Aprendizagem Automática 2022/23

Ficha prática 3

Exercício#3.1 - Conjunto de dados Breast Cancer - Inspeção dos dados

Neste exercício terá um contacto com o data set "breast cancer" procedendo-se à inspeção visual dos atributos do conjunto de dados.

1) Depois dos imports necessários, carregue o conjunto de dados que faz parte do scikit-learn com

   load_breast_cancer()

2) A visualização dos dados pode ser feita de vários modos. Uma das bibliotecas bastante útil para esta tarefa é o *seaborn* (o *seaborn* é um package que tem um vasto e poderoso conjunto de ferramentas de visualização). No entanto o seaborn usa a estrutura de dados DataFrame, enquanto o scikit-learn usa a estrutura Bunch, pelo que se torna necessário adaptar os dados

   Pode ter mais informação sobre estas estrutura de dados usadas em Aprendizagem e Ciência dos Dados em:

   https://scikit-learn.org/stable/modules/generated/sklearn.utils.Bunch.html

   https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html

Use o seguinte código para adaptar o conjunto de dados *breast_cancer* depois de o carregar para a variável *cancer.*

```
data=cancer

df = pd.DataFrame(data.data,columns=data.feature_names)

df['target'] = data.target

selecao=df.iloc[:,[1,3,5,7,30]]   #seleciona apenas colunas 1,3,5 e 7 (30 é a classe/target)
```

Em seguida use a instruções

```
sns.pairplot(selecao, hue="target")

plt.show()
```

para criar um "***pair plot***" das variáveis. Note que pode selecionar alguns atributos (neste caso a 1º, 3º, 5º e 7º) ou mesmo todos, se o pretender. No entanto, quantos mais gráficos, mais pesada e lenta será a operação .

3) Explore um pouco os dados e visualize outros atributos à sua escolha, sendo as colunas, neste conjunto de dados, numeradas de 0 a 29.
4) O que conclui? Será à partida um conjunto muito difícil de classificar ? porquê ?

Exercício#3.2 - Conjunto de dados Breast Cancer - Árvore de Decisão - overfitting

1) À semelhança dos exercícios da aula passada. Adapte o código para:

- Carregar os dados
- Fazer um split treino-teste
- Aplicar um algoritmo de árvore de decisão- DecisionTreeClassifier usando um parâmetro para indicar a máxima profundidade da árvore, igual a 5.

    DecisionTreeClassifier(max_depth=5)

- Fazer o treino (método fit)
- Avaliar o desempenho usando o método score com os dados de teste (este irá calcular a exatidão/ precision)
- Repita o procedimento calculando o desempenho usando os dados de treino.

2) Coloque estes procedimentos num ciclo fazendo variar max_depth entre 1 e 20 e guarde os 20 valores obtidos numa estrutura de dados, por exemplo uma lista, ou um array.

   test_scores= np.append(test_scores,test_accuracy)

   e de igual modo para os dados de treino

   train_scores=np.append(train_scores,train_accuracy)

   Nota: o exemplo acima usa listas, mas poderá usar arrays do módulo array do python ou array do NumPy, que são tipos semelhantes mas diferentes (veja a documentação para compreender as especificidades de cada um).

3) Use plot em ambos os dados de desempenho e tente observar a evolução do desempenho com dados de teste e treino com a variação da complexidade.
4) Consegue identificar uma zona de underfitting e overfitting ?

   (compare com o gráfico apresentado na aula teórica)

5) Repita estes procedimentos de modo a repetir este teste várias vezes (experimente com 10, 50, e 100) de modo que de cada vez seja feito um split diferente, só no fim de chamar a função plot n vezes é que deverá chamar a função plt.show()
6) O que pode concluir ?

Exercício#3.3 - Repita esta análise com o algoritmo KNN - K vizinhos mais próximos.

1) Faça variar o K de 1 até N
2) Observa as curvas de desempenho em função de K. O que nota em termos de semelhanças e diferenças em relação ao exercício anterior? Note que o hiper-parâmetro K indica o número de vizinhos mais próximos que será tido em conta. A um K mais elevado corresponderá um modelo mais complexo ou mais simples?