01

Obtendo mais informações dos usuários

Transcrição

O nosso arquivo *cursos.csv* é uma parte parcial anônima de um banco de dados real. Nele temos alunos mais antigos, mais recentes, menos ativos mais ativos, etc. Agora, vamos trabalhar com sugestões para quem é mais recente ou até mesmo para aqueles que não possuem dados, como no caso dos alunos que não fizeram nenhum curso.

Imagine a Amazon, por exemplo, como ela recomendaria um livro sem que você tenha lido e avaliado o atual? Nestes casos precisamos usar de outras estratégias para recomendar itens. Como podemos trabalhar isso?

Considerando que não temos dados para sugerir cursos para o aluno, poderíamos utilizar de perguntas para coletar informações que indiquem algum tipo de interesse, perguntas como:

- Você está na faculdade?
- Você quer receber newsletter de 10 dicas de Java?
- Você quer receber newsletter de 10 dicas de SQL?
- Você quer receber newsletter de 10 dicas de Design?

Note que apesar das perguntas serem meio aleatórias, elas podem, caso respondidas positivamente, indicar que cursos seriam de seu interesse. O aluno que deseja receber dicas de design provavelmente faria ou fará um curso de design.

Neste sentido, teríamos a mesma tabela inicial que vimos no começo do curso, porém com as colunas referentes as respostas das perguntas adicionais.

	Android	UX	Photoshop	Cordova	Mailing Mobile	Mailing Design
Guilherme	9	8		10	10	0
Joana						
João						10
Daniela		9	10	7	0	10

O caso da Joana, por exemplo, que respondeu querer receber emails sobre Design mas não de Mobile, que cursos recomendaríamos para ela?

	Android	UX	Photoshop	Cordova	Mailing Mobile	Mailing Design
Guilherme	9	8		10	10	0
Joana		ESSE	E ESSE		0	10
João						10
Daniela		9	10	7	0	10

É preciso ter atenção sobre a relação que será feita. A ordem não importa na avaliação de que quais dicas ela quer receber. Vamos relembrar dos rascunhos iniciais, por proximidade as notas da Joana serão em média próximas as notas da Daniela.

Considerando isso, utilizaremos o arquivo cursos2.csv (https://github.com/alura-cursos/machine-learning-introducao-aos-sistemas-de-recomendacoes/blob/master/src/main/resources/cursos2.csv). Nele temos mais dados que no primeiro arquivo e dentre eles estão os itens 1000, 1001, 1002 que não são cursos, mas perguntas que buscam conhecer melhor as preferências do usuário.

Se utilizarmos esse arquivo teremos adição de dados e, consequentemente, uma taxa de erro menor, de 1.20 para um valor próximo a 1.14. Lembrando que, a adição dos dados não indica que todos os usuários do sistema responderam as perguntas. No caso, são cerca 11.000 dados a mais no `cursos2.csv`, considerando 4.600 alunos temos uma média de que 2.4 das perguntas respondidas pelos usuários, de um total de 3.

É fácil de perceber que com pequenas perguntas pudemos fazer com que a assertividade do recomendador aumentasse. Perguntas simples que levam menos de um minuto para o usuário responder. A partir delas podemos agrupar preferências e identificar usuários. Esse conceito existe também na psicologia: usuários com preferências similares se agrupam e se comportam de forma similar. Essas perguntas ajudam neste ponto, pois podemos agrupar usuários em determinado grupo, mesmo com poucos dados sobre eles.

> **Observação:** Como os dados dos alunos são anônimos é normal que o usuário 15 do primeiro arquivo seja diferente do usuário 15 do segundo arquivo de cursos. Porém o mesmo grupo de alunos foi utilizados. As perguntas utilizadas também tiveram notas diferentes da escala de cursos, nos cursos é possível avaliar de 0 a 10 e nas perguntas utilizamos 0 ou 1 e essa é uma preocupação que devemos ter.

Executando nosso recomendador, veremos que ele continua funcionando como antes, mas no final das recomendações teremos algo como:

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

Lembra que os itens 1000, 1001 e 1002 são perguntas referentes as preferências do usuário? Se ele desejaria receber newsletter ou mesmo se estão na faculdade?

Essas são preocupações que precisamos ter principalmente porque utilizamos escalas de notas diferentes, enquanto algumas recomendações terão notas próximas de 10, essas perguntas recomendarão itens entre 0 e 1. Elas poderiam estar no topo da lista, mas por essa escala, sempre estarão no final.

Precisamos ter bastante cuidado ao misturar tipos de dados diferentes na análise do algoritmo. Veremos o por quê.