

Avaliando com dados reais

Transcrição

O que faltou fazer foi utilizar o avaliador para trabalhar com os dados reais que utilizamos na aula anterior. Para isso precisamos alterar a classe `Avaliador` para que passe a utilizar o modelo de cursos ao invés do de produtos:

```
public class Avaliador {  
    public static void main(String[] args) throws IOException, TasteException {  
        RandomUtils.useTestSeed();  
  
        DataModel modelo = new Recomendador().getModeloDeCursos();  
  
        RecommenderEvaluator evaluator = new AverageAbsoluteDifferenceRecommenderEvaluator();  
        RecommenderBuilder builder = new RecomendadorBuilder();  
        double erro = evaluator.evaluate(builder, null, modelo, 0.9, 1.0);  
        System.out.println(erro);  
  
    }  
}
```

Duas são as diferenças notáveis em relação ao código, a primeira delas é que antes nomeávamos o modelo de `produtos` e agora chamamos apenas de `modelo`. Isso ajuda a deixar o código mais genérico, temos um modelo e não importa se é de produtos ou de cursos. A outra é que renomeamos a classe que se chamava `RecomendadorDeProdutosBuilder` para apenas `RecomendadorBuilder` tornando-a mais genérica também. O código cria um recomendador baseado em um modelo, por isso não precisa deixar seu nome explícito que é de produtos ou cursos. Isso ajuda também a eliminar possíveis duplicações que poderíamos fazer por causa da nomenclatura das classes.

Ao executar o avaliador com os dados dos cursos temos como resultado da margem de erro o valor `1.2031058431035093`. O que esse valor indica mesmo? Que a taxa de erro é de 1.20 para mais ou para menos. Essa taxa é uma média baseada em todas as recomendações analisadas.

Lembrando que ao executar a análise, o algoritmo esconde 10% dos dados, treina as recomendações, depois junta todos os dados e avalia a diferença. Outro ponto a se observar é a escala. Em um curso em que o algoritmo chuta que a nota será 10, a taxa de erro nos indica que a nota real deverá ser algo entre 8.8 e 10. A diferença de 1.20 não é algo ruim porque o importante é que o aluno goste do curso e se o mesmo avaliar **em média** o curso entre 8 e 10, então, muito provavelmente o aluno gostará do curso.

Caso pedíssemos por exemplo, para o algoritmo recomendar 600 cursos para o usuário 15, teríamos o seguinte resultado:

```
RecommendedItem[item:222, value:5.792939]
```

Onde o curso 222 é um curso de *Mock para C#* onde, ele avaliaria com uma nota entre 4.6 e 7, o que provavelmente seria um curso ruim de se recomendar para ele naquele momento. Talvez em outro momento essa recomendações se tornasse mais apropriada.

É importante frisar que apesar de 1.2 parecer um valor grande, este é um valor que permite saber de antemão se uma recomendação será boa ou ruim. É comum que como existe um bom número de recomendações que você gosta, será com elas que começaremos a recomendar.

Em e-mail de recomendações, por exemplo, podemos escolher um top 6 de sugestões avaliadas pelo algoritmo e depois as próximas 6, sempre tomando o cuidado para que não recomendemos um curso que seja provável que o aluno não goste!

Um ponto interessante de se observar é que o usuário 1 fez apenas 2 cursos em cerca de 400 cursos, num total de 4600 alunos e em uma análise de preferência de 34 mil linhas. Mas, dá pra melhorar?

Para quem tem mais tempo na plataforma e que tenha feito, por exemplo, 20 cursos, 40 curso e até mais, as recomendações serão mais assertivas pelo acúmulo de preferências destes usuários, mas e para quem começou ontem? Como a gente faz? Dá pra sugerir algo para quem não fez nenhum curso? E para quem fez apenas um curso?