Lista de Exercícios 08

(prazo final para entrega: 06/05/2019 - segunda)

1) **Implemente as seguintes métricas de classificação:**
    a) **As métricas devem funcionar para classificação binária ou mesmo para múltiplas classes. Dica: você pode usar a função confusion_matrix do Scikit Learn para construir uma matriz de confusão e facilitar os seus cálculos. As métricas devem retornar a média ponderada com pesos baseados no suporte.**
        i) **accuracy**
        ii) **precision**
        iii) **recall**
        iv) **f1_measure**

2) **Divida de forma estratificada o Dataset Iris em apenas 10% para treino e 90% para teste. Essa divisão não é usual, mas vamos usar para tornar o problema mais desafiador.**
    - **Faça a standardização dos dados.**

3) **Crie um modelo preditivo de classificação multi-classe usando Logistic Regression.**
    - **Dica: você pode usar os parâmetros: multi_class='auto', solver='lbfgs'**

4) **Avalie usando suas implementações das métricas: accuracy, precision, recall e f1_measure.**

5) **Calcule a métrica log_loss usando a implementação do sklearn.**
    a) **Observe exemplos sobre Métricas de Classificação no github da disciplina. Lá tem exemplos completos.**
    b) **Note que y_pred em log_loss, nós chamamos de y_score, que é a probabilidade de ocorrência de cada classe para aquela linha do dataset**

6) **Desenhe a curva ROC para a classe Iris-virginica usando a implementação do sklearn.**
    a) **use a função roc_auc_score(y_true, y_score).**
        i) **preste atenção que y_score é diferente de y_pred.**

    - **Dicas: Observe exemplos sobre Métricas de Classificação no github da disciplina. Lá tem exemplos completos.**

7) **Calcule a métrica AUC (*Area Under Curve*) para a classe Iris-virginica usando a implementação do sklearn.**

    - **Dica: Observe exemplos sobre Métricas de Classificação no github da disciplina. Lá tem exemplos completos.**