

02

Avaliando o problema das escalas

Transcrição

Adicionando perguntas sobre preferências dos usuário será possível sugerir cursos de forma mais assertiva, mas temos um problema: a escala.

Quando utilizamos escalas diferentes afetamos diretamente a análise. Os cursos usam para avaliação notas de 0 a 10, portanto, temos uma taxa de erro que também pode variar de 0 a 10, que é um valor consideravelmente grande. Porém, para as perguntas fazemos uso de uma escala na qual as notas variam de 0 e 1.

Isso quer dizer que estamos praticamente dizendo ao algoritmo para que não se preocupe em recomendar o que foi obtido a partir das perguntas, mas sim focar nos cursos em si. Lembra das notas do avaliador?

```
RecommendedItem[item:1000, 0.90620375]
RecommendedItem[item:1001, 0.8568823]
RecommendedItem[item:1002, 0.0]
```

Os dois primeiros itens estão próximos de 1, considerando isso, muito provavelmente deveriam ser recomendados os dois primeiros itens, mas não o último. Porém, como já observamos, eles não serão recomendados por causa da diferença das escalas.

Uma possibilidade é transformar tudo que for 1 em 10 e deixa tudo que for 0, 0 mesmo. Mas, se fizermos isso, a taxa de erro do que antes era 1.14 passa para 1.27, ou seja, o resultado são recomendações menos assertivas.

Essa transformação aumenta a escala, porém ela tem um problema, pois as notas variam de 0 e 10, ou seja, ou 0 ou 10, com isso deixamos os dados em uma mesma escala, porém com uma variância muito grande entre os itens de pergunta.

O que estamos fazendo é, de certa forma, perturbar a análise do algoritmo por causa da diferença das escalas. Nestes casos, não temos uma resposta clara de como agir, pois tudo depende. Não tem certo, nem errado: é uma escolha.

Uma última refatoração em nosso projeto será mover os arquivos que possuem os dados de análise para dentro do `src/main/resources`. Estes arquivos são o `dados.csv`, `cursos.csv` e `cursos2.csv`.

Depois de mover os arquivos para o novo diretório, precisamos atualizar o método `getModel` na classe `Recomendador` para refletir a alteração de caminho dos arquivos:

```
private DataModel getModelo(String path) throws IOException {
    File file = new File("src/main/resources/" + path);
    return new FileDataModel(file);
}
```

Execute o projeto novamente para garantir que tudo continue funcionando normalmente.

