Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, v. 10, n. 1, 2023.

Trabalho apresentado no XLII CNMAC, Universidade Federal de Mato Grosso do Sul - Bonito - MS, 2023

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Classificação e Agrupamento de Dados Utilizando Conceitos dos Algoritmos k-NN e k-means

M. Massao-Hanaoka[1], C. E. Rubio-Mercedes[2]
Curso de Engenharia Física, Universidade Estadual de Mato Grosso do Sul, Dourados, MS

A área de aprendizado de máquina (ML - *Machine Learning*), é uma sub-área da inteligência artificial (IA) que encontra as mais diversas aplicações na era digital [1–3]. Técnicas baseadas em ML tem sido aplicadas com sucesso nas mais diversas áreas com a finalidade de analisar dados e elaborar previsões. Alguns exemplos são: visão computacional, engenharias, finanças, entretenimento, biologia computacional e medicina [2, 3]. O ML possibilita ao computador aprender com os dados - com pouca ou nenhuma intervenção humana. Há a crescente necessidade de extrair informações relevantes de grandes conjuntos de dados, algo impossível sem o auxílio de ferramentas computacionais. Neste trabalho são utilizadas ferramentas computacionais, nomeadamente, a linguagem de programação python em conjunto com as principais bibliotecas de análise de dados disponíveis gratuitamente. Espera-se obter resultados consistentes com a literatura, realizando estudos e implementações dos algoritmos propostos de maneira eficiente. Serão produzidos modelos de ML a partir dos dados, resultando em gráficos e estatísticas de grande relevância na prática de análise de dados.

Neste resumo serão apresentados os resultados parciais de classificação e agrupamento de dados utilizando os algoritmos k-NN e k-means.

Primeiramente, foi realizada a classificação utilizando o algoritmo **k-Nearest Neighbors (k-NN)**. Foi importado o dataset real $breast\_cancer$ da biblioteca $scikit-learn$. A função $train\_test\_split()$ foi aplicada sobre os componentes data e target para dividir o conjunto de dados em duas partes: treino (70%) e teste (30%). As variáveis-alvo são 'malignant' e 'benign', que indicam se o tumor foi classificado como maligno ou benigno. Os parâmetros para a classificação são especificados em $KNeighborsClassifier()$ e o método k-NN é executado através do comando $knn.fit()$. Por fim, a função $knn.score()$ calcula a acurácia para treino e testes, separadamente. Como descrito em [3], o número ideal para o parâmetro k foi de $k = 6$, que descreve o número de vizinhos ao usar o método k-NN sobre esse conjunto de dados. A predição é realizada para cada ponto dos dados de teste e comparada com resultados da literatura, de acordo com o conjunto de dados original. É possível avaliar o desempenho do modelo através do cálculo da acurácia.

Ao executar o programa, foram obtidos os seguintes resultados:

- Acurácia de treino: 92,2%.
- Acurácia de teste: 95,9%.

Depois, classificamos dados usando o algoritmo **k-means**. Foram importados dados do dataset sintético $make\_blobs$, da biblioteca $scikit-learn$. Os dados contem 5 centroides definidos para o objeto $blob\_centers$. Esses dados foram dispostos em um gráfico de dispersão que possibilitou a visualização, veja 1(a) e 1(b).

[1]mario.science@tutanota.com
[2]cosme@uems.br

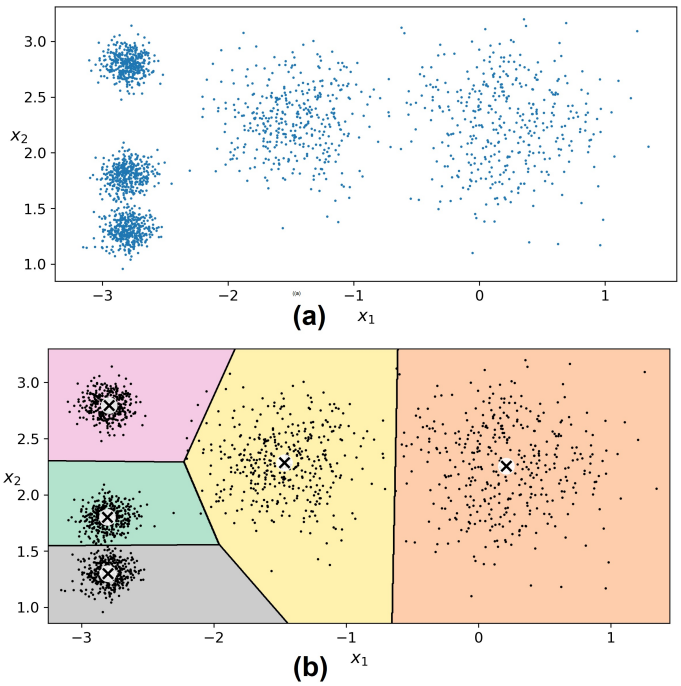

Figura 1: (a) Dispersão do conjunto *make_blobs* e (b) Dispersão com centroides e fronteiras de decisão

Para o agrupamento (*clustering*) dos dados, é necessário especificar o número $k = 5$ de clusters nos parâmetros da função $Kmeans()$. Esse número foi obtido visualmente a partir da observação do gráfico de dispersão, veja 1(a). Para a visualização da aplicação do método **k-means**, é interessante mostrar as fronteiras das regiões de decisão, bem como a localização dos centroides representados pelo símbolo X, veja 1(b). Para isso, foram definidos os parâmetros de $plot\_data()$, $plot\_centroids()$ e $plot\_decision\_boundaries()$.

No método **k-NN**, a obtenção de uma acurácia de teste maior que a de treino, sugere que ocorreu um sobreajuste nos dados. Ao aplicar o método **k-means** sobre o outro conjunto, a maioria das instâncias foram associadas ao cluster apropriado. No entanto, há poucas instâncias que foram provavelmente rotuladas erroneamente – especialmente próximo à fronteira entre o cluster do topo esquerdo e o cluster central. Esse resultado era esperado, pois o **k-means** não se comporta bem, de modo geral, quando os clusters tem diâmetros muito distintos. O principal parâmetro para o método é a distância do ponto até o centroide, o que pode causar confusão em regiões de fronteira.

Referências

[1] J. Alzubi, A. Nayyar e A. Kumar. "Machine Learning from Theory to Algorithms: An Overview". Em: **Journal of Physics Conference Series** 1142 (2018), p. 012012. DOI: 10.1088/1742-6596/1142/1/012012.

[2] C. M. Bishop. **Pattern Recognition and Machine Learning**. 1a. ed. Singapore: Springer, 2006. ISBN: 0-387-31073-8.

[3] A. C. Müller e S. Guido. **Introduction to Machine Learning with Python: A Guide for Data Scientists**. 3a. ed. Sebastopol: O'Reilly, 2017. ISBN: 978-1-449-36941-5.