13

Implementando a terceira fase do algoritmo

Dica: Aqui (https://github.com/alura-cursos/machine-learning-introducao-a-classificacao/archive/master.zip) você pode encontrar o projeto completo do curso para download.

Até agora só implementamos as duas fases do algoritmo, ou seja, o treino e teste dos algoritmos. Porém, precisamos implementar a fase final, que é justamente escolher o melhor dentre os algoritmos e então realizar o teste do mundo real, em outras palavras, testar o algoritmo com os dados de validação. Para isso, precisamos primeiro verificar qual dos algoritmos é o vencedor, podemos fazer isso adicionando um `if` que verifica qual foi o maior resultado, mas, atualmente, apenas a função `fit_and_predict` contém essa variável que é a `taxa_de_acerto` , ou seja, retorne essa variável dentro da função `fit_and_predict` .

Agora crie variáveis que irão representar cada resultado dos algoritmos, por exemplo, para o `Multinomial` crie a variável `resultadoMultinomial` e para o `AdaBoost` a variáveis `resultadoAdaBoost` , então, no momento que estiver chamando o método `fit_and_predict` para cada um dos algoritmos, atribua o retorno da função para as variáveis de cada algoritmo.

Então, faça o `if` que verifica se o valor da variável `resultadoMultinomial` é maior que `resultadoAdaBoost` , se for verdade, crie a variável `vencedor` atribua o valor da variável `resultadoMultinomial` , caso for falso, crie também a variável `vencedor` e retorne o valor da variável `resultadoAdaBoost` .

Precisamos realizar o teste do mundo real, para isso, criaremos a função `teste_real` que receberá as variáveis `vencedor` (modelo vencedor), `validacoes_dados` e `validacao_marcacoes` . A implementação desse método é muito similar ao `fit_and_predict` a única diferença é que precisa alterar as variáveis de dados e marcações para as de validação conforme abaixo:

```
    resultado = modelo.predict(validacao_dados)
    acertos = resultado == validacao_marcacoes

    total_de_acertos = sum(acertos)
    total_de_elementos = len(validacao_marcacoes)

    taxa_de_acerto = 100.0 * total_de_acertos / total_de_elementos
```

Da mesma forma que fizemos anteriormente, precisamos também imprimir a mensagem indicando o quanto o algoritmo vencedor acertou com a seguinte mensagem:

```
msg = "Taxa de acerto do vencedor entre os dois algoritmos no mundo real: {0}".format(taxa_de_acerto
    print(msg)
```

Com a função implementada, chame-a enviando os parâmetros necessários. Após a chamada da função, precisamos apenas alterar o código que imprime o resultado do algoritmo base e a quantidade de testes realizados:

```
acerto_base = max(Counter(teste_marcacoes).itervalues())
taxa_de_acerto_base = 100.0 * acerto_base / len(teste_marcacoes)
print("Taxa de acerto base: %f" % taxa_de_acerto_base)
```

```
total_de_elementos = len(teste_dados)
print("Total de teste: %d" % total_de_elementos)
```

Observe que esse código está utilizando as variáveis de teste, porém, nesse instante, estamos interessados em saber os resultados apenas para os dados de validação, ou seja, substitua todas as variáveis de teste para os de validação no trecho de código acima. Por fim, execute o código, então, cole o código e o resultado.

Responda

INSERIR CÓDIGO		FORMATAÇÃO