

Estudo e implementação de algoritmos de aprendizagem de máquina supervisionado aplicados no Diagnóstico por imagens de Patologias Renais

Pedro Antonio F. da Silva¹, Arthur C. Basílio¹, Vinícius P. Machado¹, Nayze Lucema S. Aldeman²

¹Departamento de Ciência da Computação – Universidade Federal do Piauí (UFPI) – Teresina, PI – Brasil

²Departamento de Medicina Especializada – Universidade Federal do Piauí (UFPI) – Teresina, PI - Brasil

p.antonio.f.s@gmail.com, basilio.arth@gmail.com, vinicius@ufpi.edu.br, nayzealdeman@gmail.com

Abstract: This paper proposes an application of supervised machine learning algorithms to classify diagnoses of renal pathologies. By using of random forest algorithm, two classifiers were created that can label a case among 5 groups of diagnoses, both differ in the use or not of features extracted from images. Because of the smaller number of images compared with the training registers, using only those with a respective image generated a lower accuracy in the result, 77.47%, whereas using the data in its entirety generated 82.26%.

Resumo: Este artigo propõe uma aplicação de algoritmos de aprendizagem de máquina supervisionado para classificar diagnósticos de patologias renais. Através do algoritmo floresta aleatória, foi criado dois classificadores que são capazes de rotular um caso entre 5 grupos de diagnósticos, eles se diferem no uso ou não de atributos extraídos de imagens. Devido à menor quantidade de imagens em relação aos registros de treinamento, usar apenas aqueles com imagens correspondentes gerou uma menor acurácia no resultado, 77.47%, enquanto usar os dados em sua totalidade gerou 82.26%.

1. Introdução

Este trabalho tem como objetivo promover o emprego de uma técnica de aprendizado de máquina supervisionada para ajudar no diagnóstico de patologias renais, tendo em vista o uso de imagens (lâminas) de rins e de descrições de casos feitos por um especialista.

Por meios convencionais, o diagnóstico das patologias renais envolvidas com o projeto em questão é dado através da análise particular de cada imagem por um nefrologista (médico especializado na diagnose e tratamento clínico de doenças renais). A partir dessa análise, o nefrologista obtém informações relevantes sobre o rim do paciente de acordo com as diferentes características apresentadas na imagem. Disponibilizar ferramentas de aprendizado de máquina que auxiliam esses profissionais a construir diagnósticos tem resultados bastantes relevantes para a sociedade.

Neste caso, foi utilizado a tecnologia (especificamente algoritmos de aprendizagem de máquina baseados em árvore de decisão) como forma de auxiliar os

especialistas da área. No desenvolver do trabalho, foi criado um classificador que usa tanto dados clínicos como descritores extraídos de imagens.

2. Referencial teórico

Para este trabalho analisou-se o processo de KDD (do inglês, *Knowledge-Discovery in Databases*) proposto por Fayyad et al. (1996), exposto na Figura 1.

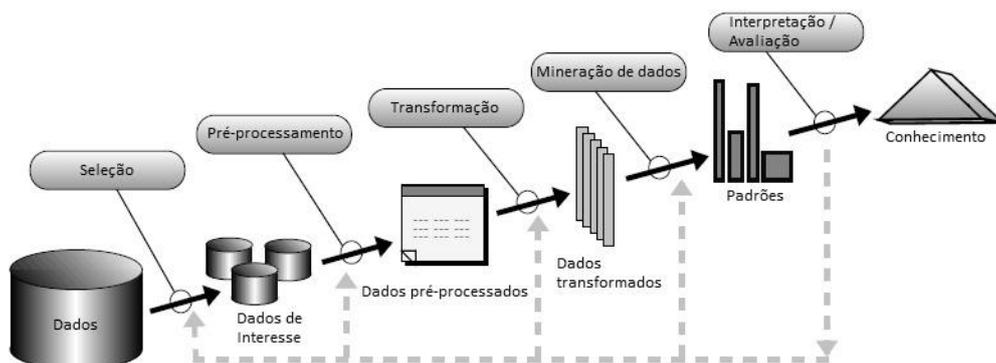


Figura 1. Esquema do processo de KDD proposto por Fayyad et al. (1996).

Segundo Fayyad et al. (1996), KDD é um processo elaborado para extrair conhecimento útil de um conjunto de dados. Nele há cinco etapas, são elas: seleção, pré-processamento, transformação, mineração de dados e avaliação ou interpretação.

Goldschmidt e Passos (2005) definem seleção como uma triagem de informações nas bases de dados obtidas para selecionar aquelas que serão de fato utilizadas nos próximos passos do processo de KDD. Nomes, matrículas, documentos e outros atributos de identificação de registros são exemplos de dados descartados.

Logo após, no pré-processamento, Boente et al. (2008) citam que há um processo de limpeza na base a fim de melhorar a qualidade das informações. Como dito por eles, dados inconsistentes, registros com algum atributo em falta e informações falsas são erros que devem ser corrigidos para não prejudicar o processo de descoberta de conhecimento.

A próxima etapa, transformação, trata-se de rotinas aplicadas aos dados para adequá-los aos algoritmos aplicados futuramente, que exigem formatos específicos de entrada ou que melhor desempenham com dados em uma determinada forma. Tornar valores contínuos em faixas discretas e normalização são exemplos de métodos aplicados.

A próxima etapa, mineração de dados, é descrita por Han et al. (2012) como um processo essencial onde métodos inteligentes são aplicados para extrair padrões de dados.

Ao final, é o momento de definir métodos para avaliar os resultados encontrados na mineração, pode-se, por exemplo, medir a acurácia de uma classificação. Também pode ser necessário interpretar e tratar os resultados para melhor adequá-los à proposta do estudo.

Como citado por Fayyad et al. (1996), esse não é um processo estático, pois muitas vezes são necessárias várias iterações nas etapas ou retroceder passos durante a execução. A Figura 1 apenas destaca o curso básico da metodologia.

Outra parte importante do trabalho foi a extração de características de imagens. Para isso, foi feito um estudo acerca de GLCM (do inglês, Grey-Level Co-occurrence Matrix). Gebejes e Huertas (2013) dizem que os atributos extraídos da GLCM partem do pressuposto que as informações de textura de uma imagem estão associadas à disposição espacial dos níveis de cinza entre *pixels* vizinhos. Segundo esses autores, há informações da frequência de ocorrência de dois níveis de cinza em cada par de *pixels* vizinhos de uma imagem na matriz de coocorrência de níveis de cinza (GLCM).

2.1. Aprendizado de máquina supervisionado

Como descrito por Goldschmidt e Passos (2005), aprendizado supervisionado engloba os modelos que produzem uma saída desejada para cada entrada de dados recebida, ou seja, uma classificação. Em sua aplicação, as entradas sempre são divididas em dois conjuntos, um para treinamento e outro para teste.

2.2. Árvore de decisão e floresta aleatória

Após testes preliminares, o algoritmo floresta aleatória mostrou melhor desempenho, logo foi o método de aprendizado de máquina supervisionado escolhido. Para entender ele, antes é necessário conhecer o algoritmo árvore de decisão.

Uma árvore de decisão é uma estrutura de árvore semelhante a um fluxograma, onde cada nó interno (nó não folha) denota um teste em um atributo, cada ramo representa um resultado do teste, e cada nó folha (ou nó externo) contém um rótulo de classe. O nó mais alto de uma árvore é o nó raiz [HAN; KAMBER; PEI. 2012, p. 330, tradução nossa].

Como descrito por Breiman (2001), floresta aleatória é formada por um conjunto de árvores de decisão. Para cada árvore há um subconjunto de mesmo tamanho e aleatório dos dados de treinamento. Cada árvore gera um voto para uma determinada classificação, assim é escolhida a mais votada.

A implementação da floresta aleatória usada no trabalho foi a disponibilizada pela biblioteca *scikit-learn* chamada *RandomForestClassifier*. A biblioteca utiliza o algoritmo CART (do inglês, *Classification And Regression Tree*) para as árvores, como citado por Géron (2017).

3. Metodologia

O desenvolvimento deste trabalho se baseia na metodologia descrita por Fayyad et al. (1996), contida na Figura 1, na seção 2, como descrito a seguir.

3.1. Seleção

Inicialmente foram fornecidos por uma especialista 100 registros com 30 atributos que caracterizam 32 possíveis diagnósticos, um exemplo de atributo é atrofia tubular assumindo valor presente ou ausente. Também existiam imagens referentes a 70 desses registros.

Para extração de descritores da fonte de imagens de modo a complementar os dados desta base, foi utilizada a biblioteca *scikit-image*, que oferece 6 possíveis atributos a partir da matriz de coocorrência. São eles: contraste, dissimilaridade, homogeneidade, ASM (do inglês, *Angular Second-Moment*), energia e correlação.

Segundo Gebejes e Huertas (2013), contraste é a variação local dos níveis de cinza na matriz de ocorrência, semelhante à dissimilaridade. Já a homogeneidade mede a uniformidade de entradas não nulas na GLCM, semelhante à energia, que é o oposto à entropia. Por sua vez, entropia é descrita por Gonzalez e Woods (2010) como a medida da aleatoriedade dos elementos da matriz. Conforme expresso por Haralick et al. (1973), ASM é uma medida de homogeneidade da imagem, e correlação mede a dependência linear dos tons de cinza.

Ao todo, extraiu-se 16 atributos de cada tipo variando os parâmetros que definem o *pixel* vizinho. O primeiro parâmetro define o ângulo em radianos entre os vizinhos, foram usados $0, \frac{\pi}{2}, \pi$ e $\frac{3\pi}{2}$. O segundo define a distância em *pixels* entre os vizinhos, foram usados 2, 4, 8 e 16. Logo, a base com descritores que servirá como parâmetro de entrada para os algoritmos de aprendizado de máquina possui 46 atributos para 70 registros.

3.2. Pré-processamento

Nesse processo houve pequenas correções na base. Os problemas encontrados foram algumas falhas de digitação, que foram corrigidos de forma semiautomática. As incorreções se resumiam a valores nominais iguais em alguns atributos, porém escritos de forma diferente, por exemplo, alguns com letras minúsculas e outros com maiúsculas. Para resolver, filtraram-se de forma automática as sentenças que possuíam mais de 80% de semelhança, sucedido de um filtro manual para finalizar a análise e, após essa identificação, os selecionados foram padronizados.

3.3. Transformação

Nesse momento, atributos nominais foram transformados em categóricos numéricos, enumerados na base decimal, pois os algoritmos de classificação utilizados na mineração exigem entradas exclusivamente numéricas.

Percebeu-se também que existiam muitas possíveis classificações para poucos casos disponíveis, cerca de 10 classes nem sequer possuíam mais que um registro. A média era 3,125 registros para cada diagnóstico. A baixa densidade de registros nas classes é um problema porque compromete a generalização do algoritmo e, por consequência, o seu desempenho. Para contornar a situação, foi decidido que era necessário agrupar os diagnósticos para diminuir a quantidade de classes. Após testes preliminares em diversas distribuições, chegou-se àquela exposta na Tabela 1.

Tabela 1. Distribuição adotada de diagnósticos em grupos.

Grupo	Integrantes
0	Doença de Lesões mínimas; Glomeruloesclerose segmentar e focal; Podocitopatias (Lesões Mínimas X Glomeruloesclerose Segmentar e Focal não amostrada); Podocitopatia: Glomeruloesclerose segmentar e Focal não amostrada; Glomeruloesclerose segmentar e focal variante colapsante.
1	Glomerulonefrite membranoproliferativa tipo I; Glomerulonefrite membranoproliferativa tipo II (Doença de depósito denso);

	Glomerulonefrite membranoproliferativa tipo III; Glomerulonefrite difusa aguda (GNDA); Glomerulonefrite crescêntica; Glomerulonefrite membranoproliferativa; Glomerulonefrite lúpica focal e global - classe III; Glomerulonefrite lúpica difusa e segmentar - classe IV; Glomerulonefrite lúpica focal e segmentar - classe III; Glomerulonefrite lúpica membranosa - classe V; Glomerulonefrite lúpica mesangial proliferativa – classe II; Glomerulonefrite lúpica segmentar e focal – classe III (A) e associação com membranosa (classe V); Glomerulonefrite lúpica segmentar e focal - classe III.
2	Nefropatia diabética; Nefropatia por IgA; Nefropatia por IgA (doença de Berger) .
3	Glomerulopatia membranosa primária; Glomerulopatia membranosa secundária; Glomerulopatia membranosa; Glomerulopatia por fibronectina.
4	Rim normal; Doença da membrana fina; Nefrosclerose vascular hipertensiva; Amiloidose; Microangiopatia trombótica.

A codificação adotada discretizou os 32 diagnósticos em apenas 5. Isso, apesar de resolver o problema exposto, limitou bastante o nosso objetivo, pois as classes passaram a ser mais genéricas.

3.4. Mineração

Nessa etapa, foram feitos testes preliminares tanto para selecionar os melhores grupos de diagnóstico, com o resultado citado anteriormente no item 3.3, quanto também para escolher o melhor algoritmo de aprendizado de máquina supervisionado, rede neural ou floresta aleatória. Ao final, o algoritmo de floresta aleatória gerou melhores resultados, portanto, foi escolhido para ser aplicado às bases de dados definidas como foi dito no item 2.2.

3.4.1. Métricas

Para avaliar o algoritmo de classificação, foram utilizadas três métricas da biblioteca *scikit-learn*, são elas: acurácia, precisão e sensibilidade. Essas métricas são detalhadas na documentação da linguagem por *Scikit-Learn Developers* (entre 2007 e 2019) e são expostas nas Equações 01, 02 e 03. Na Equação 01, “ y_i ” é o valor real, “ \hat{y}_i ” é o valor predito e “ $n_{amostras}$ ” é a quantidade total de amostras. Nas Equações 02 e 03, as variáveis “ tp ”, “ fp ”, “ tn ” e “ fn ” são, respectivamente, o total de verdadeiros positivos, falsos positivos, verdadeiros negativos e falsos negativos.

$$acurácia(y, \hat{y}) = \frac{1}{n_{amostras}} \sum_{i=0}^{n_{amostras}-1} 1(\hat{y}_i = y_i) \quad \text{Eq.01}$$

$$precisão(tp, fp, tn, fn) = \frac{tp}{tp + fp} \quad \text{Eq.02}$$

$$sensibilidade(tp, fp, tn, fn) = \frac{tp}{tp + fn} \quad \text{Eq.03}$$

4. Resultados

Após a execução dos algoritmos, obteve-se 77.47% de acurácia na base de dados com os descritores e 82.26% sem eles, além dos valores das outras métricas expostos a seguir nos Gráficos 1 e 2.

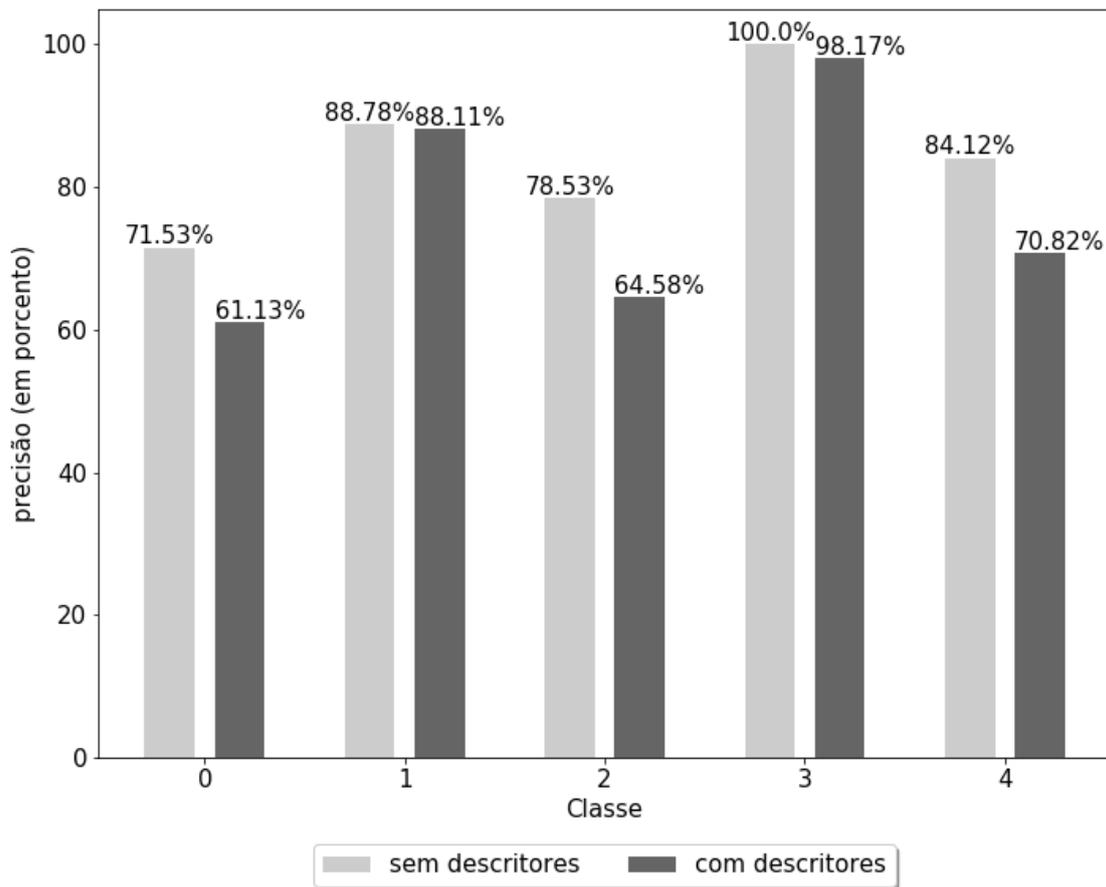


Gráfico 1. Precisão por classe

Dado os resultados obtidos pelo algoritmo, foi calculado a precisão da classificação tanto em uma base de dados com descritores quanto sem eles. No caso sem descritores, destacam-se a menor medida na classe 0, 71.53%, e o 100% obtido na classe 3. Já com os descritores, ocorreu uma queda tanto na acurácia quanto na precisão de cada classe. O erro esperado é devido à menor quantidade de registros na base de dados. Destacando a precisão na classe 2, que diminuiu 13.95%, a maior redução.

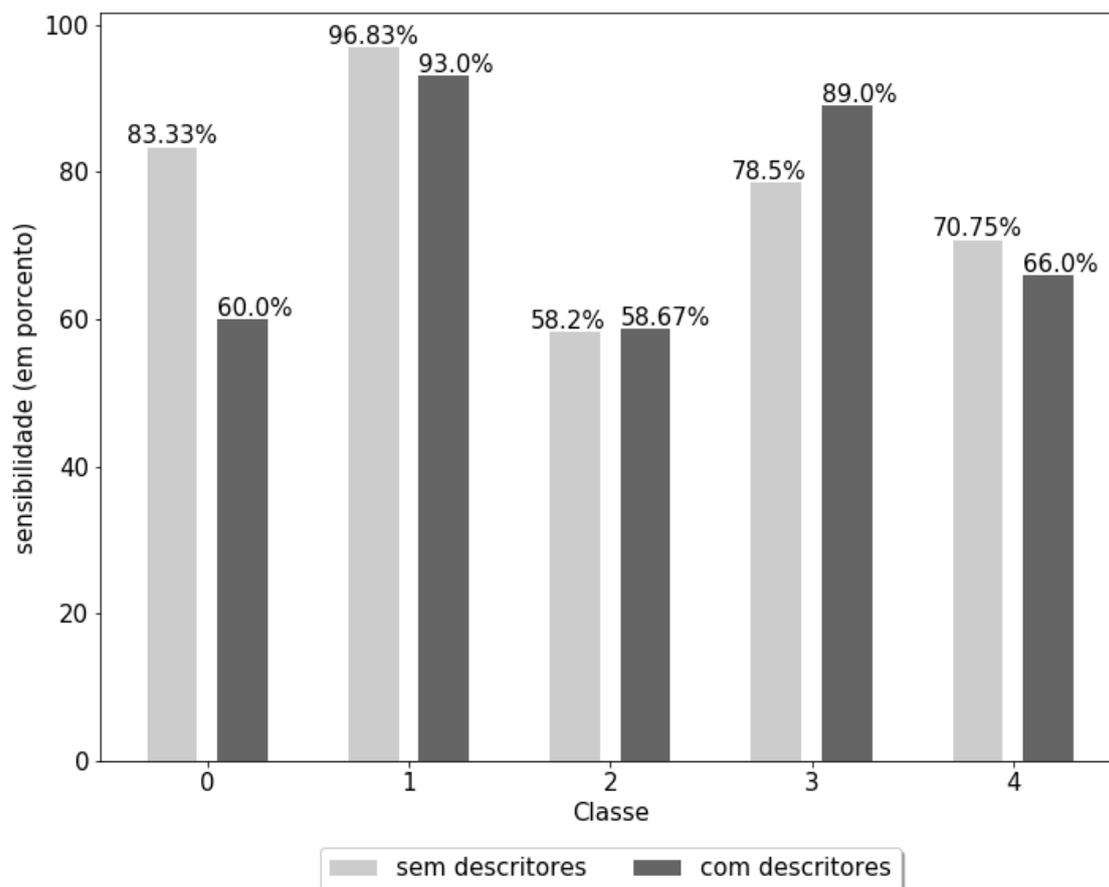


Gráfico 2. Sensibilidade por classe

Na métrica de sensibilidade, nota-se que adicionar os descritores não gerou uma queda em todas as classes, nas classes 2 e 3 os descritores melhoraram o desempenho. Ambas tiveram seu pior resultado na classe 2, 58.2% sem descritores e 58.67% com descritores. Também tiveram seu melhor índice na classe 1, 96.83% sem descritores e 93% com descritores.

É importante ressaltar que todos os resultados expostos são as médias encontradas em várias iterações com diferentes amostras de treinamento e teste, ou seja, são decorrentes de um algoritmo de validação cruzada. Para a base sem descritores foi utilizado 30% para testes e para a base com descritores 20%.

5. Conclusão

Apesar das dificuldades que foi percebido a priori, tanto a falta de registros quanto a dificuldade de encontrar descritores nas imagens que fossem viáveis a serem adicionados à base de dados, foi possível encontrar meios que contornaram esses problemas e geraram bons resultados.

A solução encontrada para a baixa densidade de casos nas classificações foi distribuir os diagnósticos em cinco grupos, isso gerou resultados satisfatórios. Chegou-se a 82.26% de acurácia. Porém houve uma leve queda nas métricas quando usado a base de dados com descritores retiradas da matriz de coocorrência de níveis de cinza das imagens, mas com trinta registros a menos, obtendo assim 77.47% de acerto total. Porém, essa redução ao adicionar os descritores extraídos da GLCM não diminui sua contribuição

para a construção do classificador, pois seria facilmente resolvida caso obtivesse imagens respectivas para todos os registros da base de dados.

Apesar dos resultados satisfatórios, nota-se que o tamanho da base de dados foi algo limitante. Trabalhos futuros com acesso a mais dados podem melhorar o desempenho do classificador e, talvez, possibilitar o uso dos diagnósticos em grupos menores ou sem qualquer agrupamento.

6. Referências

- BOENTE, A.N.P.; GOLDSCHIMIDT, R.R.; ESTRELA, V.V. Uma metodologia de suporte ao processo de conhecimento em bases de dados. In: SIMPÓSIO DE EXCELÊNCIA EM GESTÃO E TECNOLOGIA, 5., 2008. [Resende]. Anais eletrônicos [...]. Resende: AEDB Disponível em: <<https://www.boente.eti.br/publica/seget2008kdd.pdf>>. Acesso em: 19 fev. 2020.
- BREIMAN, L. Random Forest. Machine Learning, Países Baixos, v. 45, n. 1, out. 2001. Disponível em: <<https://link.springer.com/article/10.1023/A:1010933404324>>. Acesso em: 17 jun. 2020.
- FAYYAD, U. M.; PIATETSKY-SHAPIRO, G.; SMYTH, P. (1996). From Data Mining to Knowledge in Databases. AI Magazine, [Menlo Park], v.17, n.3, mar. 1996. Disponível em: <<https://www.aaai.org/ojs/index.php/aimagazine/article/view/1230>>. Acesso em: 19 fev. 2020.
- GEBEGES, A.; HUERTAS, R. Texture Characterization based on Grey-Level Co-occurrence Matrix. In: Conference of Informatics and Management Sciences, 2., 2013, Žilina, Anais eletrônicos [...]. Žilina: University of Žilina. Disponível em: <https://www.academia.edu/29807532/Texture_Characterization_based_on_Grey-Level_Co-occurrence_Matrix>. Acesso em: 17 jun. 2020.
- GÉRON, A. Hands-On Machine Learning with Scikit-Learn and TensorFlow: CONCEPTS, TOOLS, AND TECHNIQUES TO BUILD INTELLIGENT SYSTEMS. 1st ed. Sebastopol: O'Reilly Media, 2017.
- GOLDSCHIMIDT, R. R.; PASSOS, E. Data Mining: Um Guia Prático. Rio de Janeiro: Elsevier, 2005.
- GONZALEZ, R. C.; WOODS, R. E. PROCESSAMENTO DIGITAL DE IMAGENS. 3. ed. Tradução Cristina Yamagami, Leonardo Piamonte. São Paulo: Pearson Education, 2010. Título original: Digital image processing.
- HAN, J.; KAMBER, M.; PEI, J. DATA MINING: Concepts and Techniques. 3rd ed. Waltham: Elsevier, 2012.
- HARALICK, R.M.; SHANMUGAM, K.; DINSTEN, I. Textural Features for Image Classification. IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, [s. 1.], v.3, n.6, Nov, 1973. Disponível em: <<https://ieeexplore.ieee.org/document/4309314>>. Acesso em: 11 fev. 2020.
- SCIKIT-LEARN DEVELOPERS. Metrics and scoring: quantifying the quality of prediction. In: User Guide, [entre 2007 e 2019]. Disponível em: <https://scikit-learn.org/stable/modules/model_evaluation.html>. Acesso em: 26 de jun. 2020.