

1. A medida de acurácia de um classificador f é normalmente definida como:

$$acc(f) = \frac{1}{n} \sum_{i=1}^n I(y_i = f(x_i)),$$

sendo $I(\cdot)$ a função indicadora a qual retorna 1 quando a condição passada a ela é verdadeira e 0 caso contrário.

Embora a definição acima seja bastante simples e intuitiva, a mesma possui uma grande limitação em cenários nos quais as classes são desbalanceadas. Discuta tal limitação e o motivo de sua ocorrência.

2. Escreva um código que treine o classificador kNN na base de dados `regressao_seno_treino.csv` e teste-o na base `regressao_seno_teste.csv` e calcule a qualidade da solução com a medida *Mean Squared Error* (MSE).
3. Pesquise sobre a *Precision-Recall curve* (PRC) e explique-a resumidamente. Quais as principais diferenças entre ela e a curva ROC? Por fim, escreva um exemplo de código em Python que aplique o kNN na base de dados Breast Cancer Wisconsin Diagnostic, gere o *plot* da PRC e calcule a sua AUC.
4. Escreva um código Python que realize os seguintes passos:
- Divida a base de dados Iris de maneira estratificada em treinamento e teste contendo, respectivamente, 66% e 34% dos objetos.
 - Aplique o kNN com `weights = 'distance'`, `metric = 'euclidean'` e `n_neighbors = 1, 3, 5 e 10` e gere as respectivas curvas ROC (dica: para facilitar as entradas para o método `roc_curve`, utilize a função `label_binarize` do módulo `sklearn.preprocessing`).
 - Gere 3 gráficos distintos (um para cada classe da Iris) contendo as 4 curvas ROC obtidas (uma para cada configuração do kNN).

Por fim, discuta brevemente qual configuração apresentou os melhores resultados.