Grupo 25 – AMD – AP06

Exercício 3

Neste exercício é aplicado maioritariamente o método holdout, este método foca-se em separar um dataset inicial em duas partes distintas. Uma dessas partes é para testar enquanto que a outra é utilizada para treinar.

3.a)

Utilizando o método holdout com o dataset sugerido, verificou-se que o conjunto de treino ficou com a maioria das intâncias, enquanto que o conjunto de teste ficou com uma apenas. Todas as instâncias foram retiradas de forma aleatória também. O único cuidado presente que houve foi não existir repetição de valores entre conjuntos, tendo em conta que é errado.

3.b)

Ao alterar para a utilização de 50% do dataset para treino e para teste, os conjuntos passaram a ter a mesma dimensão de 3 elementos. Continuam a não apresentar representatividade.

3.c)

Neste caso, mesmo mantendo o split 50/50 os valores dentro de cada conjunto (treino e teste) são diferentes. A razão para tal deve-se à tentativa de representatividade dos valores dos conjuntos.

3.d)

No caso do repeated holdout, é mitigado o bias dos anteriores que apenas têm uma escolha. O Holdout ainda é estocástico, pelo que a repetição ajuda a adicionar carácter determinístico. A escolha inicial tem valores idênticos à anterior.

Exercício 4

Neste exercício foram analizados todos os métodos : fold_split, stratified_fold_split, repeated_fold_split, repeated_stratified_fold_split, leave_one_out e leave_p_out.

Notou-se uma semelhança entre os métodos fold_split, stratified_fold_split, repeated_fold_split, repeated_stratified_fold_split e os anteriores do exercício 3. O método base era totalmente diferente, sendo que este se baseia em fazer partições do dataset, e uma partição servirá como conjunto de teste enquando que as restantes serão conjuntos de treino. Após isto são feitas tantas repetições como o número de partições existentes, sendo que a partição de teste é sempre diferente. Em relação às variantes stratified implica haver representatividade das classes nas partições, repeated implica que há n iterações * p partições, ao invés de só p partições, e repeated stratified é a junção dessas duas variantes.

O método leave_one_out, deixa um valor de fora para teste, e treina com todos os outros, após isso escolhe um valor diferente para teste e repete. Isto é feito até todos os valores existentes no dataset terem sido utilizados para testar. O problema deste método é que é computacionalmente pesado.

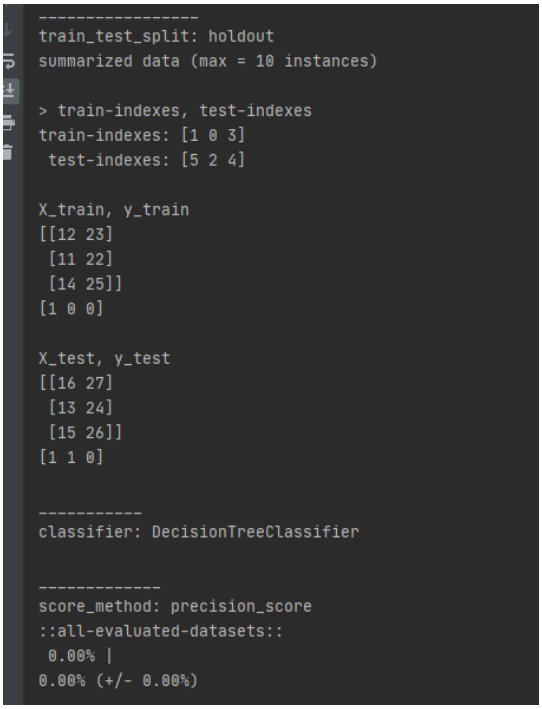
Por fim, o método leave_p_out, permite escolher qual a dimensão do conjunto de teste, em vez de este ter sempre uma unidade de dimensão. A dimensão do conjunto percorrerá todas as combinações possíveis de valores existentes no dataset, pelo que a sua complexidade computacional sobe exponencialmente. Apesar de apresentar uma grande quantidade de dados para teste, também é computacionalmente muito pesado, ainda mais que o leave_one_out.

Exercício 5

Para este exercício foi feita a implementação do método de divisão de datasets de nome Bootstrap. As variantes implementadas foram o SplitOnce que apenas faz uma divisão, e o SpliRepeated que faz a divisão N vezes, em que N é escolhido pelo programador. De forma a garantir consistência é usado uma seed para garantir modelos idênticos quando utilizada a mesma seed. Por fim foram passadas as implementações para a pasta do exercício 2 de forma a fazer lá a implementação do Bootstrap, que não é providenciado pelo SKLearn.

Exercício 6

Para este exercício foram seguidas as instruções. Tendo em conta que ambas se suportavam umas nas outras, é apresentada a figura da alínea final. Não foi possível avaliar corretamente o método de classificação, ou então ele teve mesmo muito maus resultados. Apenas foi utilizada a precisão que é um indicador que faz mais sentido ser utilizado com outros, no entanto como este teste era muito simples, o grupo achou por bem evitar complexidade desnecessária.



Exercício 7

7.f)

São apresentados os avisos de “no true samples” e de “no predicted samples”, em algumas instâncias de métricas do classificador.

7.g)

Neste caso, “no true samples” implica que não foram previstas nenhumas amostras que fossem falsos negativos e verdadeiros positivos, o nome deve-se ao cálculo do recall, que é o cálculo do true positive rate.

No caso dos “no predicted samples”, há a implicação de não existirem nem verdadeiros, nem falsos positivos. O nome do mesmo deve-se ao cálculo da precisão que é positive predictive rate.

7.h)

Caso não existam previsões que sejam falsos negativos e verdadeiros positivos, estes warnings acontecem. Isto acontece porque no cálculo do recall, no denominador apenas são utilizados os valores dos falsos negativos e verdadeiros positivos, e se o total for 0, a conta acaba a ser uma divisão por 0, que é impossível.

7.i)

Caso não existam previsões de cariz positivo (verdadeiros e falsos positivos) a previsão é necessariamente 0. Isto acontece porque no cálculo da mesma, apenas são utilizados os valores de positivos reais, sobre o valor de positivos totais, e sendo o total 0, a conta acaba a ser uma divisão por 0, que é impossível. Isto aconteceu porque não houve previsões positivas.