

Atividade 5 – Classificador com calibração automática de um hiperparâmetro.

1 Descrição

Nessa atividade, você deve construir um novo classificador usando o `scikit-learn`. Seu código irá ajustar automaticamente um dos hiperparâmetros do classificador.

O usuário do seu novo classificador vai escolher um “classificador base” F , um hiperparâmetro P deste classificador F , e uma lista de valores L para o hiperparâmetro P . Na fase de treinamento, seu código irá usar validação cruzada (sobre o conjunto de treinamento) para determinar qual o valor mais apropriado em L para o hiperparâmetro P .

Como exemplo, suponha que o usuário escolheu rodar seu código com os parâmetros:

```
F=tree.DecisionTreeClassifier,  
P=max_depth,  
L=[4, 5, 6].
```

Estes valores são válidos porque `max_depth` é, de fato, um hiperparâmetro do classificador `DecisionTreeClassifier`, e este hiperparâmetro pode assumir os valores 4, 5, ou 6. Como a lista L tem tamanho igual a 3, o método `fit` de seu código irá criar três classificadores do tipo `DecisionTreeClassifier`: um com `max_depth=4`, outro com `max_depth=5`, e outro com `max_depth=6`. Cada um deles será avaliado por meio de um experimento de validação cruzada com 2 dobras (*2-fold cross validation*)¹. Aquele classificador que obtiver maior precisão média no experimento de validação cruzada será o classificador definitivo. Este classificador definitivo deve ficar salvo para ser usado durante chamadas ao método `predict` ou ao método `score`.

Utilize como ponto de partida o código visto em sala para criação de um novo classificador, a partir do zero, usando o `scikit-learn`, disponível no SIGAA: data 02/05/2019, item “classificador da classe mais frequente”.

Como exemplo, assuma que seu novo classificador é implementado em uma classe de nome `ClassificadorComAjuste`. Seu código deve aceitar o seguinte tipo de uso:

```
>>> clf = ClassificadorComAjuste(DecisionTreeClassifier, 'max_depth', [4,5,6])  
>>> clf.fit(observacoes, classe)  
Avaliando classificador com <max_depth=4>. Acuracia: 0.8924  
Avaliando classificador com <max_depth=5>. Acuracia: 0.9170  
Avaliando classificador com <max_depth=6>. Acuracia: 0.9042  
Melhor classificador: <max_depth=5>.  
>>> clf.score(teste, classe_teste)  
0.8716
```

Para configurar o hiperparâmetro de um classificador qualquer utilizando o nome do parâmetro como uma string, você pode usar o recurso de passagem de parâmetros por meio de um dicionário:

```
parametros = dict()  
parametros['max_depth']=7  
f = DecisionTreeClassifier(**parametros)
```

O código acima cria um classificador `DecisionTreeClassifier` com `max_depth=7`. Já o código a seguir **não** funcionaria:

```
nome = 'max_depth'  
valor = 7  
f = DecisionTreeClassifier(nome=valor)
```

¹O valor 2 para a validação cruzada é fixo, independente do tamanho de L e do tipo de classificador usado.

2 Entrega

- Pontuação total pela atividade: 2,0 pontos (dois pontos).
- Entregáveis:
 - Código Python.
- Data de entrega: **15 de junho** de 2019, 23:59:59, via SIGAA. **Não** via e-mail.
- Data de entrega com adiamento automático: **17 de junho** de 2019, 23:59:59, via SIGAA. **Não** via e-mail.

3 Mais detalhes

- A atividade pode ser feita individualmente ou em dupla.