Atividade prática: Algoritmo k-Nearest Neighbors (KNN)

Objetivo da atividade:

- Compreender o processo de classificação através do algoritmo kNN
- Experimentar o uso do algoritmo kNN com diferentes valores de k, compreendo o impacto deste hiperparâmetro na decisão do modelo
- Analisar o efeito de aspectos como normalização de dados e dimensão do vetor de atributos sobre a saída do classificador kNN

Ferramentas que podem ser utilizadas:

- Python, scikit-learn e bibliotecas auxiliares (ou outra linguagem de programação e pacote/biblioteca de sua preferência).
- ChatGPT ou ferramentas de IA similares para auxiliar na estruturação inicial do código (leia com atenção as diretrizes do curso quanto ao uso de ferramentas de IA).

Orientações iniciais:

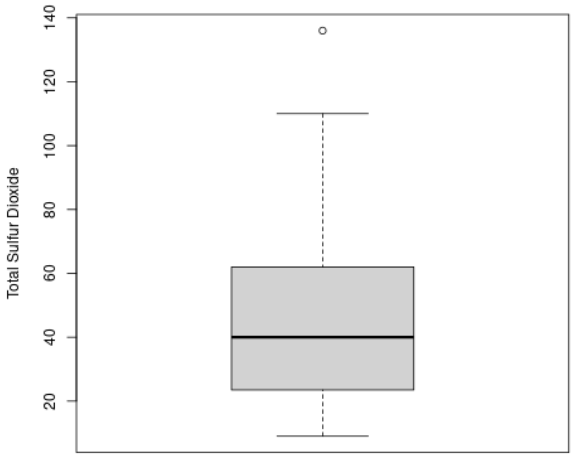
Faça o download dos dados disponibilizados no Moodle. Estes dados referem-se a uma tarefa de predizer a qualidade de vinhos tintos a partir da avaliação de um conjunto de características fisico-químicas analisadas.

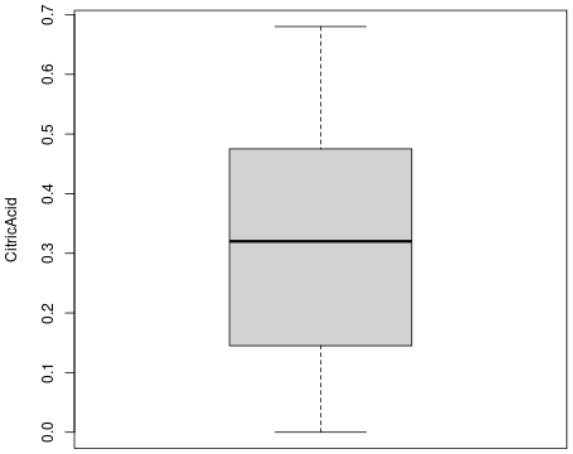
Os dados originais[1] foram pré-processados previamente para que a tarefa seja de classificação binária, diferenciando entre vinhos de alta qualidade (class=1) ou de baixa qualidade (class=0), e para obter apenas uma pequena amostra dos dados. Esta amostra disponibilizada no Moodle é composta por 44 instâncias de treinamento (utilizadas para treinar os modelos) e 4 instâncias de teste (utilizadas na aplicação do modelo, para avaliar o desempenho).

Cada instância possui um identificador único, disposto na coluna "ID". Atenção: Este atributo **não** deve ser usado na construção do modelo KNN.

Todos os atributos são numéricos, mas variam em escala. Por exemplo, para dióxido de enxofre total e ácido cítrico, temos as seguintes distribuições:

[1] https://archive.ics.uci.edu/ml/datasets/wine+quality





A estrutura das pastas fornecidas é explicada abaixo:

- `Dados_Originais_2Features`
  - Dados sem normalização, contendo apenas dois atributos selecionados, *dióxido de enxofre total* e *ácido cítrico*. Dados separados em TrainingSet (44 instâncias) e TestingSet (4 instâncias). Os arquivos possuem a coluna 'class' com a classe correta de cada instância. No TrainingSet, esta coluna será usada para guiar o aprendizado, enquanto no TestingSet, esta coluna será empregada para viabilizar a avaliação do modelo por meio da acurácia.
- `Dados_Normalizados_2Features`
  - Dados com normalização pelo método min-max (ver slides da disciplina), seguindo a mesma estrutura do item anterior (*Dados_Originais_2Features*)
- `Dados_Originais_11Features`
  - Dados sem normalização, contendo os 11 atributos disponibilizados originalmente. Dados separados em TrainingSet (44 instâncias) e TestingSet (4 instâncias). No TrainingSet, esta coluna será usada para guiar o aprendizado, enquanto no TestingSet, esta coluna será empregada para viabilizar a avaliação do modelo por meio da acurácia.
- `Dados_Normalizados_11Features`
  - Dados com normalização pelo método min-max (ver slides da disciplina), seguindo a mesma estrutura do item anterior (*Dados_Originais_11Features*).

Nesta atividade prática, você deverá treinar modelos preditivos utilizando o algoritmo kNN com a sua linguagem de programação ou software de preferência seguindo o guia de experimentos abaixo. A partir das suas observações sobre os seus resultados para cada item, responda as questões no questionário sobre o KNN disponível no Moodle da disciplina.

Atenção:

- Você pode utilizar implementações prontas[2] ou fazer a sua própria implementação do kNN (veja slides da disciplina). Entretanto, é importante que você seja capaz de recuperar as distâncias calculadas e índices/IDs dos K-vizinhos mais próximos, pois os mesmos deverão ser reportados nos resultados.
- Utilize sempre a distância euclidiana como métrica de distância no algoritmo kNN.
- A avaliação de resultados preditivos nos modelos treinados será feita através da acurácia, que se refere à taxa de acerto total: a quantidade de acertos do nosso modelo (i.e., classe predita igual à classe real) dividido pelo total da amostra. Você pode realizar a sua própria implementação ou utilizar uma função pronta para cálculo da acurácia.
- Os dados disponibilizados já estão separados em conjuntos de treinamento e teste, e devem ser usados conforme instruções a seguir. Não devem ser feitas outras divisões dos dados usando métodos como holdout, cross-validation, etc, que estudaremos posteriormente na disciplina.
- Não devem ser feitos outros pré-processamentos nos dados disponibilizados, a menos que seja especificado nas instruções abaixo. Siga estritamente as orientações para os experimentos.
- O seu código (script ou notebook) deve ser submetido no link para entrega disponibilizado no Moodle. A entrega deste item é essencial para "validar" suas respostas ao questionário.

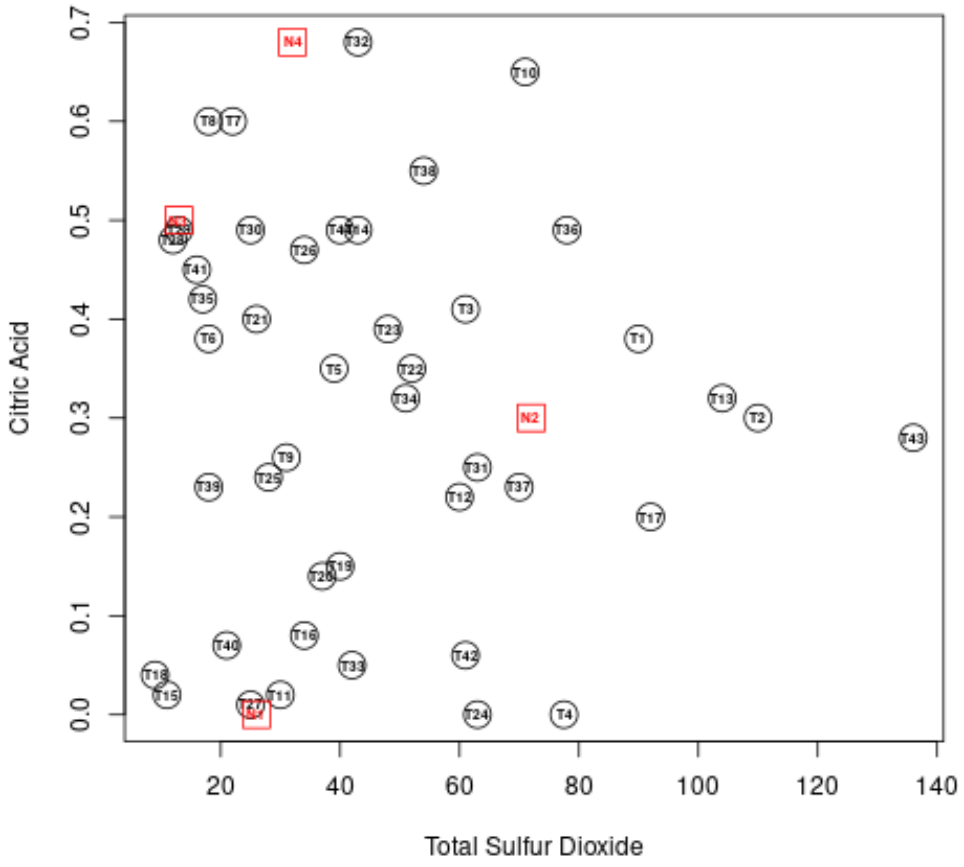
Guia de experimentos:

A. Treine modelos usando o algoritmo KNN para o conjunto de dados `Dados_Originais_2Features` (não normalizados), variando o valor de k (número de vizinhos mais próximos) entre 1, 3, 5, 7. Para cada modelo treinado, avalie seu desempenho nos dados de teste, reportando a acurácia. Repita o mesmo procedimento com os dados `Dados_Normalizados_2Features.` Compare as acurácias obtidas nos modelos treinados a partir destes dois conjuntos de dados, analisando se a normalização impactou de alguma forma

[2] Veja, por exemplo, este tutorial em Python: https://www.kaggle.com/code/prashant111/knn-classifier-tutorial

os resultados. Observe se a mudança no valor de k causou algum impacto no desempenho destes modelos (com e sem normalização dos dados) e, em caso positivo, se as variações no desempenho são as mesmas entre os modelos treinados com mesmo k, mas com dados distintos (dados originais e dados normalizados). *[Responda a pergunta 1 do questionário]*

Visualização do conjunto de dados `Dados_Originais_2Features`. Os pontos representados como quadrados vermelhos referem-se às instâncias de teste fornecidas para este dataset.



B. Considerando o modelo treinado com k=5 utilizando dados <u>não normalizados e com 2 atributos</u>, verifique quem são os k vizinhos mais próximos da instância de teste **N1** (liste os respectivos IDs)**.** Verifique como estes vizinhos estão dispostos no espaço de entrada em relação à instância de teste N1 e aos eixos x e y. Após tirar suas conclusões, analise se as mesmas se aplicam às instâncias de teste N2, N3 e N4. *[Responda as perguntas 2 e 3 do questionário]*

C. Treine dois modelos usando o algoritmo KNN com k=5 para os datasets `Dados_Normalizados_2Features` e `Dados_Normalizados_11Features`.

Aplique os modelos treinados nos respectivos dados de teste, verificando os k-vizinhos mais próximos e a classe prevista para a instância **N4**. Faça perturbações no valor do atributo “citric acid” para a instância **N4**, substituindo o valor original (1.0) por **0.3** e posteriormente por **0.85** (ou seja, gere duas novas instâncias sintéticas com esta alteração). Repita a classificação destas instâncias sintéticas com os dois modelos (isto é, modelo baseado em 2 atributos e em 11 atributos). Compare os resultados, analisando como a alteração de um atributo impactou o cálculo das distâncias euclidianas e a seleção dos k-vizinhos mais próximos em cada caso. *[Responda a pergunta 4 do questionário]*

As respostas serão submetidas via questionário no Moodle. Confira o prazo final desta atividade no questionário.