Atividade prática: Algoritmo k-Nearest Neighbors (KNN)

Objetivo da atividade:

- Fazer uma implementação própria do algoritmo KNN para classificação
- Avaliar o uso do algoritmo KNN com diferentes valores de k, através do método *holdout*
- Implementar normalização de dados e verificar o seu efeito sobre os resultados da classificação

1. Faça uma implementação própria do algoritmo KNN na sua linguagem de programação de preferência. Utilize como base o pseudocódigo fornecido no Slide 24 da aula sobre KNN.
    - Sua implementação deve permitir informar/variar o valor do hiperparâmetro k para cada execução do algoritmo
    - Utilize como medida de distância padrão a *Distância Euclidiana*. Se desejar, pode permitir que esta medida seja configurada, por exemplo, que seja usada a *Distância de Manhattan*.
    - Assuma que sua implementação se destina à análise de dados quantitativos, isto é, não é necessário se preocupar em implementar suporte ao tratamento de dados qualitativos.

2. Implemente uma função para fazer a divisão dos dados em treino e teste, de acordo com o método *holdout*. A proporção de dados mantida no conjunto de treinamento (ou, alternativamente, direcionada ao conjunto de teste) deve ser informada como parâmetro da função (por exemplo, prop_treino = 0.8 indicaria que 80% dos dados serão usados como conjunto de treinamento). A divisão dos dados deve ser **estratificada** (veja material da Aula 04).

3. Utilize o conjunto de dados fornecido junto a este enunciado para avaliar a aplicação do KNN com diferentes valores de k. Os dados se referem à classificação de tumores de mama em maligno (1) ou benigno (0), de acordo com a coluna *target*. Os atributos (29 ao total) descrevem características dos núcleos celulares presentes em uma imagem digitalizada do material coletado

na biópsia pelo método *fine needle aspirate* (FNA). Estes dados foram obtidos do repositório PLMB[1].

- Observe o intervalo em que varia cada atributo no dado de treinamento: há uma grande diferença entre os valores máximo e mínimo de cada atributo? Discuta brevemente sobre a distribuição de valores e como isto pode impactar no processo de tomada de decisão do modelo.
- Faça a divisão dos dados em um conjunto de treinamento e outro de teste com a função implementada no item 2. Sugere-se utilizar 80% para treinamento e 20% para teste.
- A partir dos dados de treinamento, classifique os dados de teste usando **k=1, k=3, k=5, e k=7** (se desejar, avalie valores adicionais para k) com base na sua implementação. Avalie o desempenho do modelo usando a métrica de *acurácia* (taxa de acerto), reportando para cada valor de k a porcentagem de instâncias de teste classificadas corretamente.
- Faça uma avaliação do resultado, brevemente discutindo os achados e se existe alguma tendência ou associação entre desempenho e valor de k.

4. Implemente uma função para normalização dos dados utilizando o método min-max visto em aula (Aula 03 - slide 23). Ao utilizar este método, os valores dos atributos serão mapeados para distribuições variando no mesmo intervalo [0,1]. Aplique esta função nos dados de treino e teste obtidos no item 3 ((isto é, mantenha a mesma divisão de dados usada no item anterior), gerando dois novos conjuntos de dados normalizados[2]. Refaça o que se pede no item 3 com os dados normalizados, avaliando os mesmos valores de k.

- Analise e comente se houveram ou não diferenças em relação à taxa de acerto do algoritmo treinado e testado com dados não normalizados. A normalização impactou? De que forma: melhorando ou piorando o desempenho? Esta tendência foi observada para todos os valores de *k?*

Entregáveis:

- Código com a implementação do algoritmo. Pode ser em formato "notebook", se o aluno preferir (mas deve ser exportado em um arquivo em formato pdf para envio).
- Breve relatório (em **pdf**). devidamente identificado, contendo os resultados comentados para os itens 3 e 4 acima. A apresentação pode ser feita por meio

---

[1] Randal S. Olson, William La Cava, Patryk Orzechowski, Ryan J. Urbanowicz, and Jason H. Moore (2017). PMLB: a large benchmark suite for machine learning evaluation and comparison. *BioData Mining* 10, page 36. https://github.com/EpistasisLab/pmlb

[2] Ao longo da disciplina vamos discutir as formas mais corretas de lidar com a normalização de dados no pipeline de desenvolvimento de modelos preditivos.

de gráficos e/ou tabelas. O aluno deve interpretar os resultados, apontando os principais achados em relação a cada experimento e suas conclusões finais.
    - É interessante que os alunos adicionem ao relatório uma seção com breves instruções de como executar o código, no caso de não serem fornecidos notebooks com os códigos e experimentos.

Atenção: para esta atividade **não serão aceitas** soluções que aplicam implementações prontas do KNN de bibliotecas como sklearn (Python), caret (R), ou ferramentas como Weka, dentre outros.

O prazo final de entrega deste exercício é dia **18 de agosto às 23:59h**.