

DATA ANALYTICS

FASE 3 | AULA 06 -

RECOMENDAÇÕES COM O ALGORITMO ALS

SUMÁRIO

O QUE VEM POR AÍ?	3
HANDS ON	4
SAIBA MAIS.....	5
O QUE VOCÊ VIU NESTA AULA?	13
REFERÊNCIAS.....	14

EMSE

O QUE VEM POR AÍ?

No mundo da personalização e da recomendação de conteúdo, o algoritmo ALS (Alternating Least Squares) se destaca como uma poderosa ferramenta para a construção de sistemas de recomendação eficientes. Se você está interessado(a) em aprender sobre o funcionamento e a aplicação do algoritmo ALS, esta aula introdutória é o ponto de partida ideal para mergulhar nesse fascinante campo.

Durante ela, você aprenderá os princípios fundamentais da filtragem colaborativa e entenderá como o algoritmo ALS é usado para criar recomendações personalizadas. Você descobrirá como esse algoritmo utiliza técnicas matemáticas avançadas para analisar padrões de preferências dos(as) usuários(as) e encontrar relacionamentos entre itens, permitindo gerar sugestões relevantes e precisas.

Ao longo do curso, você terá a oportunidade de explorar exemplos práticos e aplicar o algoritmo ALS utilizando ferramentas e bibliotecas populares, como o Apache Spark. Você aprenderá a preparar e processar os dados, treinar o modelo ALS e gerar recomendações personalizadas com base nas preferências dos usuários.

Prepare-se para desvendar a magia por trás dos sistemas de recomendação com o algoritmo ALS e compreender como ele transforma dados brutos em sugestões valiosas. Com essa aula introdutória, você estará pronto(a) para aplicar o algoritmo ALS em seus próprios projetos e criar sistemas de recomendação eficientes, impulsionando a personalização e aumentando a satisfação de usuários.

HANDS ON

Agora entenderemos mais sobre o algoritmo ALS e suas aplicabilidades práticas. Vamos colocar a mão na massa!

EMSE

SAIBA MAIS

Introdução a Sistemas de Recomendação

Sistemas de recomendação são técnicas de software que fornecem sugestões de itens a serem recomendados para um(a) usuário(a).

As sugestões fornecidas nos sistemas de recomendação visam ajudar os(as) usuários(as) em vários processos de tomada de decisão, bem como quais itens comprar, quais música escutar ou quais notícias ler.

Sistemas de recomendação provaram ser valiosos por ajudar as pessoas a lidarem com a sobrecarga de informações, ao filtrarem e recomendarem o que seria de interesse a elas.

Já para o comércio, tornou-se uma das ferramentas mais poderosas e populares ao recomendar produtos ou serviços de acordo os hábitos dos(as) usuários(as).

Existem duas arquiteturas básicas no uso de recomendação de sistemas, que são:

1. Sistemas de recomendação baseado em conteúdo

No sistema de recomendação baseado em conteúdo, o foco é nas propriedades dos itens. A similaridade de um item recomendado será medida pela similaridade com as propriedades do item que o(a) usuário(a) tenha adquirido ou pesquisado anteriormente.

Como exemplo de uso do modelo baseado em conteúdo/item, temos:

- Produtos recomendados com base em sua compra/pesquisa (E-commerce).
- Sons parecidos que começam a tocar após o término de sua playlist (Spotify).

2. Sistemas de Recomendação com Filtragem Colaborativa

No sistema de recomendação baseado em filtragem colaborativa, o foco é na relação entre usuários e itens.

A similaridade dos itens é determinada pela similaridade da avaliação deles, pelos(as) usuários(as) que tenham avaliado os mesmos itens, ou seja: se os(as) usuários(as) tiverem avaliado itens com notas similares, provavelmente eles(as) têm gostos parecidos e aceitam recomendações com base nesse critério.

Como exemplo de uso do modelo de filtragem colaborativa, temos:

- Pessoas que você talvez conheça, recomendação de amigos (Facebook).
- Recomendação de sons para escutar (Spotify).

Arquivos de Apoio

Aqui estão disponíveis os arquivos de apoio para a realização desta aula de criação de um sistema de recomendação de filmes para usuários de uma plataforma de streaming usando o algoritmo ALS: [ALS](#) e [sample movielens ratings](#).

Criação do Sistema de Recomendação

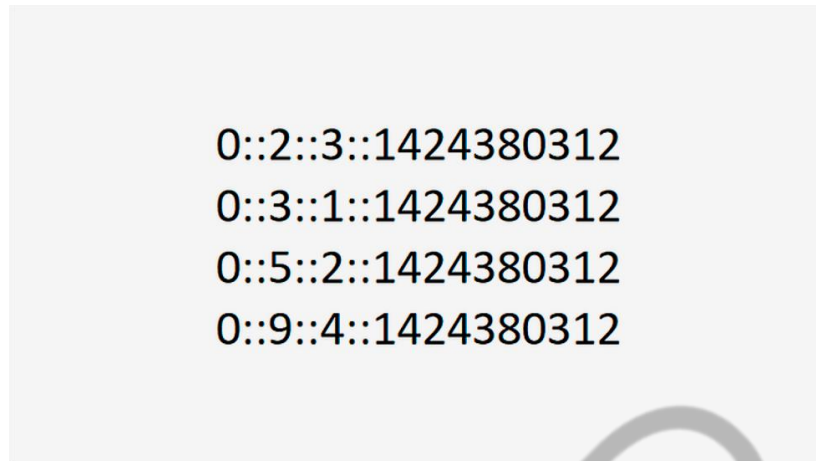
Dica: para simplificar a vida e não precisar de nenhuma instalação adicional na sua máquina, utilize o Google Colab.

Comece iniciando a sessão spark: após a preparação do ambiente e do Spark ter sido encontrado no sistema, cria-se a sessão Spark:

A screenshot of a terminal window with a dark background. At the top left, there are three colored circles (red, yellow, green) representing window control buttons. The terminal displays a single line of code in a light blue font: `spark = SparkSession.builder.master('local[*]').getOrCreate()`.

Figura 1 – Criação da sessão Spark
Fonte: Elaborado pelo autor (2024)

Carregar os dados: o conjunto de dados escolhido pertence ao Movielens. Estamos usando ele em formato txt e os dados, em cada linha, correspondem ao id do usuário, id do filme, nota e timestamp, conforme é possível ver na figura 2 – “Amostra do conjunto de dados original”.



```
0::2::3::1424380312
0::3::1::1424380312
0::5::2::1424380312
0::9::4::1424380312
```

Figura 2 - Amostra do conjunto de dados original
Fonte: Elaborado pelo autor (2024)

No código da figura 3, instanciamos os dados na variável `lines` com o RDD (Resilient Distributed Dataset), a principal estrutura de dados do Spark. Essa estrutura permite trabalhar com computação distribuída, ou seja: os dados serão distribuídos entre os nós do cluster e controlados pelo nó master. Assim, é possível processá-los em paralelo, aumentando a velocidade de processamento.



```
lines = spark.read.text("sample_movielens_ratings.txt").rdd
```

Figura 3 – Código 1
Fonte: Elaborado pelo autor (2024)

Dividir a linha em partes: como as entradas vieram juntas, na mesma linha, é necessário separar os valores a cada “::”, com o objetivo de obter um array com 4 itens. Deste modo, usa-se o método `split`, responsável por isso. Também usaremos a função `map`, que mapeia a operação, que está entre parênteses, para todas as linhas do RDD.



```
parts = lines.map(lambda row: row.value.split("::"))
```

Figura 4 – Código 2
Fonte: Elaborado pelo autor (2024)

Transformar em Row: com o módulo Row, o RDD faz a transformação para linhas do tipo Row. Essa transformação é necessária porque também serão mapeados nomes e posições das colunas, instanciando tudo na variável ratingsRDD.

```
ratingsRDD = parts.map(lambda p: Row(userId=int(p[0]), movieId=int(p[1]), rating=float(p[2]),  
timestamp=long(p[3])))
```

Figura 5 – Código 3

Fonte: Elaborado pelo autor (2024)

Apresentar dados em tabela: o próximo passo é dispor as informações em formato de tabela.

```
ratings = spark.createDataFrame(ratingsRDD)  
ratings.show()
```

Figura 6 – Código 4

Fonte: Elaborado pelo autor (2024)

Assim, obtendo a visualização da figura 7 – “DataFrame exibido com ratings.show()”.

```
+-----+-----+-----+-----+  
|movieId|rating| timestamp|userId|  
+-----+-----+-----+-----+  
|      2|   3.0|1424380312|      0|  
|      3|   1.0|1424380312|      0|  
|      5|   2.0|1424380312|      0|  
|      9|   4.0|1424380312|      0|  
|     11|   1.0|1424380312|      0|  
|     12|   2.0|1424380312|      0|  
|     15|   1.0|1424380312|      0|  
|     17|   1.0|1424380312|      0|  
|     19|   1.0|1424380312|      0|  
|     21|   1.0|1424380312|      0|  
|     23|   1.0|1424380312|      0|  
|     26|   3.0|1424380312|      0|  
|     27|   1.0|1424380312|      0|  
|     28|   1.0|1424380312|      0|  
|     29|   1.0|1424380312|      0|  
|     30|   1.0|1424380312|      0|  
|     31|   1.0|1424380312|      0|  
|     34|   1.0|1424380312|      0|  
|     37|   1.0|1424380312|      0|  
|     41|   2.0|1424380312|      0|  
+-----+-----+-----+-----+  
only showing top 20 rows
```

Figura 7 - DataFrame exibido com ratings.show()

Fonte: Elaborado pelo autor (2024)

O modelo: nesta etapa, divide-se o conjunto de dados em porções para training e test. Em seguida, instancia-se o modelo ALS, indicando os parâmetros de quantidade máxima de iterações, coeficiente de aprendizado, colunas utilizadas e desconsiderando o usuário que tiver coldstart, caso ocorra.

```
(training, test) = ratings.randomSplit([0.8, 0.2])
als = ALS(maxIter=5, regParam=0.01, userCol="userId", itemCol="movieId", ratingCol="rating",
coldStartStrategy="drop")
```

Figura 8 – Código 5
Fonte: Elaborado pelo autor (2024)

Treinamento de teste: a seguir, treinamos o modelo com o DataSet de treinamento utilizando o `als.fit()` e aplicamos o modelo no conjunto de teste para fazer as predições com `model.transform()` e avaliação do modelo.

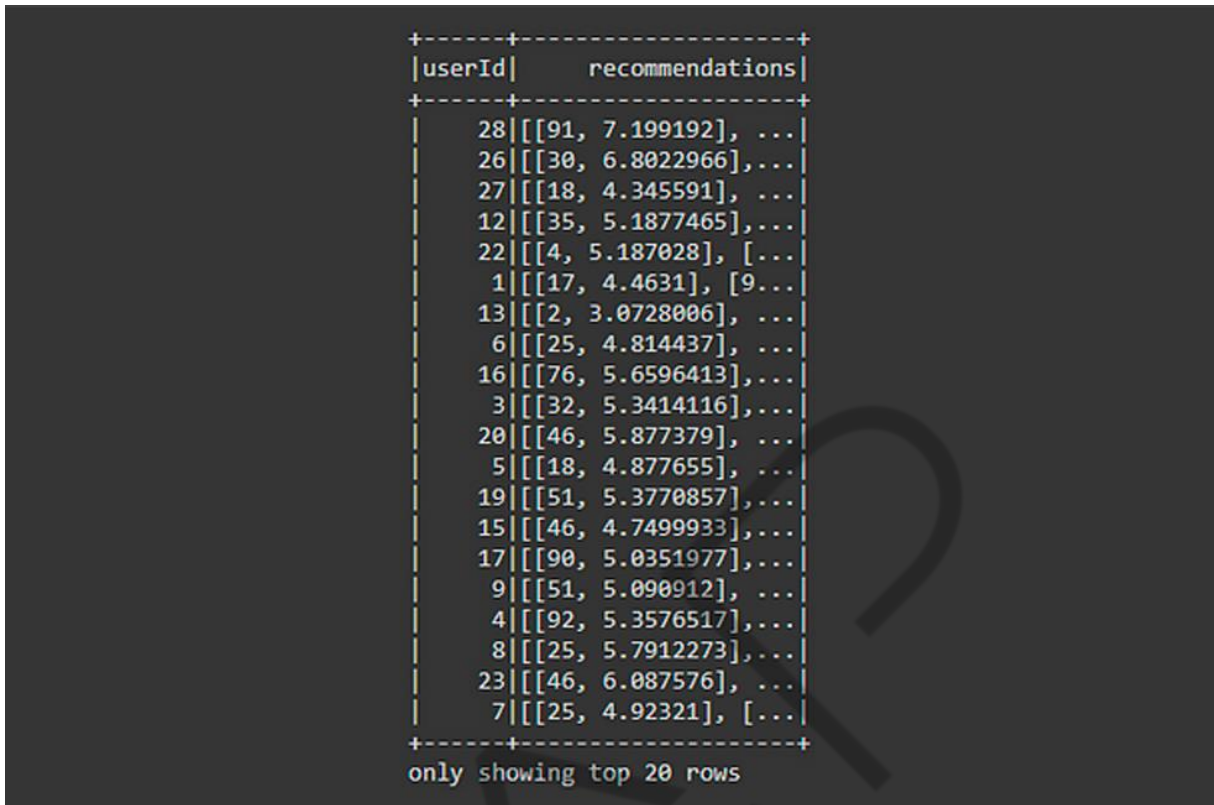
```
predictions = model.transform(test)
evaluator = RegressionEvaluator(metricName="rmse", labelCol="rating",
predictionCol="prediction")
rmse = evaluator.evaluate(predictions)
print("Erro médio quadrático = " + str(rmse))
```

Figura 9 – Código 6
Fonte: Elaborado pelo autor (2024)

Recomendações: considerando todos os usuários do conjunto de dados, geramos 10 recomendações.

```
userRec = model.recommendForAllUsers(10)
userRec.show()
```

Figura 10 – Código 7
Fonte: Elaborado pelo autor (2024)



userId	recommendations
28	[[91, 7.199192], ...]
26	[[30, 6.8022966], ...]
27	[[18, 4.345591], ...]
12	[[35, 5.1877465], ...]
22	[[4, 5.187028], [...]
1	[[17, 4.4631], [9...]
13	[[2, 3.0728006], ...]
6	[[25, 4.814437], ...]
16	[[76, 5.6596413], ...]
3	[[32, 5.3414116], ...]
20	[[46, 5.877379], ...]
5	[[18, 4.877655], ...]
19	[[51, 5.3770857], ...]
15	[[46, 4.7499933], ...]
17	[[90, 5.0351977], ...]
9	[[51, 5.090912], ...]
4	[[92, 5.3576517], ...]
8	[[25, 5.7912273], ...]
23	[[46, 6.087576], ...]
7	[[25, 4.92321], [...]

only showing top 20 rows

Figura 11 - Data frame exibido com `userRec.show()`
 Fonte: Elaborado pelo autor (2024)

Opcionalmente, foi feita a transposta da matriz de ratings a fim de recomendar usuários em potencial para itens específicos.



```
movieRecs = model.recommendForAllItems(10)
movieRecs.show()
```

Figura 12 – Código 8
 Fonte: Elaborado pelo autor (2024)

movieId	recommendations
31	[[28, 4.4335637], ...]
85	[[16, 4.4600816], ...]
65	[[23, 4.8359885], ...]
53	[[21, 4.8665752], ...]
78	[[5, 1.405179], [...]
34	[[23, 5.4059834], ...]
81	[[12, 5.1232557], ...]
28	[[18, 5.062015], ...]
76	[[16, 5.6596413], ...]
26	[[22, 4.0410576], ...]
27	[[2, 5.0788355], ...]
44	[[11, 3.253865], ...]
12	[[28, 4.8772264], ...]
91	[[28, 7.199192], ...]
22	[[26, 5.148896], ...]
93	[[27, 1.0710598], ...]
47	[[8, 4.567012], [...]
1	[[16, 4.053723], ...]
52	[[8, 4.92321], [2...]
13	[[23, 4.0366826], ...]

only showing top 20 rows

Figura 13 - Data frame exibido com movieRecs.show()

Fonte: elaborado pelo autor (2024).

Podemos, também, visualizar os filmes recomendados por usuários.

```
UserRecsOnlyItemId = userRec.select(userRec['userId'], userRec['recommendations']
['movieid'])UserRecsOnlyItemId.show(10, False)
```

Figura 14 – Código 9

Fonte: Elaborado pelo autor (2024)

userId	recommendations.movieid
28	[91, 92, 12, 81, 79, 31, 89, 49, 35, 82]
26	[30, 32, 94, 17, 22, 88, 7, 98, 90, 24]
27	[18, 2, 48, 19, 55, 66, 23, 44, 7, 33]
12	[35, 81, 17, 88, 79, 64, 69, 27, 31, 16]
22	[4, 51, 75, 74, 52, 88, 30, 9, 85, 58]
1	[17, 90, 62, 51, 69, 85, 28, 22, 38, 76]
13	[2, 52, 29, 18, 53, 9, 43, 92, 58, 83]
6	[25, 62, 51, 90, 76, 85, 58, 2, 95, 63]
16	[76, 62, 90, 29, 51, 54, 85, 1, 53, 69]
3	[32, 51, 30, 80, 7, 85, 76, 8, 29, 87]

only showing top 10 rows

Figura 15 - Data frame exibido com `UserRecsOnlyItemId.show(10, False)`
Fonte: Elaborado pelo autor (2024)

Além dessa demonstração, são inúmeras as análises que podem ser realizadas com o DataSet em questão, conforme o objetivo de exploração e utilidade a que se destina.

EMSE

O QUE VOCÊ VIU NESTA AULA?

Nesta aula, tivemos a oportunidade de conhecer mais sobre os algoritmos de recomendação ALS.

Os algoritmos de recomendação ALS (Alternating Least Squares) são uma abordagem eficaz para sistemas de recomendações que utilizam técnicas de aprendizado de máquina. Esses algoritmos são amplamente empregados em plataformas online para oferecer sugestões personalizadas aos(às) usuários(as), com base em suas preferências e histórico de interações.

O ALS opera por meio de um processo iterativo que estima a matriz de preferências de usuários e a matriz de características dos itens, minimizando a soma dos quadrados dos erros entre as avaliações reais e as estimadas.

Essa abordagem flexível e escalável torna o ALS uma escolha popular para recomendação de produtos, músicas, filmes e outros conteúdos, permitindo que as empresas aprimorem a experiência do usuário e aumentem a taxa de engajamento.

REFERÊNCIAS

BONIN, M. **Introdução à Sistemas de Recomendação**. 2023. Disponível em: <https://king.host/blog/glossario/o-que-sao-sistemas-de-recomendacao/>. Acesso em: 30 jun 2023.

CAVANI, C. **TensorFlow: Recomendação com ALS (Collaborative Filtering)**. 2017. Disponível em: <https://cirocavani.github.io/post/tensorflow-recomendacao-com-als-collaborative-filtering/>. Acesso em: 30 jun 2023.

LIAO, K. **Prototyping a Recommender System Step by Step Part 1: KNN Item-Based Collaborative Filtering**. 2018. Disponível em: <https://towardsdatascience.com/prototyping-a-recommender-system-step-by-step-part-1-knn-item-based-collaborative-filtering-637969614ea>. Acesso em: 30 jun 2023.

LIAO, K. **Prototyping a Recommender System Step by Step Part 2: Alternating Least Square (ALS) Matrix Factorization in Collaborative Filtering**. 2018. Disponível em: <https://towardsdatascience.com/prototyping-a-recommender-system-step-by-step-part-2-alternating-least-square-als-matrix-4a76c58714a1>. Acesso em: 30 jun 2023.

NEIVA, M. **Algoritmo de recomendação na prática com Python**. 2021. Disponível em: <https://www.youtube.com/watch?v=SGqX-XXGKW8>. Acesso em 30 jun 2023.

PENCHIKALA, S. **Big Data com Apache Spark - Parte 4: Spark Machine Learning**, 2020. Disponível em: <https://www.infoq.com/br/articles/apache-spark-machine-learning/>. Acesso em: 30 jun 2023.

STACK ACADEMY. **Sistemas de Recomendação**. [s.d.] Disponível em: <https://stackacademy.com.br/sistemas-de-recomendacao/>. Acesso em: 30 jun 2023.

TENSORFLOW. **Documentação Oficial**. 2023. Disponível em: <https://www.tensorflow.org/guide>. Acesso realizado em 30 jun 2023.

PALAVRAS-CHAVE

Palavras-chave: Recomendação, Python, Predição, Machine Learning, ALS, Tensorflow, KNN, SVM, Random Forest, Spark, RDD.

EMSE

The background is a dark blue field filled with numerous small, light blue dots, resembling a starry sky. Overlaid on this are several large, wavy, translucent lines in shades of blue and yellow. These lines flow from the left side towards the right, creating a sense of motion. Scattered throughout the composition are various geometric shapes: a thin vertical line, a circle containing the number '7', a small circle, an 'X' mark, a small circle, and a hexagon in the bottom right corner.

POSTECH