ESTUDO COMPARATIVO DOS ALGORITMOS DE CLASSIFICAÇÃO DE MINERAÇÃO DE DADOS J48, ONER E NAÏVE BAYES

Charles Brito Neres Júnior[1], Napoleão Povoa Ribeiro Filho[2]

[1]Estudante do Curso Superior de Tecnologia em Sistemas para Internet – IFTO. e-mail: <charlesbritoneresjr@gmail.com>
[2]Professor mestre em modelagem computacional de sistemas pela Universidade Federal do Tocantins - UFT. e-mail: <napoleao@ifto.edu.br>

Resumo: Quando há um conjunto de dados em um banco de dados, é possível obter conhecimento chamado de trivial, utilizando os recursos de um Sistema de Gerenciamento de Banco de Dados (SGBD). Porém, é possível que exista conhecimentos que não possam ser obtidos de forma trivial. Para tentar descobrir esse tipo de conhecimento, utiliza-se comumente um conjunto de técnicas chamado mineração de dados. As técnicas de mineração de dados podem ser divididas em: classificação, regressão, agrupamento e associação. O objetivo desse trabalho é apresentar os conceitos relacionados à mineração de dados e também apresentar um estudo comparativo entre 3 algoritmos de classificação, utilizando para isso o software Weka.

Palavras–chave: classificação, kdd, mineração de dados, weka

1 INTRODUÇÃO

Estar conectado e com acesso a maior parte do tempo a internet permite uma criação enorme de dados diariamente pelo usuário. Redes sociais como Instagram, WhatsApp, Facebook são exemplos de ferramentas que utilizamos no dia a dia que permitem a criação de dados o tempo todo através de postagens e downloads.

Essa quantidade enorme de armazenamento representa num primeiro momento dados que não produzem nenhuma informação relevante já que não foram ainda explorados. A partir do momento que esses dados estão armazenados em um SGBD (Sistema de Gerenciamento de Banco de Dados) é possível extrair informações a partir de consultas feitas (conhecidas como querys), utilizando uma linguagem específica para isso (SQL – Structured Query Language). Os dados obtidos a partir desse tipo de consulta são chamados de triviais. Mas é possível que exista mais informação em uma base de dados, mas que não possam ser obtidas a partir de consultas SQL. Nesse sentido, a mineração de dados surge como um recurso, ou conjunto de recursos, que permite tentar descobrir novos conhecimentos (não triviais) em uma base de dados.

Minerar dados é um recurso que pode ser utilizado em qualquer área. Uma empresa, por exemplo, pode utilizá-los para aumentar o número de vendas de um determinado produto, uma concessionária pode descobrir qual o perfil de cliente que compra um modelo específico de carro, um banco pode utilizar a mineração de dados para verificar se aprova ou não um empréstimo solicitado por um cliente.

Nesse contexto, esse artigo tem como finalidade apresentar os principais conceitos de mineração de dados definindo brevemente as tarefas que a integram e, além disso, apresentar um

ISSN 2179-5649
X JICE©2019

estudo comparativo entre três algoritmos que implementam a tarefa de classificação, utilizando para isso o software Weka.

2 METODOLOGIA

O artigo classifica-se como uma pesquisa experimental, uma vez que ao longo de sua construção os resultados foram obtidos através da utilização de testes que observaram o comportamento dos algoritmos com melhor desempenho. A pesquisa experimental consiste em determinar um objeto de estudo, selecionar as variáveis que seriam capazes de influenciá-lo, definir as formas de controle e de observação dos efeitos que a variável produz no objeto (Gil, 2007).

A base de dados utilizada neste estudo para realização dos testes foi obtida a partir da url http://prdownloads.sourceforge.net/weka/datasets-numeric.jar, de um arquivo chamado credit-g.arff. A base selecionada descreve um conjunto de dados relacionados à liberação de crédito financeiro da Universität Hamburg (Universidade de Hamburgo) realizada pelo professor Dr. Hans Hofmann.

As ferramentas utilizadas foram Weka versão 3.8.1, sistema operacional Windows 10 Pro, o editor de LaTeX online Overleaf e o processador de textos LibreOffice Writer versão 6.2.5.2 (x64).

A análise dos dados foi realizada utilizando os algoritmos selecionados em dois tipos de testes. O primeiro tipo é o Cross-validation, opção que deixa o Weka, software para mineração de dados, construir um modelo baseado em subconjuntos dos dados fornecidos e então calcular sua média para criar um modelo final, dividida em 5, 10 e 20 folds (conjuntos). O segundo tipo é o Percentage split, onde o software Weka toma um subconjunto percentual dos dados fornecidos para construir um modelo final.

Para tentar determinar o algoritmo de melhor desempenho, foram observados o percentual de instâncias classificadas corretamente, tomando como referência o rendimento do algoritmo de referência OneR, além de observar a matriz de confusão de cada teste realizado e suas respectivas taxas de real positivo e falso positivo.

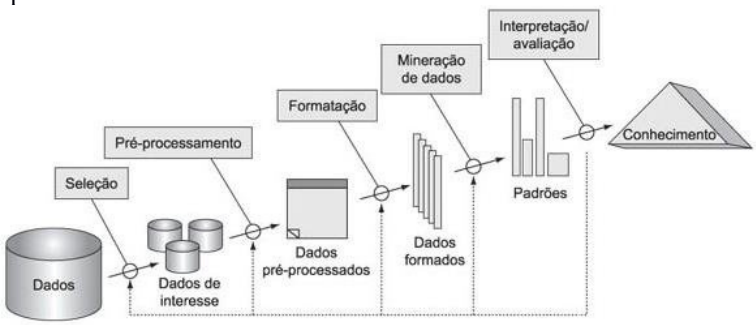
3 DESCOBERTA DE CONHECIMENTO EM BASE DE DADOS (KDD)

3.1 Definição

KDD consiste num processo de várias etapas, não triviais, interativas e iterativas, que visam extrair informações implícitas de uma base, na qual simples pesquisas em bancos de dados não seriam capazes de encontrar. Ele é um processo abrangente que resulta na descoberta de conhecimentos úteis,

a partir de referências brutas, além de identificar padrões compreensíveis, válidos e reutilizáveis (Amaral, 2016). Os processos do KDD podem ser observados na Figura 1:

Figura 1 –- Etapas do processo KDD



Fonte: (Fayyad et al., 1996).

A etapa de seleção é a fase que dá início ao processo de descoberta de conhecimento e pode ter sua origem em lugares como banco de dados, planilhas eletrônicas, diários escolares, entre outros. O pré-processamento é a fase responsável pela limpeza dos dados. Nela serão retirados ruídos, campos vazios, dados incoerentes, incorretos, além de ser realizado o preenchimento ou remoção de valores nulos e exclusão de registros duplicados (De Medeiros e Padilha, 2018).

A etapa de formatação ou transformação exige que os dados tenham sido modelados corretamente na fase anterior e consiste basicamente na mudança da estrutura dos dados para um formato compreendido pelos algoritmos de mineração. Mineração de Dados é uma subcategoria de um processo mais amplo que é o KDD. Nesta etapa será utilizado o método e algoritmo escolhido para explorar e analisar o volume de dados em busca de padrões (Macedo, 2010).

Por fim, a etapa de interpretação ou avaliação consiste na interpretação dos resultados obtidos e na validação do conhecimento adquirido através da submissão do produto à avaliação de um especialista no assunto (Macedo, 2010).

3.2 Mineração de Dados

Como brevemente citada na seção anterior, mineração de dados compõe uma etapa do processo de Descoberta de Conhecimento em Base de Dados (KDD), onde a exploração dos dados é de fato realizada. Porém, antes de falarmos sobre a forma como a mineração de dados é feita, devemos primeiramente entender com mais detalhes o que ela é, suas funcionalidades e benefícios.

Mineração de dados é um processo que explora e analisa grandes volumes de dados em busca de padrões, previsões, erros, associações (Amaral, 2016). Ela submete dados levantados anteriormente a algoritmos desenvolvidos com a capacidade de fazer com que computadores produzam novos conhecimentos baseados em dados antigos. Uma simples consulta ao banco de dados ou a uma

planilha eletrônica dificilmente produziria resultados semelhantes aos obtidos pela mineração de dados, já que utiliza técnicas e algoritmos programados para extrair informações relevantes antes dispersas e desconhecidas.

Mineração de dados possui tarefas, as quais contêm vários algoritmos que processam e transformam dados com o propósito final de produzir um modelo que melhore a forma como vemos e interpretamos nossos dados no presente e numa perspectiva futura. Para que isso aconteça, a determinação da tarefa adequada é um ponto imprescindível para a obtenção de um resultado positivo.

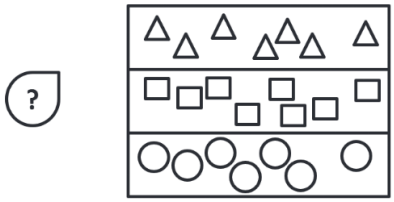
3.3 Tarefas de Mineração de Dados

A obtenção de um bom resultado na mineração de dados está relacionada a aplicação da tarefa adequada para seu modelo de dados. Portanto, mesmo com uma grande massa de dados em mãos, a escolha da técnica inadequada para as características do projeto tornaria os dados da base em informações inúteis.

Segundo (Amaral, 2016), as tarefas podem ser divididas em quatro grandes grupos: classificação, regressão, agrupamentos e regras de associação.

Na classificação, temos o surgimento do atributo chamado classe ou atributo-classe que pode ocorrer uma ou mais vezes na base de dados. O objetivo dessa tarefa será tentar prever ou descrever as características da classe ao final do trabalho. Normalmente ela estará na última posição do seu conjunto de dados e o Weka considerará isto, mas, caso não esteja, será possível informar o lugar que estiver. Consideraremos a Figura 2 para exemplificarmos o funcionamento básico da tarefa de classificação:

Figura 2 –- Classificação



Fonte: (Amaral, 2016).

A Figura 2 possui três retângulos sobrepostos e uma forma externa com uma interrogação centralizada. Observe que a forma externa que possui uma interrogação no centro não foi classificada em nenhuma das três, portanto não teve seu tipo encontrado, ela é nosso atributo especial, é a classe que queremos identificar. O algoritmo analisará as características que definem cada forma já conhecida e classificará a forma externa desconhecida em uma das três categorias.

A regressão pode ser comparada a classificação e também tentará prever um atributo-classe ao final do trabalho, porém enquanto na classificação a classe é composta por um dado nominal, ou seja, um nome, descrição ou categoria, na regressão esse valor de saída será numérico. O preço de venda de uma casa é um exemplo da tarefa de regressão. Atributos como tamanho da casa, quantidade de cômodos, metragem do terreno, vagas de garagem, entre outros serão utilizados para produzir o modelo que informará o preço de venda do imóvel.

Em Clustering ou Agrupamento, não existirá o atributo classe e essa será uma das principais diferenças em relação as tarefas vistas anteriormente. O intuito dessa tarefa é a criação de grupos segundo o grau de semelhança entre suas instâncias. Os grupos serão criados analisando as semelhanças entre os próprios dados. Com isso, dependendo do tipo de agrupamento utilizado, um elemento poderá pertencer a mais de um, apenas um, ou nenhum grupo. Um elemento não agrupamento será chamado ruído.

Regra de Associação é a tarefa que tenta identificar atividades que ocorrem associadas a outras. Ela buscará encontrar o surgimento de itens e movimentos que acontecem simultânea e frequentemente em uma base de dados. O exemplo clássico da aplicação desse método é a cesta de compras, onde quem comprou um produto A também comprou um produto B (Silva et al., 2010).

3.4 Estudo Experimental

3.4.1 Algoritmos utilizados

OneR, abreviação de “One Rule”, é um algoritmo de classificação simples e objetivo, que produz uma regra para cada atributo nos dados e, em seguida, seleciona a regra com o menor erro total como sua regra única. O algoritmo OneR foi escolhido como base de referência (baseline). A escolha do algoritmo baseline é experimental e optou-se pelo OneR por ser um classificador simples, por utilizar um método de classificação de custo reduzido e obter uma alta acurácia (Manhães et al., 2012).

J48 tem a finalidade de produzir uma árvore de decisão baseada em um conjunto de dados de treinamento, sendo este modelo usado para classificar as instâncias no conjunto de teste. Para a montagem da árvore, o algoritmo J48 decompõe um problema complexo em subproblemas mais simples, aplicando repetidamente a mesma tática a cada subproblema, dividindo o espaço definido pelos atributos em subespaços, associando-se a eles uma classe (Librelotto e Mozzaquatro, 2014).

Naïve Bayes é um dos mais simples classificadores probabilísticos. As probabilidades são estimadas pela contagem da frequência de cada valor de característica para as instâncias dos dados de treino. Dada uma nova instância, o classificador estima a probabilidade de essa instância pertencer a

uma classe específica, baseada no produto das probabilidades condicionais individuas pra os valores característicos da instância (Gomes, 2019).

4 RESULTADOS E DISCUSSÕES

Os resultados alcançados, em relação as instâncias classificadas corretamente para o algoritmo OneR foram de 68%, 71.6667%, 68.5% utilizando o teste do tipo Percentage split configurado com 60%/40%, 70%/30%, 80%/20% respectivamente. Onde o primeiro valor é referente ao percentual de instâncias utilizadas para treinar o algoritmo e o segundo valor referente ao percentual utilizado para testar o algoritmo já treinado. Para o algoritmo J48, utilizando as mesmas condições de análise, os resultados obtidos foram 71.25%, 73.6667% e 77%. Já para o algoritmo Naïve Bayes, também utilizando as mesmas condições de análise, os resultados alcançados foram 77%, 75.3333%, 74.5% conforme mostra a Figura 3:

Figura 3 –- Percentage Split

Percentage Split	OneR		J48		NaiveBayes	
	Acerto	Tempo	Acerto	Tempo	Acerto	Tempo
60% / 40%	68,00%	0.03 s	71.25%	0.03 s	77,00%	0.03 s
70% / 30%	71.6667%	0 s	73.6667%	0 s	75.3333%	0.01 s
80% / 20%	68.5%	0.01 s	77,00%	0.01 s	74.5%	0.02 s

Com referência ao teste do tipo Cross-validation, a porcentagem de instâncias classificadas corretamente para o algoritmo OneR foram 67.3%, 66.1%, 65.3% para 5, 10 e 20 folds respectivamente. Para o algoritmo J48, utilizando as mesmas condições de avaliação, os resultados foram 73.3%, 70.5%, 69.8%. Por fim para o algoritmo Naïve Bayes os resultados obtidos nas mesmas circunstâncias dos resultados analisados anteriormente foram 75.1%, 75.4%, 75.5% conforme mostra a Figura 4:

Figura 4 –- Cross-Validation

Cross-Validation	OneR		J48		NaiveBayes	
	Acerto	Tempo	Acerto	Tempo	Acerto	Tempo
Folds 5	67.3%	0.01 s	73.3%	0.02 s	75.1%	0 s
Folds 10	66.1%	0.01 s	70.5%	0.02 s	75.4%	0 s
Folds 20	65.3%	0 s	69.8%	0.02 s	75.5%	0 s

A matriz de confusão que é uma tabela que mostra as frequências de classificação para cada classe do modelo, onde as instâncias classificadas corretamente aparecem na diagonal principal e os demais valores fora da diagonal principal são instâncias classificadas incorretamente, apresentou os resultados esquematizados, em função do Percentage split, composta pela classe positiva, bom

pagador, e negativa, maus pagadores de crédito com as taxas de acerto e erro dos classificadores, conforme observado na Figura 5:

Figura 5 –- Classificadores

PS	C	OneR	J48	Naive Bayes	PS	C	OneR	J48	Naive Bayes	PS	C	OneR	J48	Naive Bayes
60%/40%	MC	247 46 82 25	243 50 65 42	245 48 44 63	70%/30%	MC	201 20 65 14	192 29 50 29	186 35 39 40	80%/20%	MC	127 22 41 10	127 22 24 27	119 30 21 30
	VP	0,843	0,829	0,836		VP	0,91	0,869	0,842		VP	0,852	0,852	0,799
	FN	0,157	0,171	0,164		FN	0,09	0,131	0,158		FN	0,148	0,148	0,201
	VN	0,234	0,393	0,589		VN	0,177	0,367	0,506		VN	0,196	0,529	0,588
	FP	0,766	0,607	0,411		FP	0,823	0,633	0,494		FP	0,804	0,471	0,412

PS = Persentage split; C = Classificador; MC = Matriz de Confusão; VP = Verdadeiro Positivo; FN = Falso Negativo; VN = Verdadeiro Negativo; FP = Falso Positivo.

4.1 Análise de resultados

A análise dos resultados da opção Percentage split visando encontrar os maiores percentuais de acertos, apresentaram valores dos algoritmos J48 e Naïve Bayes bem melhores que o baseline OneR. No entanto, o algoritmo J48 com o Percentage split de 80%/20% apresentou o maior percentual de acerto com um bom tempo de resposta de 0.01 segundos.

Na matriz de confusão observamos também que a porcentagem de instâncias classificadas incorretamente para o split de 80%/20% referente ao algoritmo J48 apresentou o mais baixo percentual, que implica dizer que existem poucas instâncias classificadas de forma errada e consequentemente mais instâncias classificadas acertadamente.

Por outro lado, analisando os resultados da opção Cross-validation, percebemos que dentre os percentuais de acerto, nenhum deles alcança a taxa atingida pela opção Percentage-split de 77%. Entre os resultados com melhor desempenho estão os percentuais de 75.1%, 75.4% e 75.5% todos atingidos com o algoritmo Naïve Bayes.

Logo, considerando a análise dos resultados acima apresentados, concluímos que o algoritmo J48 através da opção Percentage split numa proporção de 80%/20% apresentou o melhor desempenho dentre os escolhidos. Também foi quem mais se destacou em comparação aos resultados obtidos pelo algoritmo baseline OneR, evitando o problema chamado superajuste, que ocorre quando o modelo é criado perfeitamente para um conjunto de dados, mas somente para esses dados, garantindo com isso, que o modelo criado preverá, de maneira exata, valores futuros desconhecidos.

5 CONSIDERAÇÕES FINAIS

Este artigo procurou explicar o que é a descoberta de conhecimento em base de dados (KDD), quais etapas compõe esse processo, evidenciando a mineração de dados como o etapa principal de todo o procedimento. Apresentou também a ferramenta de mineração de dados de código aberto Weka como software utilizado para realização de testes.

Além disso, foram explicadas as tarefas que compõem a mineração de dados, com destaque para a tarefa de classificação, escolhida para a construção do trabalho. Conceituou-se os três algoritmos de classificação selecionados, OneR, J48 e Naïve Bayes. Testes foram feitos, utilizando uma mesma base de dados para todos os algoritmos. Os resultados obtidos foram interpretados e comparados, no intuito de identificar o melhor classificador para a base de dados utilizada.

REFERÊNCIAS

AMARAL, F. (2016). **Aprenda Mineração de Dados – Teoria e Prática.** Alta Books, 1th edition.

DE MEDEIROS, L. B. G. AND PADILHA, T. P. P. (2018). **Mineração de dados para detectar evasão escolar utilizando algoritmos de classificação.** CIET: EnPED.

FAYYAD, U. M., PIATETSKY-SHAPIRO, G., SMYTH, P., ET AL. (1996). **Knowledge discovery and data mining: Towards a unifying framework**. In KDD, volume 96, pages 82–88.

FRANÇA, R. S. AND AMARAL, H. J. C. (2013). **Mineração de dados na identificação de grupos de estudantes com dificuldades de aprendizagem no ensino de programação**. RENOTE, 11(1).

LIBRELOTTO, S. R. AND MOZZAQUATRO, P. M. (2014). **Análise dos algoritmos de mineração j48 e apriori aplicados na detecção de indicadores da qualidade de vida e saúde**. Revista Interdisciplinar de Ensino, Pesquisa e Extensão, 1(1).

MACEDO, DAYANA CARLA E MATOS, S. N. (2010). **Extração de conhecimento através da mineração de dados**. Revista de Engenharia e Tecnologia, 2(2):Páginas–22.

MANHÃES, L. M. B., DA CRUZ, S. M. S., COSTA, R. J. M., ZAVALETA, J., AND ZIMBRÃO, G. (2012). **Previsão de estudantes com risco de evasão utilizando técnicas de mineração de dados**. In Brazilian symposium on computers in education (simpósio brasileiro de informática na educação-sbie), volume 1.

SILVA, C. F., RODRIGUES, C. T., AND MONTEIRO, M. V. B. (2010). **Inteligência artificial-uso de regras de associação para descoberta de conhecimento na produtividade de açaí no estado do Amapá**. Anais SULCOMP, 5.

GOMES, Pedro César Tebaldi. **Classificação com Naive Bayes.** Disponível em: https://www.datageeks.com.br/naive-bayes/. Acesso em: 25 set. 2019.