

AVALIAÇÃO DO MÉTODO LÓGICA CLÁSSICA NA BASE DE DADOS MONK'S PROBLEMS (MONK-3)

Vitória Polliany de Oliveira Silva ¹
Lucilla Vieira Carneiro ²
Ronei Marcos de Moraes ³

RESUMO

O processo de tomada de decisão consiste na escolha de uma ação dentre várias possíveis, no intuito de prevenir ou solucionar problemas. A lógica clássica é um método de tomada de decisão bastante eficiente em eventos finitos e determinísticos. É desenvolvida computacionalmente com regras pré-estabelecidas e pode ser aplicada em bancos de dados onde as informações relativas ao problema apresentam-se agrupadas em classes. Este artigo tem como objetivo avaliar se o método de decisão lógica clássica se adequa a base de dados MONK's Problems (MONK-3). Trata-se de um estudo descritivo e exploratório, com abordagem quantitativa. O software weka foi utilizado para aplicação do método lógica clássica sobre o banco de dados selecionado. Para testes neste estudo foi utilizado o algoritmo JRip. Os dados foram testados pelos parâmetros cross-validation e percentagem split. Além disso, para analisar os resultados apresentados foi utilizado o coeficiente Kappa. De acordo com os testes realizados, o melhor resultado foi obtido utilizando o parâmetro cross-validation com 30 folds, onde se alcançou o valor Kappa de 0,9954 e 99,7685% de precisão de acertos na matriz de confusão. Conclui-se que o método lógica clássica se mostrou adequado ao conjunto de dados.

Palavras-chave: Lógica Clássica, Algoritmo JRip, Base de dados Monk's Problems.

INTRODUÇÃO

O processo de tomada de decisão consiste na escolha de uma ação dentre várias possíveis, no intuito de prevenir ou solucionar problemas. Formalmente, as decisões podem ser classificadas quanto a natureza em programadas ou não programadas. As decisões programadas se repetem com frequência e são determinadas por regras.

¹ Mestranda do Curso de Modelos de Decisão e Saúde, da Universidade Federal da Paraíba - UFPB, vitoriapolliany1@gmail.com;

² Doutoranda do Curso de Modelos de Decisão e Saúde, da Universidade Federal da Paraíba - UFPB, lucilla.vc@hotmail.com;

³ Professor orientador: Doutor em Computação Aplicada, Docente da Graduação e Pós-graduação no Centro de Ciências Exatas e da Natureza, da Universidade Federal da Paraíba - UFPB, ronei@de.ufpb.br;

Ademais, são tomadas sob condições de certeza ou pouca incerteza. Enquanto que as decisões não programadas, se referem a situações não previstas e são baseadas em incertezas (PINTO, 2018).

As decisões programadas estão fundamentadas na teoria da lógica clássica, conhecida anteriormente como lógica aristotélica tradicional. A referida lógica proposta por Aristóteles, é fundamentada em três princípios: Princípio da identidade - todo objeto é idêntico a si próprio; Princípio da não contradição - uma proposição não pode ser verdadeira e falsa sob a mesma perspectiva; Princípio do terceiro excluído - uma proposição é verdadeira ou falsa, não havendo outra valoração (PINTO, 2018).

A lógica clássica representa uma forma simples de lógica e possibilita a elaboração de regras simples. Somado a isso, também admite a construção de regras compostas por outras regras simples, desde que não ocorra contradição. Dessa maneira, pode ser utilizada para construir regras do tipo “SE condição ENTÃO consequência SENÃO consequência”. Somado a isso, na construção dessas regras alguns operadores booleanos (AND, OR, NOT) também são utilizados dentro desse modelo (MOURA, 2017; PINTO, 2018).

É importante ressaltar, que dentre as principais vantagens da lógica clássica destaca-se a sua fácil implementação, embora exija um bom planejamento. Além disso, o seu diferencial é que apresenta eficiência em eventos finitos e determinísticos (MACHADO; VALENÇA; MORAIS, 2016). Todavia, em algumas situações essa técnica é limitada, já que não permite classificar informações imprecisas. Ademais, pode propiciar a ocorrência de conclusões aparentemente lógicas, que não são verdadeiras. Nesse ínterim, em certas ocasiões impossibilita descrever a realidade total utilizando apenas o falso e o verdadeiro.

Entretanto, esse método também é bastante eficaz e vem sendo utilizado no processo de tomada de decisão em diversas áreas, como na computação e na saúde. A sua utilização é bastante precisa em sistemas onde o fluxo de acontecimentos é bem definido, como é o caso do *software* “SADEAACC”, voltado para o atendimento de crianças e adolescentes em situação de adoecimento crônico (PINTO, 2018). Assim como, do *serious game* “Uma Aventura na Floresta da Dentolândia”, direcionado ao aprendizado de saúde bucal em bebês (MACHADO; VALENÇA; MORAIS, 2016).

Nesse contexto, pode-se afirmar que a lógica clássica é desenvolvida computacionalmente com regras pré-estabelecidas e representa um importante modelo

de apoio à tomada de decisão. Com isso, pode ser aplicada em bancos de dados onde as informações relativas ao problema apresentam-se agrupadas em classes (PINTO, 2018).

Nesse ínterim, algumas bases de dados disponíveis no UCI Machine Learning Repositor apresentam essa característica, a exemplo da MONK's Problems, cujos dados encontram-se distribuídos em duas classes (0 ou 1).

O conjunto de dados da MONK's Problems foi doado em 1992 por Sebastian Thrun e serviu como base para a primeira comparação internacional de algoritmos de aprendizagem. Essa comparação foi desenvolvida por uma coleção de pesquisadores e cada um foi defensor da técnica que testou. O referido banco de dados é constituído por três problemas sob o mesmo domínio: MONK-1, MONK-2 e MONK-3 (TSYMBAL; PUURONEN; TERZIYAN, 1999).

Todavia, embora seja difícil determinar qual algoritmo é mais adequado para um conjunto de dados específico, através do software Weka é possível ter uma interface de usuário uniforme para um grande número de algoritmos de aprendizado de máquina, fazendo com que se tenha maior precisão nos resultados (SMITH; FRANK, 2016).

Diante do exposto, o estudo tem como objetivo avaliar se o método de decisão Lógica Clássica se adequa a base de dados MONK's Problems (MONK-3), através de testes realizados no software Weka.

METODOLOGIA

Trata-se de um estudo descritivo e exploratório, com abordagem quantitativa, onde foi realizada a aplicação do método Lógica Clássica sobre os dados fornecidos pelo banco de dados MONK's Problems (MONK-3), disponível em um repositório internacional que pode ser acessado por meio do link: <https://archive.ics.uci.edu/ml/datasets/MONK's+Problems>. Este banco de dados possui 432 instâncias, 7 atributos e 2 classes, conforme demonstrado na Tabela 1.

Tabela 1. Atributos da Base de Dados

ATRIBUTOS DA BASE DE DADOS MONK's Problems (MONK-3)

1. class: 0, 1
2. a1: 1, 2, 3
3. a2: 1, 2, 3
4. a3: 1, 2
5. a4: 1, 2, 3
6. a5: 1, 2, 3, 4
7. a6: 1, 2
8. Id

Fonte: Base de Dados MONK's Problems, 1992.

Segundo Paiva (2014), na lógica clássica só é permitido que um elemento pertença ou não a um dado conjunto. Sendo assim, na concepção da Teoria Clássica dos Conjuntos, a função de pertinência de um conjunto clássico A contido em um universo U é dada por:

$$\mu_A(x) = \begin{cases} 1 & \text{se } x \in A \\ 0 & \text{se } x \notin A \end{cases}$$

O software utilizado para aplicação do método lógica clássica sobre o banco de dados selecionado, foi o Weka (Waikato Environment for Knowledge Analysis), versão 3.8.5. Trata-se de um pacote de mineração de dados e apresenta diversos algoritmos de modelo de decisão. Cada algoritmo possui padrões pré-definidos e são executados conforme a filosofia de aprendizagem de máquina. Foi desenvolvido na Universidade de Waikato na Nova Zelândia e escrito em linguagem JAVA. Além disso, é considerado um software de referência e apresenta acesso gratuito (BOUCKAERT *et al.*, 2016).

Geralmente, no software Weka os conjuntos de regras e de árvores de decisão são aprendidos mais rapidamente, podendo fornecer representações mais compactas e inteligíveis quando comparados aos demais métodos disponíveis (SMITH, FRANK, 2016).

Dessa maneira, dentre os diversos algoritmos utilizados durante o processo de tomada de decisão, destaca-se o Repeated Incremental Pruning to Produce Error Reduction (RIPPER), também conhecido como JRip. Trata-se de um modelo baseado

em regras proposicionais ou condicionais e bastante aplicado em diversas áreas, tais quais: bioinformática, ciência, economia, biologia e saúde (OLEGÁRIO *et al.*, 2020).

O algoritmo JRip se divide em duas fases. Na primeira fase, é gerado um conjunto de regras para a comparação. Enquanto que na segunda fase, o conjunto de regras iniciais é otimizado para diminuir erros e tornar o processo mais seletivo. Esses passos são repetidos diversas vezes no software WEKA (OLEGÁRIO *et al.*, 2020).

Contudo, para testes neste estudo foi utilizado o algoritmo JRip. Os dados foram testados pelos parâmetros cross-validation e percentage split. O parâmetro percentage split faz a utilização de porcentagens definidas pelo usuário. Assim sendo, essa abordagem designa a porcentagem informada de dados para treinamento e o restante é utilizada para testes (BOUCKAERT *et al.*, 2016).

Alguns problemas são evidenciados com essa técnica, tendo em vista que quando a amostra não é grande, torna-se inviável descartar uma boa parte dos dados para testes. Além de que, quando é selecionado apenas um conjunto de dados para testes, a estimativa do erro pode não apresentar ampla veracidade. Assim sendo, essas limitações podem ser supridas através da técnica cross-validation (SMITH; FRANK, 2016).

O parâmetro cross-validation é bastante utilizado para a avaliação de desempenho de modelos de aprendizado de máquina. Nessa abordagem é possível dividir os dados em um número de observações de tamanhos iguais ou dobras (k). Uma dobra dessas (k) é usualmente selecionada como teste e as demais ($k-1$) são usadas para fins de treinamento. Assim, o processo é realizado k -vezes. Esse método é bastante eficaz e evita a aleatoriedade de estimativas produzidas através da divisão dos dados apenas uma vez (KOUL; BECCHIO; CAVALLO, 2018). Diferentemente do percentage split, a grande vantagem dessa técnica é permitir que todos os dados sejam utilizados para treino e para teste em algum momento.

Assim sendo, utilizando o algoritmo JRip foram realizados 10 testes, sendo 5 deles sob o parâmetro cross-validation e mais 5 testes com a técnica percentage split. Em ambos os modelos, os testes do cross-validation foram realizados utilizando 10, 15, 20, 25 e 30 folds. Enquanto que os testes com o parâmetro percentage split, foram realizados empregando os valores: 66%, 70%, 75%, 80% e 85%.

Para analisar o conjunto de dados foi utilizado o coeficiente Kappa, uma medida robusta e ponderada, que é obtida a partir da matriz de confusão. É constituída pelos

acertos e erros das decisões do modelo, onde os acertos ficam dispostos na diagonal principal da matriz e os erros são distribuídos na diagonal secundária. Trata-se de uma medida eficaz e de ampla qualidade, que verifica o comportamento das decisões, testando a sua confiabilidade e precisão (VIEIRA *et al.*, 2018).

Segundo Moraes e Machado (2014), o coeficiente Kappa é calculado por:

$$K = \frac{(P_0 - P_c)}{(1 - P_c)}$$

onde: $P_0 = \sum_{i=1}^M \frac{n_{ii}}{N}$, em que: n_{ii} são elementos da diagonal principal da matriz de confusão; e $P_c = \sum_{i=1}^M \frac{n_{i+}n_{+i}}{N^2}$, em que: n_{i+} é o total da linha i na matriz de confusão, n_{+i} é o total da coluna na mesma matriz, M é o número de decisões possíveis e N é o número total de decisões apresentadas na matriz.

O coeficiente Kappa pode variar entre <0 (discordância total) e 1 (concordância total). De acordo com Landis e Koch (1997), o valor de Kappa $\geq 0,8$ é tido como associação quase perfeita, conforme evidenciado na Tabela 2.

Tabela 2. Interpretação do Coeficiente Kappa

Valor Kappa	Grau de Concordância
$< 0,0$	Pobre
$0,0 - 0,19$	Leve
$0,20 - 0,39$	Bom
$0,40 - 0,59$	Moderado
$0,60 - 0,79$	Considerável
$0,80 - 1,00$	Quase perfeito

Fonte: LANDIS; KOCH, 1977.

RESULTADOS E DISCUSSÃO

Os resultados obtidos a partir dos testes realizados com o algoritmo JRip, foram organizados nas Tabelas 3 e 4 de acordo com o pior e melhor resultado apresentado pelos parâmetros cross-validation e percentage split. Para cada grupo, foi informado o valor Kappa encontrado e a classificação correspondente.

Tabela 3. Resultados dos testes sob o parâmetro cross-validation (JRip).

Folds	Coefficiente Kappa	Classificação do Kappa
10	0,9768	Quase perfeito
15	0,9814	Quase perfeito
20	0,9814	Quase perfeito
25	0,9861	Quase perfeito
30	0,9954	Quase perfeito

Fonte: Dados da pesquisa, 2021.

Tabela 4. Resultados dos testes sob o parâmetro percentage Split (JRip).

Split	Coefficiente Kappa	Classificação do Kappa
66	0,9728	Quase perfeito
70	1	Quase perfeito
75	1	Quase perfeito
80	1	Quase perfeito
85	1	Quase perfeito

Fonte: Dados da pesquisa, 2021.

Observou-se que em todos os testes realizados sob os parâmetros supracitados, o coeficiente Kappa obteve classificação quase perfeito. Ademais, o valor máximo de Kappa foi encontrado no teste percentage split, com os splits 70%, 75%, 80% e 85%. Apesar disso, pode-se afirmar que esses não foram os melhores resultados encontrados, já que esses testes não possibilitam que todos os dados sejam utilizados para treino e para teste.

Mediante a esse cenário, pode-se afirmar que o melhor resultado foi obtido utilizando o parâmetro cross-validation com 30 folds, onde se alcançou o valor Kappa de 0,9954. É importante destacar, que esse parâmetro possibilita treinar e testar o modelo com todos os dados disponíveis, contribuindo para que os resultados sejam mais sólidos.

Na matriz de confusão do melhor resultado (Figura 1), é possível detectar na diagonal principal 99,7685% de precisão de acertos. Desse modo, infere-se que o método é adequado aos dados.

Figura 1. Matriz de confusão utilizando 30 folds (JRip).

```

=== Confusion Matrix ===
      a    b  <-- classified as
204    0  |    a = 0
  1 227  |    b = 1
  
```

Fonte: Dados da pesquisa, 2021.

Com a utilização do algoritmo JRip, 6 regras foram formadas para os testes desenvolvidos. Foi possível identificar que para o melhor resultado encontrado, houve a classificação correta de 227 instâncias para a classe 1 e de 204 instâncias para a classe 0. Além disso, apenas uma instância foi classificada incorretamente, como segue demonstrado na Tabela 5.

Tabela 5. Regras formadas para o parâmetro cross-validation com 30 folds (JRip).

REGRAS	RESULTADOS	ACERTOS	ERROS
SE a2=3 and a4=3	ENTÃO class=0	48	0
SE a5=4	ENTÃO class=0	96	0
SE a2=3 and a4=2	ENTÃO class=0	36	0
SE a2=3 and a5=1	ENTÃO class=0	12	0
SE a2=3 and a5=2	ENTÃO class=0	12	0
	ENTÃO class=1	227	1

Fonte: Dados da pesquisa, 2021.

CONSIDERAÇÕES FINAIS

Mediante os resultados apresentados, conclui-se que o método lógica clássica se mostrou adequado ao conjunto de dados. Foi possível identificar que o algoritmo JRip

tem uma solução satisfatória para classificação dos dados do estudo. Além dos resultados positivos, os testes apresentaram agilidade na execução.

REFERÊNCIAS

BOUCKAERT, R. R. *et al.* **WEKA manual for version 3-8-1**. University of Waikato, New Zealand, 2016.

KOUL, A.; BECCHIO, C.; CAVALLO, A. Cross-validation approaches for replicability in psychology. **Frontiers in Psychology**, v. 9, p. 1117, 2018.

LANDIS, J. R.; KOCH, G. G. The measurement of observer agreement for categorical data. **Biometrics**. v.33, p.159-75, 1977.

MACHADO, L. S.; VALENÇA, A. M. G.; MORAIS, A. M. A serious game for education about oral health in babies. **Tempus Actas de Saúde Coletiva**, v. 10, p. 167-188, 2016.

MORAES, R. M.; MACHADO, L. S. Psychomotor skills assessment in medical training based on virtual reality using a weighted possibilistic approach. **Knowledge-Based Systems**, v. 70, p. 97-102, 2014.

MOURA, I. F. M. S. **Simulação e avaliação de incisões cirúrgicas com realidade virtual**. 2017. Dissertação (Mestrado em Modelos de Decisão e Saúde) - Universidade Federal da Paraíba, João Pessoa, 2017.

OLEGÁRIO, W. K. B. *et al.* Identificação de Arritmias Cardíacas pelo Eletrocardiograma Utilizando JRIP-RIPPER. **International Journal of Development Research**, v. 10, n. 05, p.35732-35735, 2020.

PAIVA, P. V. F. **Um ambiente virtual colaborativo para a educação de equipes cirúrgicas**. 2014. Dissertação (Mestrado em Modelos de Decisão e Saúde) - Universidade Federal da Paraíba, João Pessoa, 2014.

PINTO, M. M. P. S. **Sistema de apoio à decisão espacial para o acesso ao atendimento de crianças/adolescentes em condições crônicas**. 2018. Dissertação (Mestrado em Modelos de Decisão e Saúde) - Universidade Federal da Paraíba, João Pessoa, 2018.

SMITH T.C.; FRANK E. Introducing machine learning concepts with WEKA. **Methods in molecular biology**, p. 353-378, 2016.

TSYMBAL, A.; PUURONEN, S.; TERZIYAN, V. Arbiter meta-learning with dynamic selection of classifiers and its experimental investigation. In: **East European Conference on Advances in Databases and Information Systems**. Springer, Berlin, Heidelberg, p. 205-217, 1999.