Aprendizagem Automática 2022/23

Ficha prática 2

Exercício#2.1 - Conjunto de dados Iris - Algoritmo KNN

Neste exercício terá um contacto inicial com o software scikit-learn, e a aplicação dum algoritmo de classificação num conjunto de dados "Iris". Este é um conjunto de dados reais bastante conhecido, mas simples, com um reduzido número de instâncias (150) distribuídos de modo equilibrado em 3 classes, o qual permite classificar flores de três espécies próximas (*Iris Setosa*, *Iris virginica*, e *Iris versicolor*) com base em 4 atributos - comprimento e largura de pétalas e sépalas das flores.

Pode ter mais informação sobre este dataset em:

https://archive.ics.uci.edu/ml/datasets/iris ou

https://www.kaggle.com/datasets/arshid/iris-flower-dataset

Note-se que este conhecido conjunto de dados, que é disponibilizado no scikit-learn, é bastante simples e não coloca especiais desafios do ponto de vista da classificação automática, sendo considerado frequentemente como um "toy dataset". O scikit-learn disponibiliza à partida 7 conjuntos de dados reais (Boston, Iris, Diabetes, Digits, Linnerrud, Wine, Breast cancer). Pode consultar mais informação em

https://scikit-learn.org/stable/datasets/toy_dataset.html

Matriz de confusão

Uma matriz de confusão mostra para cada classe verdadeira qual o resultado do classificador automático. Note-se que apesar de a disposição mais usual ser a classe verdadeira estar disposta em linhas e o resultados do classificador (predição) em colunas, nalguns contextos (ferramentas de software e bibliografia) estas surgem trocadas. Pelo é sempre necessário ter atenção como está definida a matriz de confusão.

		Predicted			
Actual		$\mathbf{C_1}$	$\mathbf{C_2}$	$\cdots$	$\mathbf{C_n}$
	$\mathbf{C_1}$	N_{11}	N_{12}	$\cdots$	N_{1n}
	$\mathbf{C_2}$	N_{21}	N_{22}	$\cdots$	N_{2n}
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$\mathbf{C_n}$	N_{n1}	N_{n2}	$\cdots$	N_{nn}

1. Carregue o código do programa "iris.py" que permite carregar o dados Iris disponível no scikit-learn e aplicar o algoritmo de classificação **KNN** (K nearest neighbors) ao conjunto de dados. Execute-o e observe a matriz de confusão.

2. A partir da matriz de confusão verifique o número de exemplos nos conjuntos de treino e teste.

3. Identifique o que faz cada uma das chamadas de função no código.

4. Procure a função train_test_split() em https://scikit-learn.org/stable/ e aplique-a ao conjunto de treino gerando um novo conjunto com 100 amostras (sub amostra). Construa um modelo com o novo conjunto de treino e aplique-o ao conjunto de teste original. Observe o resultado em termos de desempenho.

5. Repita o procedimento para um conjunto de sub-amostras sucessivamente mais pequeno de: 100, 80, 60, 40, 20.

6. Utilizando a biblioteca "matplotlib" faça um gráfico do desempenho (exatidão) em função do número de exemplos do conjunto de treino.

Exercício#2.2 - Conjunto de dados Iris - Algoritmo KNN

1. Consulte no scikit-learn o algoritmo de classificação bayesiano GaussianNB/naive_bayes e adapte o código usado no exercício anterior de modo a aplicar este algoritmo.

2. Repita os procedimentos 1 a 6 do exercício anterior.

3. Usando a biblioteca matplotlib faça um gráfico comparando o desempenho de KNN com GaussianNB.