É composta pelos atributos: Visual, Temperatura, Umidade, Vento e pelo atributo classe “Jogar” que pode assumir os valores “Sim ou Não”, possui 14 exemplos sendo 09 positivos (Jogar = Sim) e 05 negativos (Jogar = Não).

Tabela 4.1 – Conjunto de dados weather

| Exemplos | Atributos | | | | Jogar |
|-----------------|------------------|--------------------|----------------|--------------|--------------|
| | Visual | Temperatura | Umidade | Vento | |
| 1 | Sol | Quente | Alta | Fraco | Não |
| 2 | Sol | Quente | Alta | Forte | Não |
| 3 | Nublado | Quente | Alta | Fraco | Sim |
| 4 | Chuva | Agradável | Alta | Fraco | Sim |
| 5 | Chuva | Frio | Normal | Fraco | Sim |
| 6 | Chuva | Frio | Normal | Forte | Não |
| 7 | Nublado | Frio | Normal | Forte | Sim |
| 8 | Sol | Agradável | Alta | Fraco | Não |
| 9 | Sol | Frio | Normal | Fraco | Sim |
| 10 | Chuva | Agradável | Normal | Fraco | Sim |
| 11 | Sol | Agradável | Normal | Forte | Sim |
| 12 | Nublado | Agradável | Alta | Forte | Sim |
| 13 | Nublado | Quente | Normal | Fraco | Sim |
| 14 | Chuva | Agradável | Alta | Forte | Não |

Fonte: Software Weka versão 3.6.9 (2013)

Para decidir quais os atributos mais importantes, faz-se uso das métricas que selecionam o melhor atributo como já foi visto no capítulo 3 seção 3.1.4. Primeiramente calculamos a entropia da base de dados tabela 4.1, antes de dividi-la por qualquer atributo, utilizando a equação 3.1 temos:

$$\text{Entropia (weather)} = -9/14 \times (\log_2 (9/14)) - 5/14 \times \log_2 (5/14) = 0,94.$$

Em seguida cada atributo será avaliado individualmente. Calcula-se a entropia de cada atributo com o objeto de descobrir qual deles resultará em um maior ganho de informação, esse será utilizado como primeiro teste na árvore. Começando pelo atributo Visual, o conjunto de treinamento será dividido em três partições, uma vez que este atributo pode assumir três valores possíveis: (chuva, nublado, sol), da seguinte forma:

1. Três exemplos positivos e dois negativos são definidos pelo atributo Visual = chuva;
2. Quatro exemplos positivos são definidos por Visual = nublado;
3. Dois exemplos positivos e três negativos são definidos por Visual = sol.

A seguir calcula-se a entropia para as partições geradas a partir do atributo Visual:

$$\begin{aligned} \text{Entropia (Visual)} &= 5/14 \times (-3/5 \times \log_2 (3/5) - 2/5 \times \log_2 (2/5)) \\ &+ 4/14 \times (-4/4 \times \log_2 (4/4) - 0/4 \times \log_2 (0/4)) \\ &+ 5/14 \times (-2/5 \times \log_2 (2/5) - 3/5 \times \log_2 (3/5)) = 0,694. \end{aligned}$$

Logo o ganho de informação do atributo Visual será calculado pela diferença da entropia do conjunto antes da divisão, tabela 4.1 pela entropia do atributo Visual fazendo uso da equação 3.4 tem o seguinte:

$$\text{Ganho(Visual)} = \text{Entropia(weather)} - \sum_{i=1}^{K=3} \left[\left(\frac{|t_i|}{|t|} \right) \text{Entropia}(t_i) \right] = 0,940 - 0,694 = 0,246.$$

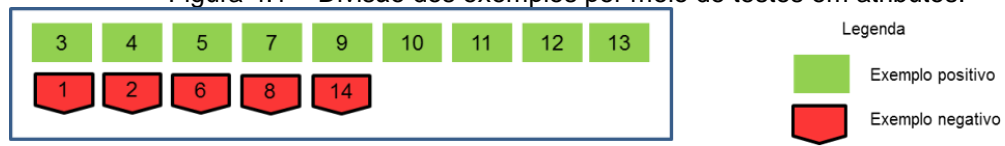
Onde K = 3 representa a quantidade de valores possíveis que o atributo Visual pode assumir e t_i a partição associada a cada um desses valores.

De forma análoga será feito o cálculo do ganho de informação para os demais atributos, sendo que aquele que resultar em um maior ganho de informação será o primeiro atributo a dividir o conjunto de treinamento, ou seja, será o ponto de separação no nó raiz. Abaixo está apresentado de forma resumida, o ganho de informação para os demais atributos:

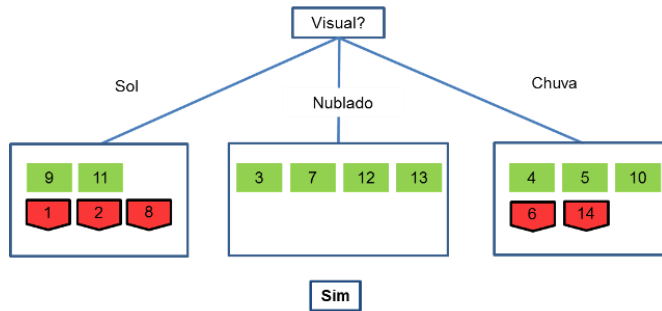
- Ganho (Umidade) = $0,940 - 0,789 = 0,151$
- Ganho (Vento) = $0,940 - 0,892 = 0,048$
- Ganho (Temperatura) = $0,940 - 0,911 = 0,029$.

Portanto o atributo que foi selecionado e será usado como teste no nó raiz, uma vez que no início todos os exemplos estão no nó raiz, foi o atributo Visual, pois o ganho de informação para este atributo foi o maior. A figura 4.1 mostra a divisão do conjunto de treinamento em função dos atributos Visual e Umidade que obtiveram o maior ganho de informação respectivamente.

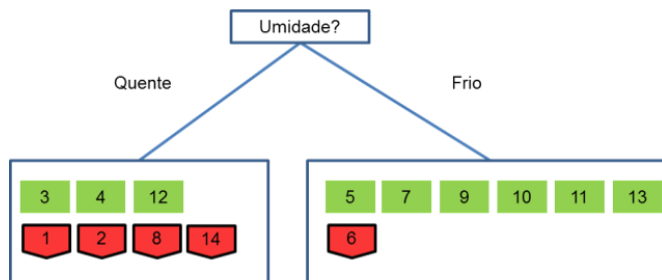
Figura 4.1 – Divisão dos exemplos por meio de testes em atributos.



(a) Conjunto de treinamento com exemplos positivos e negativos.



(b) A divisão pelo atributo visual, nos permite responder Sim se a resposta ao teste for nublado.



(c) O atributo umidade não faz uma boa divisão dos exemplos positivos e negativos.

Fonte: adaptado de Russel e Norvig (2004).

Russell e Norvig (2004) definem como atributo perfeito aquele que divide o conjunto de treinamento exatamente em subconjuntos nos quais todos são positivos ou todos negativos. Não é o caso do atributo Visual, mas ainda assim se mostra razoavelmente bom, uma vez que se a resposta ao teste for Nublado, ficamos somente com exemplos positivos (3, 7, 12 e 13), figura 4.1(b), logo a classe é Sim, isso significa que é um bom dia para se jogar tênis. Em contrapartida o atributo umidade é um atributo fraco, uma vez que dividiu o conjunto em dois subconjuntos onde ambos possuem exemplos positivos e negativos figura 4.1(c).

Diferente da Árvore de Decisão no classificador *Naive Bayes* todos os atributos do conjunto de treinamento são igualmente importantes, uma vez que todos são utilizados na tarefa de classificar um exemplo. Para ilustrar essa situação

vamos considerar um dia de sol, quente, de alta umidade e vento forte. Aplicando o classificador de Bayes temos o seguinte:

Primeiramente vamos encontrar a probabilidade anterior de cada classe que podem ser calculadas levando em consideração a fração de registros de treinamento (figura 4.1) pertencente a cada classe. Uma vez que nove registros pertencem à classe Sim e cinco a classe Não, temos $P(\text{Sim}) = 9/14$ e $P(\text{Não}) = 5/14$.

Tabela 4.2 – Probabilidade condicional para os atributos da tabela 4.1

| | |
|---|---|
| $P(\text{Visual} = \text{sol} \mid \text{Não}) = 3/5$ | $P(\text{Umidade} = \text{alta} \mid \text{Não}) = 4/5$ |
| $P(\text{Visual} = \text{sol} \mid \text{Sim}) = 2/9$ | $P(\text{Umidade} = \text{alta} \mid \text{Sim}) = 3/9$ |
| $P(\text{Temperatura} = \text{quente} \mid \text{Não}) = 2/5$ | $P(\text{Vento} = \text{forte} \mid \text{Não}) = 3/5$ |
| $P(\text{Temperatura} = \text{quente} \mid \text{Sim}) = 2/9$ | $P(\text{Vento} = \text{forte} \mid \text{Sim}) = 3/9$ |

Utilizando as informações da tabela 4.2, que mostra as probabilidades condicionais somente para os valores de atributo que participam do exemplo acima, podemos definir a probabilidade posterior pelo produto entre a probabilidade anterior de classe ($P(\text{Sim})$ e $P(\text{Não})$) e as probabilidades condicionais de classe $\prod_{i=1}^d P(X|Y)$, como já foi discutido na seção 3.2.5 utilizando a equação 3.13.

$P(\text{Jogar} = \text{Sim} \mid \text{sol, quente, alta umidade, vento forte}) =$

$$\begin{aligned}
 &P(\text{Sim}) \times P(\text{visual} = \text{sol}/\text{Sim}) \times \\
 &P(\text{temperatura} = \text{quente}/\text{Sim}) \times \\
 &P(\text{umidade} = \text{alta}/\text{Sim}) \times \\
 &(\text{vento} = \text{forte}/\text{Sim}) = 9/14 \times 2/9 \times 2/9 \times 3/9 \times 3/9 = \mathbf{0,0035}.
 \end{aligned}$$

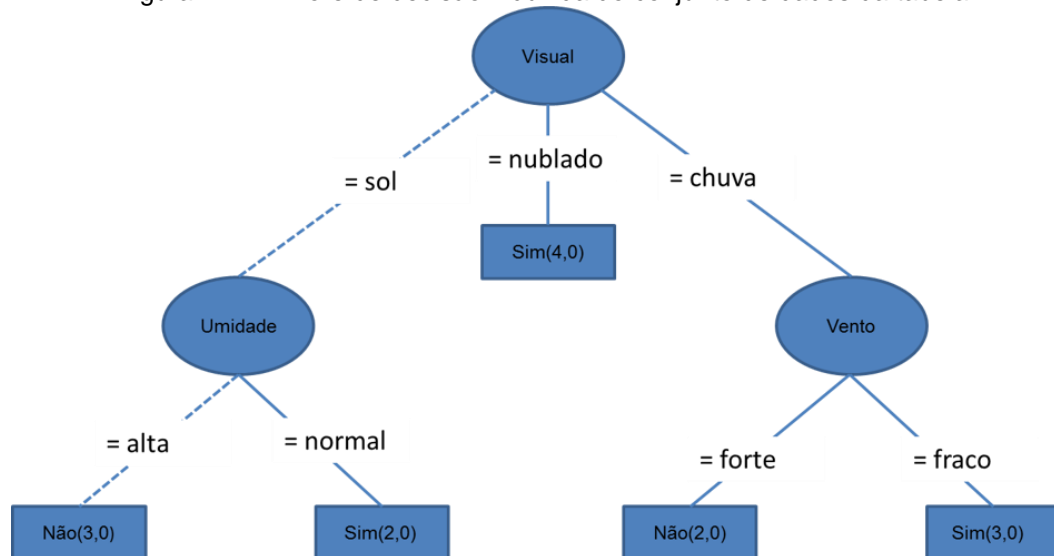
$P(\text{Jogar} = \text{Não} \mid \text{sol, quente, alta umidade, vento forte}) =$

$$\begin{aligned}
 &P(\text{Não}) \times P(\text{visual} = \text{sol}/\text{Não}) \times \\
 &P(\text{temperatura} = \text{quente}/\text{Não}) \times \\
 &P(\text{umidade} = \text{alta}/\text{Não}) \times \\
 &P(\text{vento} = \text{forte}/\text{Não}) = 5/14 \times 3/5 \times 2/5 \times 4/5 \times 3/5 = \mathbf{0,0411}.
 \end{aligned}$$

Uma vez que $P(\text{Jogar} = \text{Não}) > P(\text{Jogar} = \text{Sim})$, o exemplo é classificado como Não, logo não é um bom dia para jogar tênis. Seguindo a linha pontilhada da Árvore de Decisão figura 4.2, chegaremos a mesma conclusão após uma série de testes sobre os atributos da Árvore da seguinte forma:

A primeira questão é qual é o visual do dia. Se a resposta for nublado, definitivamente é um bom dia para jogar tênis. No caso da instância acima a resposta para o primeiro questionamento é sol. O próximo passo é saber qual é a umidade do dia, obtemos como resposta alta, portanto pode-se concluir que não é um bom dia para se jogar tênis, ou seja, a classe resultante é NÃO.

Figura 4.2 – Árvore de decisão induzida do conjunto de dados da tabela 4.1.



Fonte: Adaptada do software Weka versão 3.6.9 (2013).

4.2 Atributos irrelevantes e redundantes presentes no conjunto de treinamento

A seção 4.2 descreve o comportamento apresentado pelos classificadores *Naive Bayes* e *Árvore de Decisão* quando o conjunto de treinamento possui atributos irrelevantes e atributos redundantes.

A figura 4.2 deixa evidente que o atributo “temperatura”, que faz parte do conjunto de treinamento, não foi utilizado pela *Árvore de Decisão*, o que sugere que a escolha dos melhores atributos para dividir o conjunto de treinamento, deixa de fora aqueles atributos que não influenciam ou que influencia muito pouco na classificação, esses atributos são chamados de irrelevantes. Segundo Goldschmidt e Passos (2005), uma *Árvore de Decisão* é induzida a partir do conjunto de dados, de tal modo que aqueles atributos que não são utilizados na árvore são considerados irrelevantes para o problema.

De acordo com Russell e Norvig (2004), se um conjunto de exemplos for dividido por um atributo irrelevante, espera-se de um modo geral que os

subconjuntos resultantes dessa divisão tenham aproximadamente as mesmas proporções de exemplos positivos e negativos que o conjunto original. O ganho de informação estará perto de zero. Desse modo, o ganho de informação dá uma boa indicação de que o atributo é irrelevante.

Dessas observações, surge outra importante e significativa diferença entre os dois classificadores, que é a utilização de atributos irrelevantes da base de dados. O *Naive Bayes*, ao contrário da Árvore de Decisão, como já foi dito, utiliza todos os atributo inclusive temperatura que tem ganho de informação 0,029, de todos é o que mais se aproxima de zero.

Ocorre que o classificador *Naive Bayes* não tem seu desempenho degradado, por mais que utilize atributos que não agregam informação útil no processo de classificação. Os classificadores de *Naive Bayes* são robustos para atributos irrelevantes, uma vez que um atributo X_i é irrelevante, a probabilidade condicional de classe para esse atributo se torna quase que uniformemente distribuído e não gera impacto no cálculo geral da probabilidade posterior (TAN; STEINBACH; KUMAR, 2009).

Em contrapartida, a Árvore de Decisão é sensível a atributos irrelevantes. De acordo com Tan; Steinbach e Kumar (2009) há situações em que a presença de muitos atributos irrelevantes na base de dados pode levar a escolha acidental de um deles durante o processo de crescimento da árvore, a consequência disso seria uma árvore de tamanho maior que o necessário.

Tão indesejáveis quanto os atributos irrelevantes são os atributos redundantes, uma vez que o objetivo da maioria dos algoritmos de classificação é chegar a modelos que obtenham a maior precisão ou, conseqüentemente, a menor taxa de erro no conjunto de treinamento. Atributos redundantes mantêm uma correlação com outros atributos, de tal forma que não agregam novas informações. Nas palavras de Tan; Steinbach e Kumar (2009), características redundantes em uma base de dados, duplicam muitas ou todas as informações inseridas em um ou mais atributos.

Vamos supor que na tabela 4.1 tivesse um atributo com nome de Aparência com os mesmo valores do atributo Visual para os 14 exemplos. Diante de uma situação evidente de atributo redundante um algoritmo de aprendizagem de Árvore

de Decisão, se comportaria do seguinte modo de acordo com Tan; Steinbach e Kumar (2009), um dos atributos redundante será descartado, ou seja, não será usado para fazer a divisão assim que o outro tiver sido escolhido. Consequentemente a precisão de uma Árvore de Decisão, não é afetada adversamente na presença de atributos redundantes.

Em contrapartida o classificador *Naive Bayes* é sensível à presença de muitos atributos redundantes. De acordo com Kira e Rendell (1992), os atributos redundantes prejudicam o desempenho dos algoritmos de aprendizagem tanto na velocidade quanto na taxa de acerto. Os métodos de aprendizagem indutiva funcionam melhor quando são alimentados com características relevantes (RUSSELL; NORVIG, 2004).

A tabela 4.3 mostra de forma resumida as características dos classificadores *Naive Bayes* e Árvore de Decisão, em relação aos critérios que fazem parte dessa discussão:

Tabela 4.3 – Características da Árvore de Decisão e *Naive Bayes*

| Classificadores | Seleção interna de atributos | Atributos Irrelevantes | Atributos redundantes |
|------------------------|-------------------------------------|-------------------------------|------------------------------|
| Árvore de Decisão | Sim | Sensível | Não sensível |
| <i>Naive Bayes</i> | Não | Não sensível | Sensível |

- Os algoritmos de Árvore de Decisão descrito neste trabalho fazem a seleção interna de atributos, com o objetivo de encontrar os atributos mais importantes para a construção do modelo de aprendizagem.
- Para o algoritmo de *Naive Bayes* todos os atributos são igualmente importantes, uma vez que não há uma seleção com o proposito de escolher os atributos mais relevantes.
- Os algoritmos de Árvore de Decisão são sensíveis a atributos irrelevantes, mas não são sensíveis aos atributos redundantes;
- O algoritmo *Naive Bayes* não apresenta sensibilidade aos atributos irrelevantes, uma vez que os mesmos não afetam o cálculo geral da probabilidade posterior $P(Y|X)$.

5. CONCLUSÃO

Este trabalho introduziu uma discussão sobre dois classificadores o *Naive Bayes*, provavelmente é a implementação mais utilizada dos classificadores bayesianos na tarefa de classificação e a Árvore de Decisão. Os algoritmos de classificação de Árvore de Decisão utilizados neste trabalho foram o ID3 e C4.5, mas existem muitos outros importantes como: Hunt, CART, CHAID e See 5 entre outros. A preferência pelo ID3 e C4.5 se justifica pelo fato de serem bastante aceitos no meio acadêmico e científico e frequentemente serem usados como padrão de comparação em relação a outros algoritmos. A discussão gira em torno da escolha dos atributos mais importantes do conjunto de dados, dos atributos irrelevantes e redundantes. Entenda por mais importante, aqueles atributos que fazem uma melhor discriminação do conjunto de treinamento, ou seja, separa de maneira mais precisa os exemplos positivos e negativos nas suas respectivas classes.

Os atributos mais importantes para os algoritmos de Árvore de Decisão, são escolhidos com base na entropia e no ganho de informação que são as métricas utilizadas pelos algoritmos ID3 e C4.5 citados neste trabalho. Por sua vez o *Naive Bayes* utiliza todos os atributos da base de dados, ou seja, não existe um atributo de maior importância do que outro, uma vez que todos são utilizados pelo algoritmo durante a construção do modelo de aprendizagem. Isso leva a outro ponto importante, o *Naive Bayes* ao utilizar todos os atributos da base de dados, consequentemente utiliza também os atributos irrelevantes e os redundantes. O *Naive Bayes* é robusto a atributos irrelevantes, isso por que eles não causam impactos no cálculo da probabilidade condicional de classe. Para exemplificar essa situação, no cálculo da probabilidade condicional (tabela 4.2) se os valores $P(\text{Temperatura} = \text{quente} \mid \text{Não}) = 2/5$ e $P(\text{Temperatura} = \text{quente} \mid \text{Sim}) = 2/9$ fossem retirados do cálculo da probabilidade posterior, ainda sim a instância citada no exemplo seria classificada como Jogar = Não.

Em contrapartida a Árvore de Decisão é sensível a atributos irrelevantes. Tal fato fica evidente, pois o atributo temperatura não foi utilizado na construção do modelo de aprendizagem, o que fica evidente na figura 4.2 que demonstra a Árvore de Decisão induzida a partir do conjunto de dados da tabela 4.1. Já os atributos redundantes não afetam adversamente a precisão dos algoritmos de Árvore de

Decisão, pois apenas um dos atributos redundantes será selecionado para a divisão do conjunto de dados, o outro será descartado. A suposição de independência do *Naive Bayes*; o quer dizer que o valor de um atributo não influencia no valor de outro atributo, dada a informação da classe; funciona bem para os atributos irrelevantes, mas não para os atributos redundantes, já que os atributos redundantes são multiplicados.

Uma vez que o objetivo é a descoberta de conhecimento a partir de dados, a presença de atributos redundantes pode interferir na precisão do modelo induzido pelo *Naive Bayes* assim como os atributos irrelevantes podem influenciar nos resultados dos algoritmos de Árvore de Decisão. Sendo assim, a presença de atributos redundantes e irrelevantes é indesejável em uma base de dados.

Este trabalho deixa sua contribuição, tendo em vista que promove uma discussão sobre importantes diferenças entre dois tipos de classificadores bem aceitos e que obtém bons resultados em situações adequadas a cada um. Apesar de não ser esse o foco deste trabalho, mas de certa forma, confirma o que foi enunciado no teorema de NFL (*No Free Lunch Theorem*), não existe um algoritmo de classificação que seja superior a todos os outros em qualquer problema de classificação.

Esse trabalho mostrou uma discussão entre duas técnicas de mineração de dados: a Árvore de Decisão e *Naive Bayes*. Existem algumas alternativas que podem ser aplicadas em trabalhos futuros:

- Levando em consideração a presença de atributos irrelevantes e redundantes na base de dados e que existem bons algoritmos de seleção de atributos. Seria interessante comparar a precisão dos modelos construídos pelo *Naive Bayes*, a partir do subconjunto de atributos selecionados pelo algoritmo de seleção com os modelos construídos a partir do conjunto original.
- Propõe-se realizar uma discussão nos moldes da vista neste trabalho. Comparando o *Naive Bayes* com outros algoritmos de classificação que não utilizem a entropia como função de divisão, mas a distância entre as instâncias, como função de divisão do conjunto de dados, para auxiliar na classificação de novos exemplos, assim como o k-NN (*K-Nearest Neighbors*).

- Com relação ao classificador de *Naive Bayes*, seria interessante aprofundar as pesquisas com o objetivo de investigar o problema que é ocasionado quando os exemplos de treinamento não cobrem todos os valores de atributos. Por exemplo, se $P(\text{Temperatura} = \text{quente} / \text{Sim})$ for igual a zero em vez de $2/9$, então um registro com conjunto de atributos $X=(\text{Visual} = \text{Nublado}, \text{Temperatura} = \text{Quente}, \text{Umidade} = \text{Alta}, \text{Vento} = \text{Fraco})$ possui as seguintes probabilidades:

$$P(X | \text{Não}) = 0 \times 2/5 \times 4/5 \times 2/5 = 0.$$

$$P(X | \text{Sim}) = 4/9 \times 0 \times 3/9 \times 6/9 = 0.$$

Logo o classificador de *Naive Bayes* não conseguirá classificar o registro.

- Os métodos para avaliar o desempenho de um classificador e evitar os problemas de *overfitting* e *underfitting* como o *Holdout*, Validação Cruzada e *Bootstrap* não foram apresentados neste trabalho, por isso fica como uma sugestão para trabalho futuro.

- Outro tema importante e que também não foi abordado neste trabalho e que daria uma boa discussão é a prevalência de classes. Quando uma classe existe em quantidade bem superior a outra, os modelos de classificação induzidos a partir desses dados são tendenciosos as classes majoritárias. Para contornar esse problema é utilizado o conceito de Matriz de Custo. Em uma Matriz de Custo, o peso do erro associado aos registros cujas classes sejam menos numerosas é maior.

REFERÊNCIAS

- BRAZDIL, P. B.; SOARES, C.; COSTA, J. P. D. (2003) Ranking learning algorithms: Using ibl and meta-learning on accuracy and time results. Machine Learning.
- CARVALHO, A. P. de L. F. de. Redes neurais artificiais. Disponível em: <<http://www.icmc.sc.usp.br/~andre/neural1.html>>. Acessado em: 20 de novembro de 2015.
- DWBRASIL. Construindo um data warehouse e analisando suas informações com Data Mining e OLAP, 2002. Disponível em: <<http://www.dwbrasil.com.br/html/dmining.html>>. Acessado em: 12 de dezembro de 2015.
- FAYYAD, U; PIATETSKY-SHAPIO, G; SMYTH, P. From Data Mining to Knowledge Discovery in Databases. American Association for Artificial Intelligence, 1996.
- GAMA, J. Árvores de Decisão, 2000. Disponível em: <<http://www.liaad.up.pt/area/jgama//Mestrado/ECD1/Arvores.html>>. Acessado em: 5 maio de 2016.
- GARCIA, S. C. O uso de Árvores de Decisão na descoberta de conhecimento na Área da Saúde. SEMANA ACADÊMICA. Universidade Federal do Rio Grande do Sul, 2000.
- GOLDSCHMIDT, R.; PASSOS, E. Data Mining: um guia prático. Editora Campus, Rio de Janeiro: Elsevier. 2005.
- GUARDA, A. Inteligência Artificial em Controle de Automação. Editora Edgard Blücher, 2000.
- HAN, J.; KAMBER, M. Data Mining: Concepts and Techniques. 2ª. ed. San Francisco: Morgan Kaufmann, 2006.
- KIRA, K.; RENDELL, L. A., The Feature Selection Problem: Traditional Methods and a New Algorithm, In: Proc. 10th Conference on Artificial Intelligence, Menlo Park, CA, 1992.
- OSÓRIO, F. Sistemas Adaptativos Inteligentes - Indução de Árvores de Decisão. Disponível em: <<http://www.inf.unisinos.br/~osorio/sadi.html>>. Acessado em: 05 de setembro de 2015.
- QUINLAN, J. R. C4.5: Programs for machine learning. Morgan Kaufmann, 1993.

QUINLAN, J. R. Induction of Decision Trees. Machine Learning, 1986.

SOUZA, Erick Nilsen Pereira De. Explorer Fuzzy Tree – uma ferramenta para experimentação de técnicas de classificação baseadas em árvores de decisão fuzzy. Universidade Federal da Bahia, 2007.

STUART, Russel; PETER, Norving - Inteligência Artificial. Tradução: Publicare Consultoria 2 ed. Rio de Janeiro – RJ: Elsevier, 2004.

TAN, P.-N., STEINBACH, M. e KUMAR, V. Introduction to Data Mining. Pearson Addison-Wesley, 2006.

TAN, P-N; STEINBACH, M.; KUMAR, V. Introdução ao Data Mining. Ciência Moderna, 2009.

WAIKATO, U. O. WEKA. Disponível em: < <http://www.cs.waikato.ac.nz/ml/weka/>>. Acessado em janeiro de 2016.

WIKIPEDIA. Thomas bayes. Disponível em: <http://en.wikipedia.org/wiki/Thomas_Bayes>. Acessado em: abril de 2016.