



RECOMMENDATION
SYSTEMS

JOSÉ LUIZ PAGNOSSIM
PROFESSOR

ESTOU PROFESSOR PAGNOSSIM



profjose.pagnossim@fiap.com.br



linkedin.com/in/jose-luiz-maturana-pagnossim-3454898



(11) 980587052

Experiência Acadêmica

- FIAP - Professor MBA - Sistemas de Recomendação (2021-atual)
- UniDrummond - Professor Graduação (2003-atual)

Formação Acadêmica

- Mestrado: Sistemas de Informação – USP
- Especialização: Engenharia de Software – Unicamp
- Graduação: Processamento de Dados - Unicamp

Experiência Profissional

- Accenture do Brasil (2008-atual): Gerente de Desenvolvimento
- Analista/Programador (desde 1996). Passagem por empresas como: Brasilprev, Itautec e IBM.

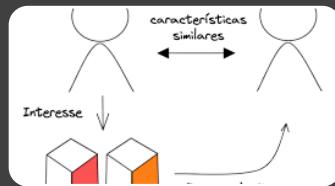


SOU: JOSÉ LUIZ MATURANA PAGNOSSIM



- Alegre
- Humanista com fé
- Trabalha duro
- Gosta de viajar
- Acredita na educação e na ciência
- Ama a família
- Inimigo do fim (*)
- Palmeirense correnteiro com orgulho

SOBRE A DISCIPLINA 3 AULAS DE 4H CADA (CARGA HORÁRIA 12H) + PROJETO SR



Aula 1

- Definição, Aplicações e Mercado
- Propriedades e Problemas
- Distância e similaridade
- Hands-on Algoritmo para Filtro Colaborativo
- Projeto SR (Kick-off)



Aula 2

- Técnicas, Algoritmos e Abordagens
- Recomendação na área financeira
- Estudo de Caso Open Finance
- Hands-on Recomendação baseada em conhecimento
- Projeto SR (Objetivo, Arquitetura e Protótipo)



Aula 3

- Avaliando (evaluate) SRs e estudo de casos recomendação de notícias
- Hands-on Gen.AI
- Tendências dos SRs e finalização da disciplina
- Prévia do Projeto SR (Objetivo, Arquitetura e Protótipo)



Projeto final

- Entrega final do Projeto de SR (Objetivos, Arquitetura, Protótipo e Implementação de uma Prova de Conceito)
- Entrega Portal: 2 semanas após aula 3

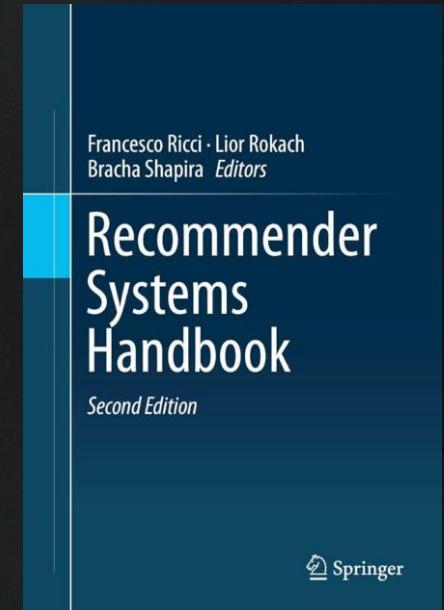
CAPÍTULO 1

Definição, aplicações e mercado

DEFINIÇÃO/**CONCEITO**

“Sistemas de Recomendação são softwares nos quais implementa-se uma ou mais técnicas que são capazes de sugerir itens para os usuários.”

[Francesco Ricci - Recommender Systems Handbook – Springer]



DEFINIÇÃO - EXEMPLO

Caso George Floyd: tropas ocupam ruas dos EUA após protestos violentos contra o racismo



pressionou o pescoço da vítima com o joelho durante uma abordagem na última segunda-feira.



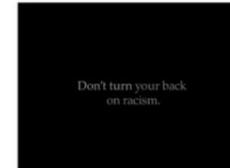
RELACIONADAS



De Obama a Kim Kardashian, personalidades protestam por George Floyd



A longa história de segregação e conflito racial em Minneapolis, onde George Floyd foi morto pela polícia



Nike lança manifesto contra o racismo e ganha apoio da Adidas



Referência: Portal Uol

Indique suas plataformas favoritas



<https://www.menti.com/wxujd139v3>

Word Cloud turma fev/23



Word Cloud turma abr/23



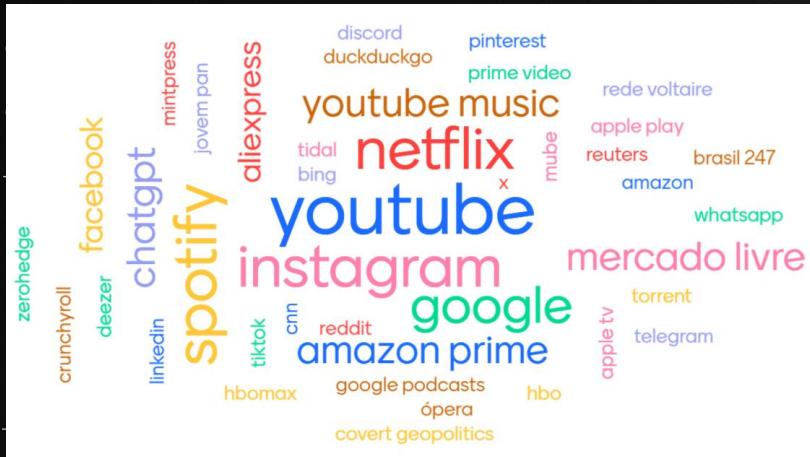
Word Cloud turma nov/22



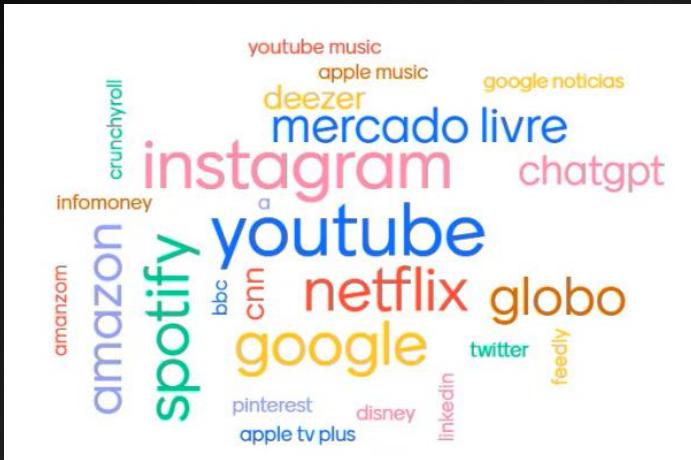
Word Cloud turma mar/23



4DTSR – FEV/24



5DTSR – SET/24



6DTS – NOV/24



7DTSR – MAR/25



FILMES, SÉRIES **E DOCUMENTÁRIOS**



NETFLIX



globoplay



apple
tv+



Disney+
STAR+



amazon
prime video



HBO
max

IMAGEM 2 – Principais Streamers

CARACTERÍSTICAS DOS PRINCIPAIS SRS



IMAGEM 3 - NETFLIX

NETFLIX

- Similaridade de Características;
- Similaridade entre usuários;
- Popularidade;
- Personalização
- Diversidade
- Serendipidade
- Features: Surpreenda-me e Relevância.

CARACTERÍSTICAS DOS PRINCIPAIS SRS

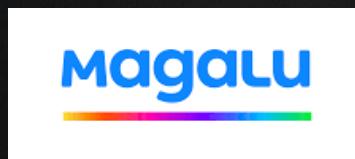
NETFLIX

 netflixtechblog.com/netflix-recommendations-beyond-the-5-stars-part-1-55838468f429

We have adapted our personalization algorithms to this new scenario in such a way that now 75% of what people watch is from some sort of recommendation. We reached this point by continuously optimizing the member experience and have measured significant gains in member satisfaction whenever we improved the personalization for our members. Let us now walk you through some of the techniques and approaches that we use to produce these recommendations.

IMAGEM 4 - NETFLIX RECOMENDADOR

COMÉRCIO, PRODUTOS, MERCADO E ALIMENTAÇÃO/ENTREGA



CARACTERÍSTICAS DOS PRINCIPAIS SRS

The screenshot shows a product page for a Samsung Smart TV. The main product is a 65" Crystal 4K Samsung 65AU7700 Wi-Fi - Bluetooth HDR Alexa-enabled TV, priced at R\$ 3,598,60. Below the main product, there's a section titled "Quem viu este produto, viu estes também" (Who viewed this product, also viewed) featuring five other TV models from different brands (Samsung, LG, and TCL) with their respective prices and ratings.

Produtos patrocinados do Magalu para você:

- Smart TV 65" 4K UHD QLED TCL 65C725
- Smart TV 55" 4K UHD DLED Philips
- Smart TV 55" 4K UHD QLED TCL 55C725
- Smart TV 55" 4K LED TCL 55P635 VA Wi-Fi

MAGALU

- Similaridade de Características;
- Similaridade entre usuários;
- Popularidade;
- Tempo no produto;
- Navegação/clicks/favoritos.

IMAGEM 6 - MAGALU

CARACTERÍSTICAS DOS PRINCIPAIS SRS



Entre tantas plataformas de e-commerce, utilizando ferramentas similares, parece-nos que a Amazon é a plataforma de ponta que melhor o faz. Basta mencionar que **35% da receita da Amazon.com** é gerada por seu mecanismo de **recomendação** com uma solução de IA – Inteligência Artificial altamente adaptável.

escoladeecommerce.com/artigos/sucesso-do-sistema-de-recomendacao-da-amazon/

Chama a atenção o fato deles creditarem a seus sistemas de recomendação o **aumento de 29% no total de vendas**, adicionando 135.99 bilhões de vendas anuais em 2016.

IMAGEM 7 - AMAZON

AMAZON

- Similaridade de Características;
- Navegação/Clicks;
- Compra;
- 35% das vendas originadas por recomendação;
- 29% aumento em vendas pelo SR (2016).

MÚSICA, RÁDIO, **AUDIO** e **VÍDEO**



CARACTERÍSTICAS DOS PRINCIPAIS SRS



IMAGEM 9 - YOUTUBE

YOUTUBE

- Watch time;
- Popularidade;
- Frequência de postagem do youtuber.

CARACTERÍSTICAS DOS PRINCIPAIS SRS

SPOTIFY

- Similaridade de Características;
- Similaridade entre usuários;
- Áudio time;
- Privilegia artistas que publicam a cada 4 semanas;
- Serendipidade / Personalização
- Nunca pára de recomendar
- Feature: Enriquecimento de playlist e ordem aleatória inteligente.



IMAGEM 10 - SPOTIFY

MISCELÂNEA

SOCIAL, CONTATOS e
AMIZADES



CONTEÚDO,
NOTÍCIAS

FOLHA DE S.PAULO

ESTADÃO

g1

UOL

Google News

VIAGENS e
LAZER



Booking



PESQUISA,
CONSULTA e BUSCA



bing

Baidu

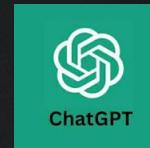
Yandex



CHAT / GEN. AI



Gemini



Copilot

智谱·AI
zhipuai.cn

EDUCAÇÃO

alura

Udemy



Sala de aula na
China

IMPACTOS DOS SRS **NA SOCIEDADE**

IMPACTOS

- Os SRS impactam a sociedade como um todo: consumo, preferências, amizades, relações, comportamento, carreira, lazer, eleições e tomadas de decisões em geral.



IMAGEM 14 - IMPACTOS

ENERGIZE – SR CAP 1

Join at [menti.com](#) | use code 6503 0101



<https://www.menti.com/al9bw86z1rr9>

CAPÍTULO 2

Propriedades e Problemas dos SRs

CAPÍTULO 2

2.1. Propriedades

PROPRIEDADES

ALEATORIEDADE

Escolher qualquer item para recomendar, possibilitando a combinação desta com outra propriedade.

SIMILARIDADE

Escolher itens com base no cálculo da similaridade entre eles.

PERSONALIZAÇÃO DO USUÁRIO

Recomendações direcionadas para um determinado perfil. Se obtém a personalização por Modelagem de características, descoberta de preferência do usuário e histórico consumo e interação do usuário com o sistema.

PROPRIEDADES

COBERTURA

Proporção de itens recomendados pelo total de itens do seu estoque, visando explorar ao máximo sua base.

POPULARIDADE

Escolher itens a recomendar, com base em indicadores de consumo, por exemplo: volume de compras, cliques, compartilhamentos, visualizações e leituras.

NOVIDADE

Sugerir itens novos com base em data/hora de entrada deles no sistema.

PROPRIEDADES

RANQUEAMENTO

Ordenar a lista de recomendação com base em um critério iniciando a partir do melhor item encontrado.

SERENDIPIDADE

Trata-se do efeito de uma surpresa positiva causada por uma recomendação.

DIVERSIDADE

Capacidade do recomendador em sugerir itens diversos, com base em distintos critérios e propriedades.

CAPÍTULO 2

2.2. Problemas

PROBLEMAS

COLD START

Falta de capacidade do recomendador em sugerir itens por ausência de indicadores (itens ou usuários). Problema encontrado no início do SR ou em novos itens e usuários.

ALEATORIEDADE

Escolher qualquer item para recomendar, dando a sensação ao usuário de que a recomendação é irrelevante.

PREVISIBILIDADE

Escolher sempre itens com base em um mesmo critério (exemplo: similaridade e conhecimento), podendo deixar o usuário entediado com o sistema.

PROBLEMAS

FALTA DE CONHECIMENTO DO USUÁRIO

Ausência de características do usuário armazenadas pelo sistema e falta de histórico de interação do usuário com o sistema.

RANQUEAMENTO

Não encontrar uma ordem boa para listar os itens, deixando itens piores no começo e melhores no final da lista. Ou nem aplicar ranqueamento.

SURPRESA NEGATIVA

Trata-se do efeito de uma surpresa causada por uma recomendação ruim. Exemplo: um filme recomendado com base no perfil de outro usuário, que exponha um dos usuários ou desgrade o outro.

ENERGIZE - QUAL O PROBLEMA?

- Exemplos

NOTÍCIA EM LEITURA

Autor conta histórias de animais notívagos do bioma brasileiro.

Trump acusa FBI de politizar investigações.

Fake News são desafios para institutos de estatística.

RECOMENDAÇÃO DE NOTÍCIA

Nenhuma recomendação encontrada.

Brasil deve ter 600 mil novos casos de câncer por ano.

Conselho de comunicação cria comissão para avaliar projetos sobre fake News.

PROBLEMA

ENERGIZE - QUAL O PROBLEMA?

- Exemplos

NOTÍCIA EM LEITURA	RECOMENDAÇÃO DE NOTÍCIA	PROBLEMA
Autor conta histórias de animais notívagos do bioma brasileiro.	Nenhuma recomendação encontrada.	Cold start
Trump acusa FBI de politizar investigações.	Brasil deve ter 600 mil novos casos de câncer por ano.	Aleatoriedade
Fake News são desafios para institutos de estatística.	Conselho de comunicação cria comissão para avaliar projetos sobre fake News.	Previsibilidade

CAPÍTULO 3

DISTÂNCIA E SIMILARIDADE

DISTÂNCIA E SIMILARIDADE - **CONCEITOS E CÁLCULO**

ANTES DE INICIAR

Antes de calcular a distância, vamos colocar nosso conjunto de dados em uma matriz para podermos visualizar a distância entre os clientes, por meio de um gráfico de dispersão.

GRÁFICO DE DISPERSÃO

O gráfico de dispersão indica o quanto dispersos os itens estão. Dessa forma, a similaridade se dá conforme encontramos dois itens mais próximos (menos distantes).

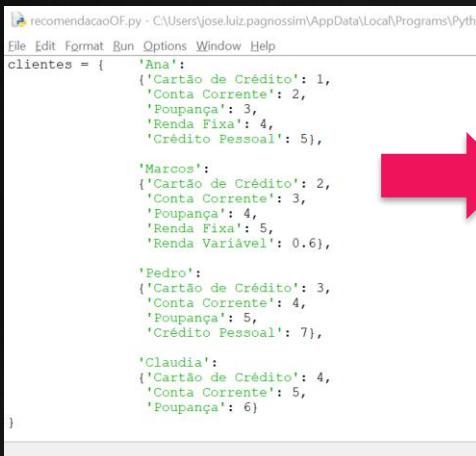
COORDENADAS

Cada relação cliente x produto vai representar um valor de referência e a partir dessa matriz, define-se o quanto dispersos os clientes estão um dos outros.

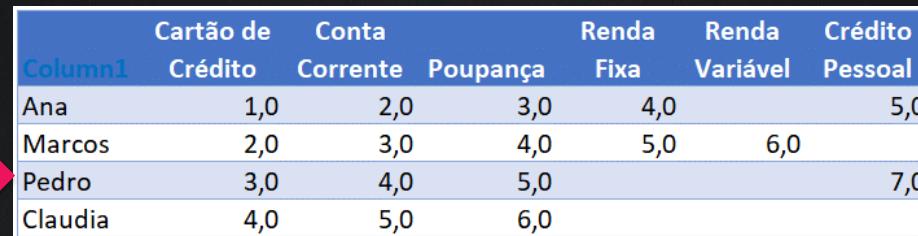
DISTÂNCIA E SIMILARIDADE - CONCEITOS E CÁLCULO

- A Matriz completa representando o data set

Partimos de uma matriz com os clientes na linha e os produtos nas colunas e o gráfico terá em um eixo um produto e no outro eixo outro produto. A coordenada {linha, coluna} representa o valores de referência do cliente para o produto.



```
recomendacaoOF.py - C:\Users\jose.luis.pagnossim\AppData\Local\Programs\Python\3.8.5\recomendacaoOF.py
File Edit Format Run Options Window Help
clientes = {
    'Ana': {
        'Cartão de Crédito': 1,
        'Conta Corrente': 2,
        'Poupança': 3,
        'Renda Fixa': 4,
        'Crédito Pessoal': 5,
    },
    'Marcos': {
        'Cartão de Crédito': 2,
        'Conta Corrente': 3,
        'Poupança': 4,
        'Renda Fixa': 5,
        'Renda Variável': 0.6,
    },
    'Pedro': {
        'Cartão de Crédito': 3,
        'Conta Corrente': 4,
        'Poupança': 5,
        'Crédito Pessoal': 7,
    },
    'Claudia': {
        'Cartão de Crédito': 4,
        'Conta Corrente': 5,
        'Poupança': 6
    }
}
```



Column1	Cartão de Crédito	Conta Corrente	Poupança	Renda Fixa	Renda Variável	Crédito Pessoal
	Ana	1,0	2,0	3,0	4,0	5,0
Marcos	2,0	3,0	4,0	5,0	6,0	
Pedro	3,0	4,0	5,0			7,0
Claudia	4,0	5,0	6,0			

IMAGEM 44 – MATRIZ DATA SET

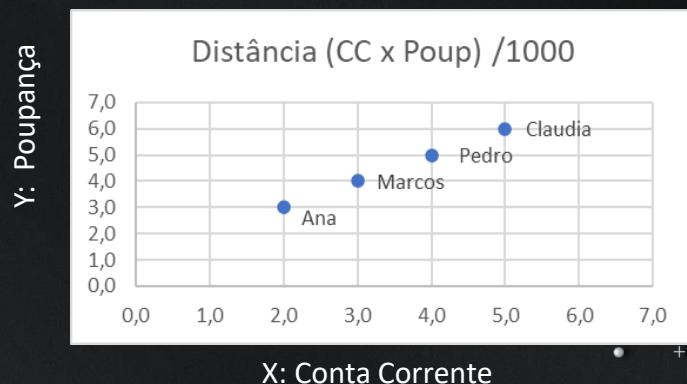
DISTÂNCIA E SIMILARIDADE - CONCEITOS E CÁLCULO

- Representando os dados em um gráfico de dispersão

Passo a passo para criar o gráfico de dispersão (scatter) em Excel:

- Create blank Graphic scatter x y;
- Select data;
- Add legend entries;
- Select series name and coordinates x, y;
- Add data label;
- Format data label, value from cell, select names, unselect x e y;
- Vídeo de apoio para criação do gráfico.
- <https://www.youtube.com/watch?v=IJLKkr4M9ko>

Conta	Corrente	Poupança
Column1		
Ana	2,0	3,0
Marcos	3,0	4,0
Pedro	4,0	5,0
Claudia	5,0	6,0



DISTÂNCIA E SIMILARIDADE - CONCEITOS E CÁLCULO

- Representando os dados em um gráfico de dispersão

Importante saber que o gráfico de dispersão, para poder realizar o cálculo, leva em consideração somente clientes que tenham produtos em comum. Caso contrário, não é possível estabelecer as coordenadas x e y.

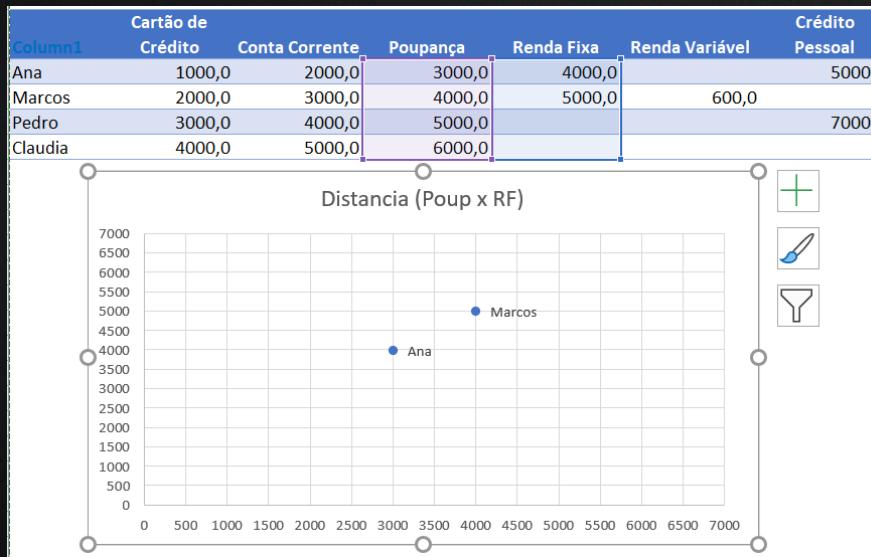


IMAGEM 46 – Gráfico 2

Obs. Note a importância de reduzir a escala para efeito de análise de dados, as coordenadas ficam distantes dificultando a análise visual dos dados.

DISTÂNCIA E SIMILARIDADE - CONCEITOS E CÁLCULO

Fórmula da Distância Euclidiana e da Similaridade

Para Distância, quanto maior o indicador, menos similar; Para Similaridade, quanto maior o indicador, mais similar.

	i = 1	i = 2
	Conta Corrente	Poupança
Ana i (y)	2,0	3,0
Marcos i (x)	3,0	4,0
Subtração	-1,0	-1,0
Elevar ao quadrado	1,00	1,00
Soma	2,00	
Raiz Quadrada (DE)	1,41	
Similaridade	0,41	

	Conta Corrente	Poupança
Ana i (y)	2,0	3,0
Pedro i (x)	4,0	5,0
Subtração	-2,0	-2,0
Elevar ao quadrado	4,00	4,00
Soma	8,00	
Raiz Quadrada (DE)	2,83	
Similaridade	0,26	

IMAGEM 47 – Gráfico 3

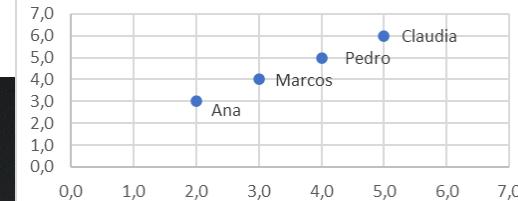
Distância Euclidiana (DE)

$$d(x, y) = \sqrt{\sum_{i=1}^n (y_i - x_i)^2}$$

Similaridade

$$\text{Sim} = 1 / (1 + DE)$$

Distância (CC x Poup) /1000



CAPÍTULO 4

Hands-on: Desenvolvimento de um algoritmo para cálculo de
similaridade em Python, criação do Data set de exemplo e preparação
para a técnica de Filtro Colaborativo

DESENVOLVENDO UM **SISTEMA DE RECOMENDAÇÃO**

Ambiente Colab

O ambiente de desenvolvimento Colab ou Colaboratory é uma plataforma da Google para desenvolvimento na nuvem utilizando linguagem Python.

Se você já tem uma conta google e já a vinculou ao Colab, excelente, basta criar um novo arquivo para usar nesse Hands-on.

Caso não tenha uma conta, você poderá criá-la e vinculá-la rapidamente por meio do link:

<https://colab.research.google.com/>

DESENVOLVENDO UM **SISTEMA DE RECOMENDAÇÃO**

Ambiente Colab

- Utilize o botão Sign in para logar, dessa forma você pode deixar seus arquivos e projetos organizados na sua conta google.
- Feito o login, crie um novo arquivo utilizando o menu
- File

New notebook

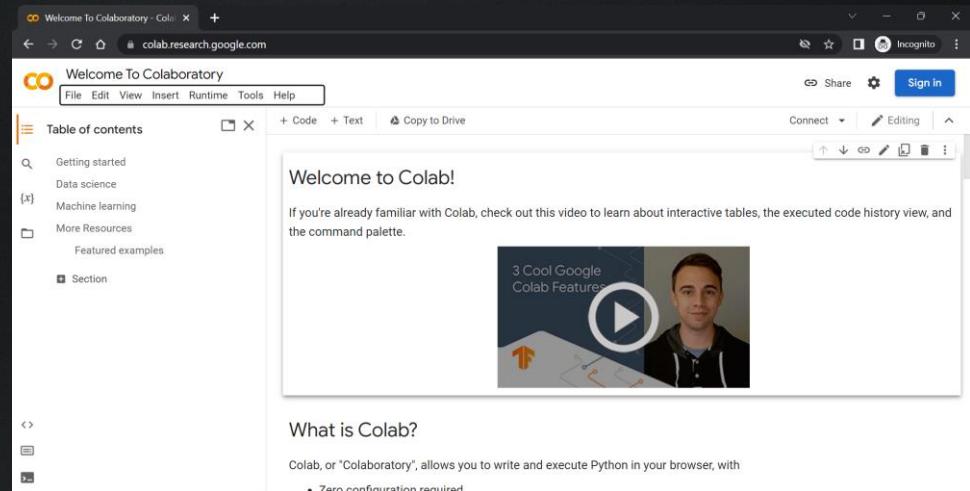


IMAGEM 36 -COLAB 1

DESENVOLVENDO UM **SISTEMA DE RECOMENDAÇÃO**

Ambiente Colab

- Pronto, você já tem um arquivo novo pronto para ser usado

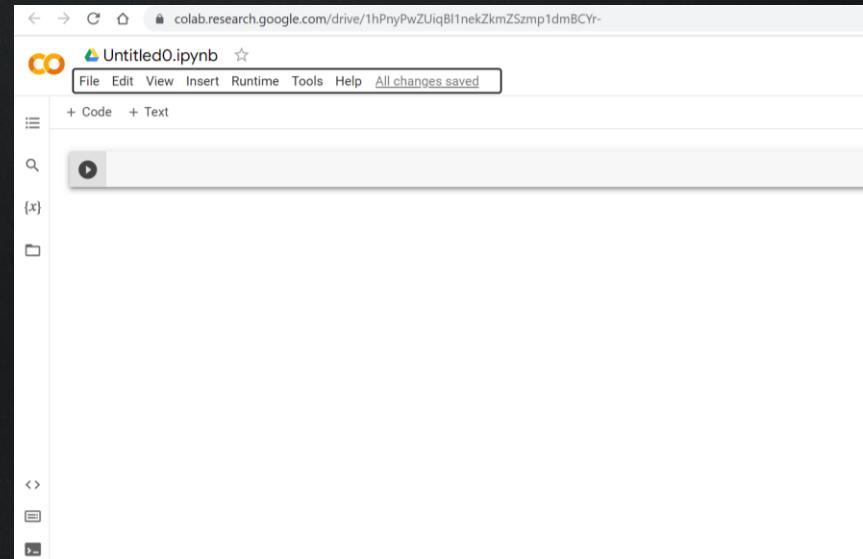


IMAGEM 37 -COLAB 2

DESENVOLVENDO UM **SISTEMA DE RECOMENDAÇÃO**

- Ambiente Colab e criação do Dataset

- Criando um Data set:

- Crie um dicionário de dados chamado "clientes", que será composto por uma relação de nomes de clientes, uma lista de produtos contratados por estes clientes em outra instituição e um valor de referência que está relacionado ao limite de crédito (no caso de cartão) e ao investimento (no caso dos outros produtos).

```
clientes = {
    'Ana': {
        'Cartão de Crédito': 1,
        'Conta Corrente': 2,
        'Poupança': 3,
        'Renda Fixa': 4,
        'Crédito Pessoal': 5},
    'Marcos': {
        'Cartão de Crédito': 2,
        'Conta Corrente': 3,
        'Poupança': 4,
        'Renda Fixa': 5,
        'Renda Variável': 0.6},
    'Pedro': {
        'Cartão de Crédito': 3,
        'Conta Corrente': 4,
        'Poupança': 5,
        'Crédito Pessoal': 7},
    'Cláudia': {
        'Cartão de Crédito': 4,
        'Conta Corrente': 5,
        'Poupança': 6}}
```

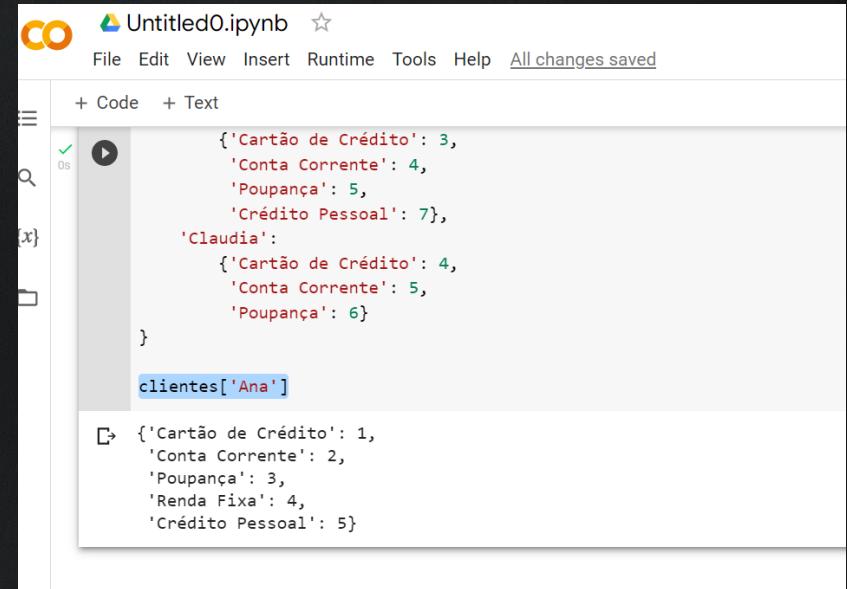
IMAGEM 42 -DATA SET

DESENVOLVENDO UM **SISTEMA DE RECOMENDAÇÃO**

- Configuração do Ambiente Python (IDLE) e criação do Data set

Testando o Data set:

- Logo abaixo da estrutura do seu Data set, digite:
`clientes['Ana']`
- Como resultado, deverá ser exibida a lista de produtos da Ana com os respectivos valores
- Obs. Lembrando que tanto o nome das estruturas quanto as chaves para busca são “case sensitive”
- Teste outras chaves dentro da lista clientes como `clientes['Pedro']`



The screenshot shows a Jupyter Notebook interface with a single cell containing Python code. The cell starts with a comment '#', followed by a dictionary assignment and a print statement. The code is as follows:

```
#  
{'Cartão de Crédito': 3,  
 'Conta Corrente': 4,  
 'Poupança': 5,  
 'Crédito Pessoal': 7},  
'Claudia':  
{'Cartão de Crédito': 4,  
 'Conta Corrente': 5,  
 'Poupança': 6}  
}  
  
clientes['Ana']  
  
{'Cartão de Crédito': 1,  
 'Conta Corrente': 2,  
 'Poupança': 3,  
 'Renda Fixa': 4,  
 'Crédito Pessoal': 5}
```

The cell has a green checkmark icon and a play button icon. The status bar at the bottom right indicates 'All changes saved'.

IMAGEM 43- DATA SET TESTE

DESENVOLVENDO UM **SISTEMA DE RECOMENDAÇÃO**

- Algoritmo para cálculo da distância euclidiana

```
from math import sqrt  
  
def euclidiana(usuario1, usuario2):  
    return sqrt(sum([pow(clientes[usuario1][item] - clientes[usuario2][item], 2)  
                    for item in clientes[usuario1] if item in clientes[usuario2]]))
```

Verifica se o item do usuário 1 existe no usuário 2

$$d(x, y) = \sqrt{\sum_{i=1}^n (y_i - x_i)^2}$$

IMAGEM 48 - Euclidiana 2

Obs. Implemente este algoritmo no mesmo arquivo do dicionário de dados para testar.

DESENVOLVENDO UM **SISTEMA DE RECOMENDAÇÃO**

- Algoritmo para cálculo da distância euclidiana

Testando o algoritmo de cálculo da função euclidiana:

Após implementar o código da função de distância, realize testes passando dois usuários para identificar a distância entre eles. Exemplo:

```
euclidiana('Ana', 'Marcos')
```

2.0

```
euclidiana('Ana', 'Pedro')
```

4.0

```
'Conta Corrente': 5,
'Poupança': 6}

from math import sqrt
def euclidiana(usuario1, usuario2):
    return sqrt(sum([pow(clientes[usuario1][item] - clientes[usuario2][item], 2)
                    for item in clientes[usuario1] if item in clientes[usuario2]]))

#Testes
#clientes['Pedro']
euclidiana('Ana', 'Marcos')
```

IMAGEM 48 – Teste Euclidiana

DESENVOLVENDO UM **SISTEMA DE RECOMENDAÇÃO**

- Algoritmo para cálculo da similaridade

```
def similaridade(usuario1, usuario2):  
    de = euclidiana(usuario1, usuario2)  
    sim = 1/(1+de) ←  
    return sim
```

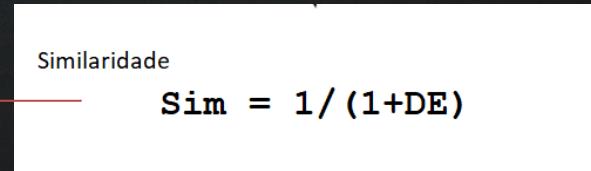


IMAGEM 49 - SIMILARIDADE

Obs. Implemente este algoritmo no mesmo arquivo do dicionário de dados para testar.

DESENVOLVENDO UM SISTEMA DE RECOMENDAÇÃO

- Algoritmo para cálculo da similaridade

- Testando o algoritmo de cálculo da função de similaridade:

- Depois de codificar a função de similaridade, execute testes para avaliar os resultados. Exemplos:

```
similaridade('Ana', 'Marcos')
```

```
0.33
```

```
similaridade('Ana', 'Pedro')
```

```
0.2
```

The screenshot shows a code editor interface with a sidebar containing icons for search, file, and folder operations. The main area displays the following Python code:

```
'Conta Corrente': 5,  
'Poupança': 6}  
}  
  
from math import sqrt  
def euclidiana(usuario1, usuario2):  
    return sqrt(sum([pow(clientes[usuario1][item]  
                    for item in clientes[us  
  
def similaridade(usuario1, usuario2):  
    de = euclidiana(usuario1, usuario2)  
    sim = 1/(1+de)  
    return sim  
  
#Testes  
#clientes['Pedro']  
#euclidiana('Ana', 'Marcos')  
similaridade('Ana', 'Marcos')
```

At the bottom of the code editor, there is an output window showing the result of the last command: `similaridade('Ana', 'Marcos')`, which is `0.3333333333333333`.

IMAGEM 49 - TESTE SIMILARIDADE

DESENVOLVENDO UM **SISTEMA DE RECOMENDAÇÃO**

Agora que já temos uma função para calcular a distância, outra para calcular a similaridade, resta uma última função que retorne uma lista de similaridade entre usuários. Esta função é a base para implementação da estratégia de filtro colaborativo.

```
def getSimilarares(usuario):
    listaSimilaridade = [(similaridade(usuario, outro), outro)
        for outro in clientes if outro != usuario]
    listaSimilaridade.sort()
    listaSimilaridade.reverse()
    return listaSimilaridade
```

- 1 Define a função que retorna a similaridade do usuário;
- 2 Cria uma lista que chama a função similaridade, passando o usuário recebido via parâmetro e varrendo a lista de clientes com outros usuários diferentes;
- 3 Ordena a lista;
- 4 Inverte a ordenação para pegar a ordem decrescente;
- 5 retorna a lista de usuários similares.

DESENVOLVENDO UM **SISTEMA DE RECOMENDAÇÃO**

- Algoritmo para retornar a lista de usuários similares

- Testando o algoritmo que retorna a lista de usuários similares e o percentual de similaridade:

- Implemente o algoritmo de listagem da lista de usuário similares e execute testes para validar, exemplo:

```
getSimilares('Ana')
```

```
def getSimilares(usuario):
    listaSimilaridade = [(similaridade(usuario, outro), outro)
                          for outro in clientes if outro != usuario]
    listaSimilaridade.sort()
    listaSimilaridade.reverse()
    return listaSimilaridade

#Testes
#clientes['Pedro']
#euclidiana('Ana', 'Marcos')
#similaridade('Ana', 'Marcos')
getSimilares('Ana')
```

```
[(0.3333333333333333, 'Marcos'),
 (0.2, 'Pedro'),
 (0.16139047779640892, 'Claudia')]
```

IMAGEM 50 - TESTE GET SIMILARES

O resultado desta função é a base para utilização da técnica de filtro colaborativo.

FIM DA AULA DE HOJE

- Antes da Chamada: vamos falar do projeto integrado
- NÃO SE ESQUEÇA DE PREENCHER A **SURVEY**
- Responda a chamada
- Estarei até o final do horário tirando dúvidas e suportando a realização do hands-on
- Qualquer dúvida posterior à aula é só me acionar pelos contatos fornecidos

PROJETO INTEGRADO

Case QuantumFinance
Sistemas de Recomendação

- • • Projeto Intgrado +
- • • Recommendation systems
- + Case Open Finance:
-



Com as primeiras etapas do Open Finance já concluídas pelas instituições financeiras, foi criada e populada uma base de dados de clientes e produtos de diferentes instituições que consentiram compartilhar seus dados via Open Finance com sua Fintech.

Com uma base de clientes e produtos financeiros é possível utilizar esses dados para oferecer produtos e serviços mais relevantes ao clientes.

Considerando o conjunto de dados de Open Finance fornecido, elabore um SR para sugerir aos clientes recomendações de serviços e/ou produtos relevantes com apoio de propriedades e técnicas estudadas na área de Recommendation Systems.

Materiais de apoio:

- Dataset reduzido compartilhado no material da aula 01 (handson).
- [Banco central – Open Finance - área do desenvolvedor](#)
- [Banco central – Open Finance - diretório de urls com dados abertos e fechados das IFs](#)

• • • • • +

Prazos (*)

3a. Aula
(previa)

+ 2 Semanas
Da 3a aula

Recomendation Systems

Utilizar o Case Open Finance ou alguma outra necessidade de Recomendação dentro da QuantumFinance para aplicar os conceitos e técnicas de Sistemas de Recomendação.

Para qualquer uma das opções, o escopo abaixo é obrigatório na entrega:

1. Elaborar um slide com um contexto da Fintech e o objetivo para utilização de um SR.
2. Elaborar um desenho técnico da arquitetura da solução, incluindo as técnicas de recomendação escolhidas e devidamente justificadas ao contexto do projeto.
3. Criar e apresentar um protótipo operacional de como o SR é aplicado ao seu projeto integrado da Fintech. Inclua cenários de uso (fictícios) e/ou interfaces gráficas para materializar o que deseja demonstrar.
4. Implementar um código-fonte que aplique ao menos uma técnica de recomendação para sua Fintech. Essa parte deve ser construída em nível de prova de conceito, apenas para demonstrar que a utilização da técnica escolhida é factível ao Dataset disponibilizado e ao contexto do projeto integrado. Essa prova de conceito será uma contribuição relevante dessa disciplina e servirá de acelerador na construção de um SR para utilização na sua Fintech.



RECOMMENDATION **SYSTEMS**

CAPÍTULO 5

Técnicas, Algoritmos e Abordagens

JOSÉ LUIZ PAGNOSSIM
PROFESSOR

CAPÍTULO 5

5.1. Técnicas

TÉCNICAS

FILTRO COLABORATIVO

- Recomenda itens a um usuário que outro usuário, similar a este, consumiu no passado.
- O termo consumiu depende da abordagem a ser adotada e pode representar uma compra, uma curtida, uma leitura ou uma música ouvida, por exemplo.
- A similaridade entre usuários é a mais comum, porém é possível estabelecer também similaridade entre itens.



IMAGEM 18 – FILTRO COLABORATIVO

TÉCNICAS

Recomendação baseada em conhecimento:

Considera o conhecimento específico (e a priori) a respeito do domínio e características dos itens.

The screenshot shows a search results page for the query "dot". The top navigation bar includes links for "BATE-PAPO", "UOL MEU NEGÓCIO", "ESTUDE ONLINE", and "PAGBANK". Below the navigation bar, there is a yellow header with categories: "PRODUTOS", "NOTÍCIAS", "CARROS", "ECONOMIA", "FOLHA", and "ESPORTE". A red arrow points upwards from the bottom left towards the "ESPORTE" category. To the right of the header, the word "dot" is displayed in large letters. Below the header, there is a large image of a black Echo Dot smart speaker. At the bottom of the page, there is a section titled "Passe o mouse para ampliar a imagem" (Move the mouse to zoom in) with a button labeled "Veja o que Alexa pode fazer" (See what Alexa can do).

Echo Dot (4^a Geração): Smart Speaker com Alexa | Música, informação e Casa Inteligente - Cor Preta

Marca: Amazon

5★ 156.966 avaliações de clientes | 1000+ perguntas respondidas

Escolha da Amazon para "echo dot 4 geração"

R\$ 379,05

à vista (5% off)
ou R\$ 399,00 em até 12x de R\$ 33,25 sem juros Ver parcelas disponíveis

Pagamentos e Segurança Enviado pela Amazon Política de devolução

Escolha uma versão Veja as diferenças

 Echo Dot (4 ^a Geração) A partir de: R\$ 399,00	 Echo Dot (4 ^a Geração Relógio) A partir de: R\$ 499,00
--	--

Cor: Preta

IMAGEM 15 - CONHECIMENTO

TÉCNICAS

- Recomendação baseada em conteúdo:

Considera palavras-chave ou o conteúdo textual de um item para ser usado como comparação e encontrar similaridade com outros itens.

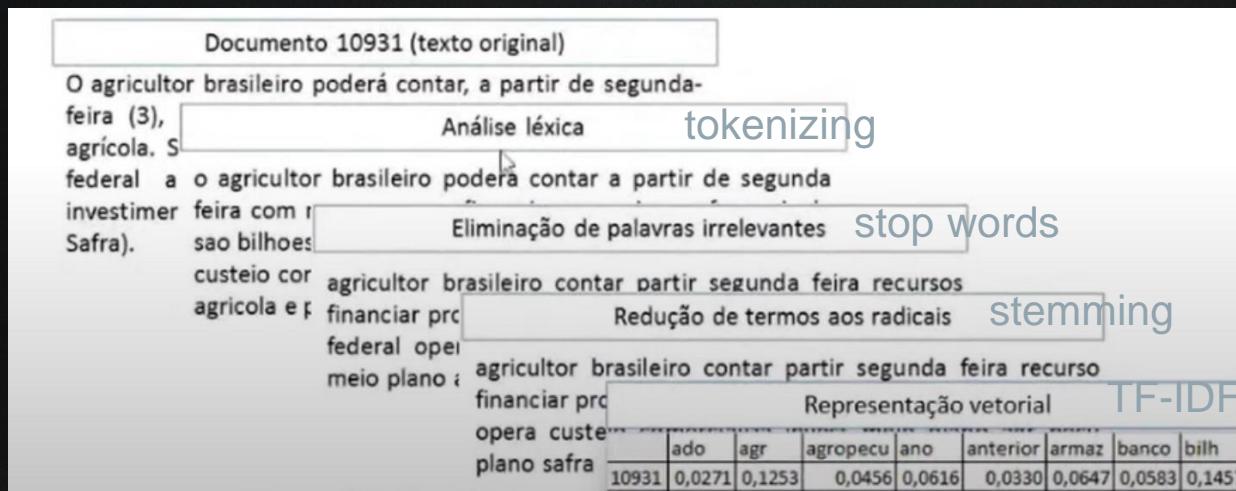


IMAGEM 16 - CONTEÚDO

TÉCNICAS

- Pré-processamento de texto:

Considera rotinas para organização, otimização e representação de textos em formatos mais estruturados como vetores, matrizes, tabelas, objetos, etc.



TÉCNICAS

- Raciocínio baseado em casos e Recomendação baseada em casos:

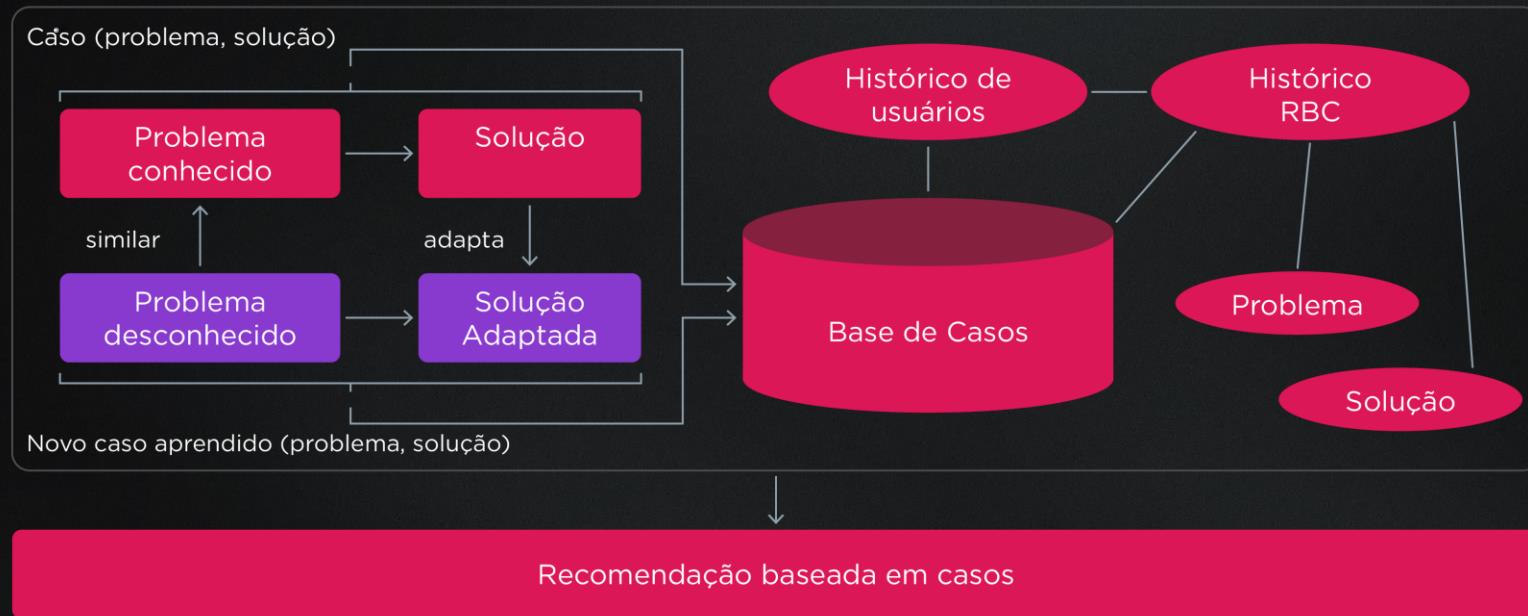


IMAGEM 19 - CASOS

Raciocínio baseado em casos considera a associação de um problema a uma solução (caso).

Recomendação baseada em casos é uma extensão do Raciocínio com modelagem que usa a recomendação baseada em conhecimento.

TÉCNICAS

- Agrupamento de Dados:

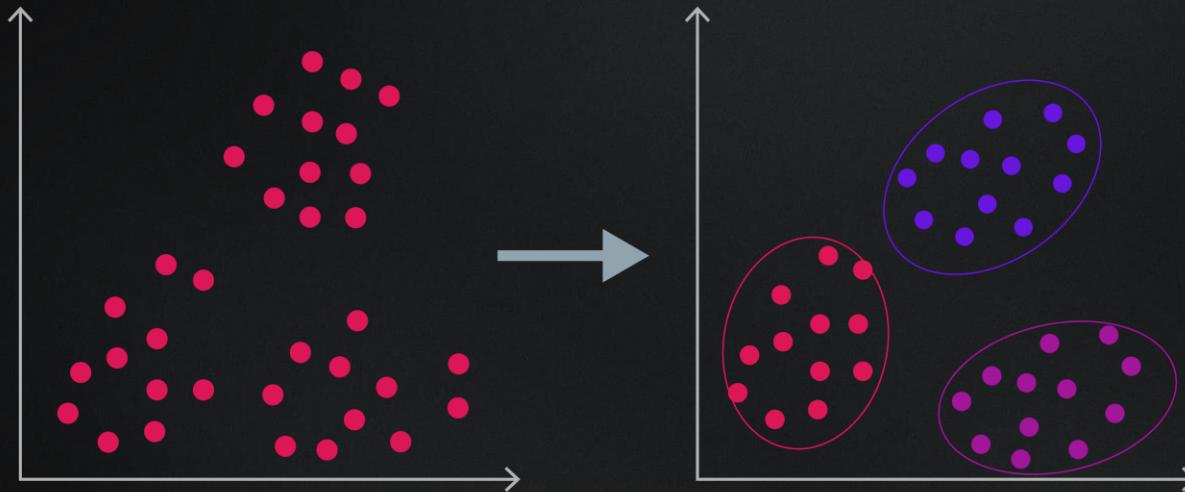


IMAGEM 20 - AGRUPAMENTO

Formar grupos sem conhecimento a priori, com base em similaridade e com objetivo de maximizar a similaridade intra grupo e minimizar a similaridade inter grupo.

TÉCNICAS

- Classificação:

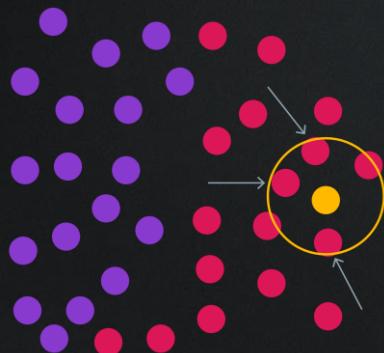


IMAGEM 22 - CLASSIFICAÇÃO

A classificação é uma técnica de aprendizagem supervisionada que usa dados rotulados com antecedência, com objetivo de fazer previsões futuras.

Isso quer dizer que treinamos os dados para que, dada a entrada de um novo item, esperamos “prever” em qual rótulo ou classe iremos classificar esse item.

TÉCNICAS

- Regras de associação:

ID	Pão	Leite	Fralda	Cerveja	Ovo	Café
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

IMAGEM 21 - ASSOCIAÇÃO

Modela os dados de forma que itens associados que pertençam a um mesmo sub-conjunto de uma base, se encontrados em uma dada ocorrência, irão implicar na ocorrência de um outro item. Como exemplo, $\{leite, fralda\} \rightarrow \{cerveja\}$ é uma associação que diz que quando se encontra os itens *leite* e *fralda* em uma ocorrência, é esperado que a *cerveja* também apareça na transação.

SISTEMAS DE RECOMENDAÇÃO

RECOMENDAÇÃO COM BASE NA CIÊNCIA DE REDE OU TEORIA DOS GRAFO



SISTEMAS DE RECOMENDAÇÃO - TÉCNICA: REDE OU GRAFO

Exemplos

REDES SOCIAIS E MÍDIAS SOCIAIS

Redes Sociais



Redes Sociais Digitais ou Mídias Sociais

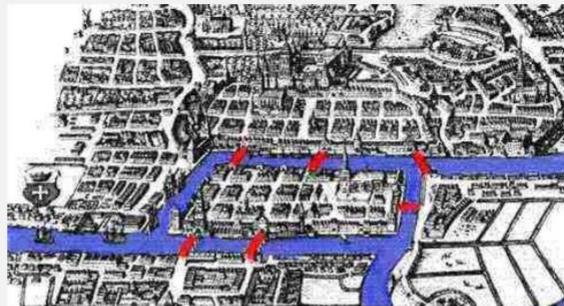


Two screenshots of a LinkedIn search interface. The top screenshot shows the search bar with 'Search' and dropdown menus for 'People' (set to '1st') and 'Locations' (set to 'Current'). Below it, a message says 'About 1,100 results'. The bottom screenshot shows the same interface but with 'People' set to '2nd' and 'Locations' set to 'Current'. Below it, a message says 'About 680,000 results'.

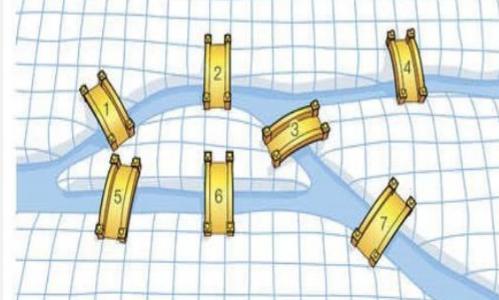
SISTEMAS DE RECOMENDAÇÃO - TÉCNICA: REDE OU GRAFO

Origem da teoria dos grafos

AS PONTES DE KÖNIGSBERG



Bridges of Königsberg



SISTEMAS DE RECOMENDAÇÃO - TÉCNICA: REDE OU GRAFO

- • • Origem da teoria dos grafos

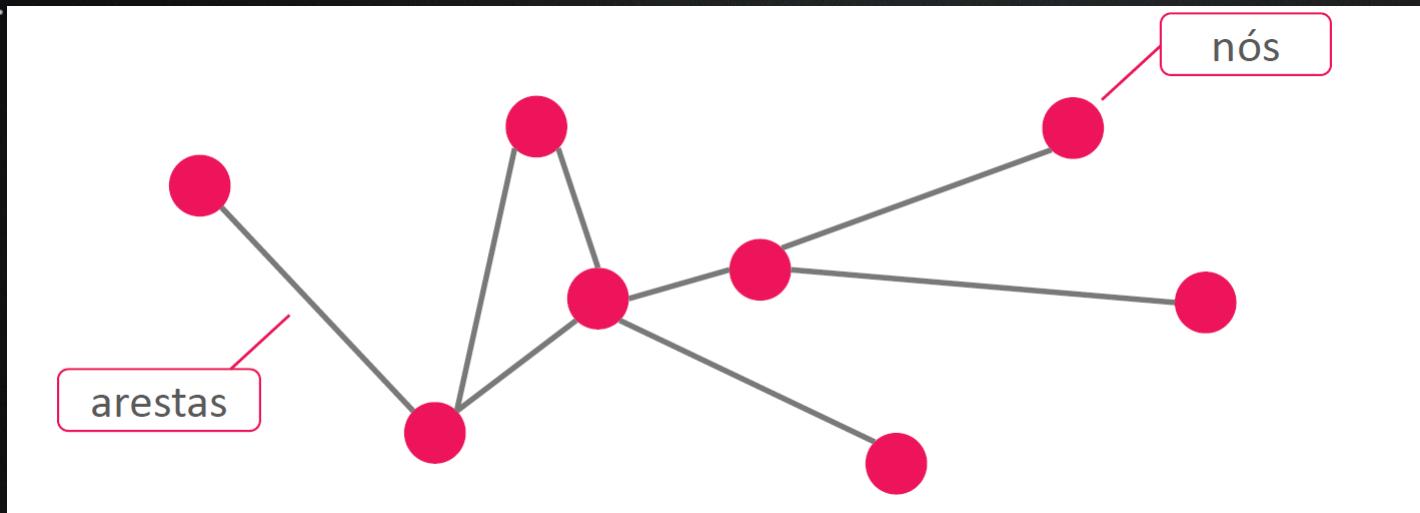
AS PONTES DE KÖNIGSBERG

O rio Pregel divide o centro da cidade de Königsberg (Prússia no século XVII, atual Kaliningrado, Rússia) em quatro regiões. Essas regiões são ligadas por um complexo de 7 pontes.

Discutia-se nas ruas da cidade a possibilidade de atravessar todas as pontes, voltando ao lugar de onde se saiu, sem repetir alguma. Havia-se tornado uma lenda popular a possibilidade da façanha, quando Euler, em 1736, provou que não existia caminho que possibilitasse tais restrições.

SISTEMAS DE RECOMENDAÇÃO - TÉCNICA: REDE OU GRAFO

Componentes

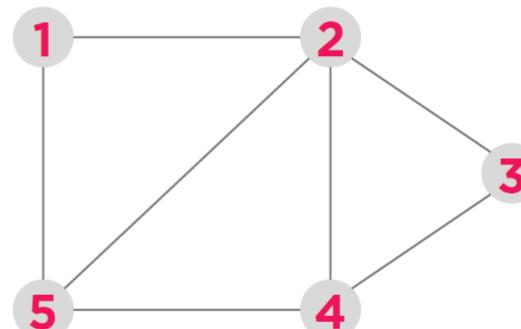


SISTEMAS DE RECOMENDAÇÃO - TÉCNICA: REDE OU GRAFO

Técnica da representação em uma estrutura de dados do tipo Matriz

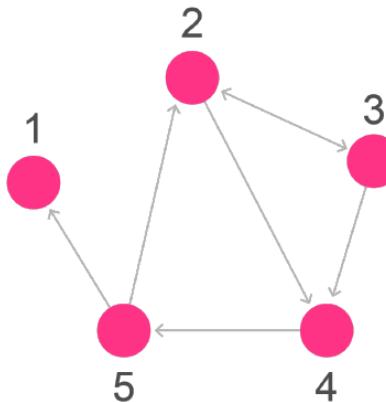
Matriz de adjacências

	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	1	1
3	0	1	0	1	0
4	0	1	1	0	1
5	1	1	0	1	0



SISTEMAS DE RECOMENDAÇÃO - ENERGIZER

Exercício: Dado o grafo a seguir e as conexões, responda

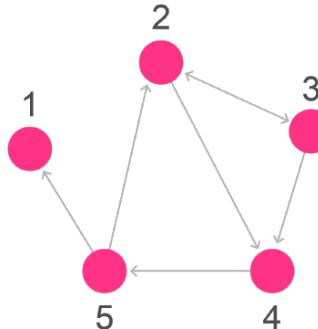


$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

- O nó 1 tem alguma conexão ?
- Quantas conexões tem o nó 2 ?
- Sugira uma nova conexão para o nó 2.

SISTEMAS DE RECOMENDAÇÃO - ENERGIZER

Exercício: Dado o grafo a seguir e as conexões, responda



$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

- a) O nó 1 tem alguma conexão ? Resposta = Não
- b) Quantas conexões tem o nó 2 ? Resposta = 2
- c) Sugira uma nova conexão para o nó 2. Resposta = R: 5, pois 2 se conecta com 3 e com 4. 3 só conhece 2 e 4, 2 é o alvo (descarta), 4 o alvo já conhece (descarta). 4 conhece 5 e 5 já segue o alvo, então 5 é a melhor recomendação de conexão ao usuário 2.

CAPÍTULO 5

5.2. Controles e Algoritmos

CONTROLES E **ALGORITMOS**

- Podemos construir um SR implementando controles relativamente simples em termos de algoritmo:

**Controle
Algoritmo**

Click
{
 onClick, +
}

Curtida
{
 onClick, +,
 -
}

Compartilhamento
{
 onClick, +
}

Tempo de
Permanência
{
 Timer()
}

**Controle
Algoritmo**

Navegação
{
 List, Tree
}

Filtro exato
{ SQL
 Where, =
}

Palavra chave
{ SQL
 Where, like
}

Modelagem de
Características
{
 Abordagem
 Relacional
}

CONTROLES E **ALGORITMOS**

- Podemos tornar um SR mais sofisticado utilizando algoritmos um pouco mais complexos:

**Controle
Algoritmo**

Preferências do usuário
{
Abordagem Relacional
}

Histórico do usuário
{
Abordagem Relacional
}

Recomend. B em conhecimento
{
Abordagem Relacional
}

**Controle
Algoritmo**

Recomend. B em conteúdo
{ Dist
Pre-proc}

Pré-
processamento
{ R (TM),
Python
(NLTK) }

Distância e Similaridade
{ Euclidiana,
Cosseno,
Sim}

CONTROLES E **ALGORITMOS**

- Podemos aperfeiçoar ainda mais um SR com uso de técnicas estudadas pela Inteligência artificial e algoritmos de Datamining e Machine Learning, o que torna o desenvolvimento mais complexo.

**Controle
Algoritmo**

Filtragem
colaborativa
{ Sim +
abordagem}

Raciocínio
Baseado em
casos
{ Case based
reasoning
(CBR) }

Recomend.
Baseada casos
{ CBR + Abord
relacional. }

**Controle
Algoritmo**

Regras de
Associação
{ Apriori ,
Abord.
relacional }

Agrupamento
{ K-Means }

Classificação
{ K-NN,
RNA}

CAPÍTULO 5

5.3. Abordagens

ABORDAGENS

Exemplo de utilização de uma abordagem híbrida:

PROBLEMA	FILTRO COLABORATIVO	CONTEÚDO	CONHECIMENTO	ABORDAGEM HÍBRIDA
COLD START DE ITEM	⬇️ ✗	⬆️ ✓	⬆️ ✓	⬆️
PREVISIBILIDADE	⬆️ ✓	⬇️ ✗	⬇️ ✗	⬆️
CONHECIMENTO DO USUÁRIO	⬆️ ✓	⬇️ ✗	⬆️ ✓	⬆️
SERENDIPIDADE	⬆️ ✓	⬇️ ✗	⬇️ ✗	⬆️

IMAGEM 22 - abordagens

No limite, a abordagem híbrida empata com a eficácia de uma técnica isolada, mas ao longo do tempo, na média, o uso da abordagem híbrida tende a ter uma eficácia maior em relação a uma técnica individual.

TÉCNICAS, ALGORITMOS E ABORDAGENS

CONCLUSÕES

- É um consenso entre os estudiosos que o uso de uma propriedade isolada para sugerir itens, já define um SR.
- Algumas propriedades, se usadas de forma isoladas ou fora de contexto, podem trazer problemas ao SR.
- Para atacar um problema, implementamos outra propriedade que também pode trazer seus problemas específicos.
- Dessa forma, o uso de abordagem híbrida equilibra os pontos fracos de técnicas isoladas, fornecendo recomendações em diferentes cenários de problemas.

TÉCNICAS, ALGORITMOS E ABORDAGENS

CONCLUSÕES

- O uso de uma abordagem híbrida tende a minimizar os problemas e melhorar alguns aspectos da recomendação. Dessa forma, o uso do mix adequado de propriedades é o que define um bom recomendador. Por outro lado, este tipo de abordagem traz consigo um aumento no esforço de desenvolvimento, na complexidade do algoritmo e maior volume de dados e transações (maior custo), podendo ainda gerar dificuldades de performance.
- Adicionalmente, seu SR pode implementar algoritmos que suportem na ponderação e calibragem do uso das propriedades, fazendo com que seu sistema se torne mais inteligente ao longo do tempo.

CAPÍTULO 6

6.1. SR na área financeira

SISTEMAS DE RECOMENDAÇÃO NA ÁREA **FINANCEIRA**

Desafios dos SRs para área financeira: Enquanto SRs de domínios mais tradicionais, como os vistos anteriormente, privilegiam ações mais imediatas, na área financeira há questões que envolvem:

- Comprometimento de longo prazo;
- Impacto financeiro da recomendação na vida do usuário;
- Mudança de natureza dos produtos ao longo do tempo;
- Regulamentações;
- Moedas;
- Poucos indicadores de feedbacks dos usuários;
- Privacidade e leis de proteção de dados (risco de cold-start).

CAPÍTULO 6

6.2. Estudo de Caso: SR na área financeira utilizando Open Finance

ESTUDO DE CASO: ÁREA FINANCEIRA UTILIZANDO OPEN FINANCE

Objetivo: Apresentar uma proposta de arquitetura e um protótipo de um SR, utilizando Open Finance para clientes Pessoa Física do Banco Santander.

O que é o Open Finance?

Open Finance é a liberdade de compartilhar seus dados financeiros. Que dados? Ora, todo o seu histórico financeiro até hoje. Informações sobre as **contas que já pagou**, o **tipo de cartão** que você tem, seus **limites disponíveis**, os **emprestimos realizados**, **produtos contratados** muito mais. Agora, sabe por que isso é legal? Você aumenta seu poder de escolha.

Ao conhecer seus dados, as instituições financeiras de sua escolha poderão apresentar a você propostas mais atraentes, como oferecer **serviços extras na conta corrente**, **isenções em cartões**, **taxas menores** no **crédito** ou **condições melhores para renovar um seguro**, por exemplo.



IMAGEM 31 -OPEN FINANCE O QUE É

O Estudo de caso iniciou acessando o portal aberto do Santander, pesquisando sobre Open Finance e quais dados poderiam ser compartilhados para serem usados em um SR.

ESTUDO DE CASO: ÁREA FINANCEIRA UTILIZANDO OPEN FINANCE

Quais outras informações relevantes para montagem da arquitetura e do protótipo?



Flexibilidade

Em breve, você poderá movimentar todas as suas contas, por meio dos canais digitais do Santander, mesmo que sejam de outra instituição financeira, a qualquer hora e lugar. Os nossos canais digitais estarão preparados para você consultar saldo, extrato, fatura, realizar investimentos e até fazer pagamentos, e, o melhor, com toda segurança do Banco Santander.

Como será o compartilhamento de dados entre as instituições financeiras?

As instituições devem disponibilizar as informações por meio de interfaces digitais (APIs).

O que é API?

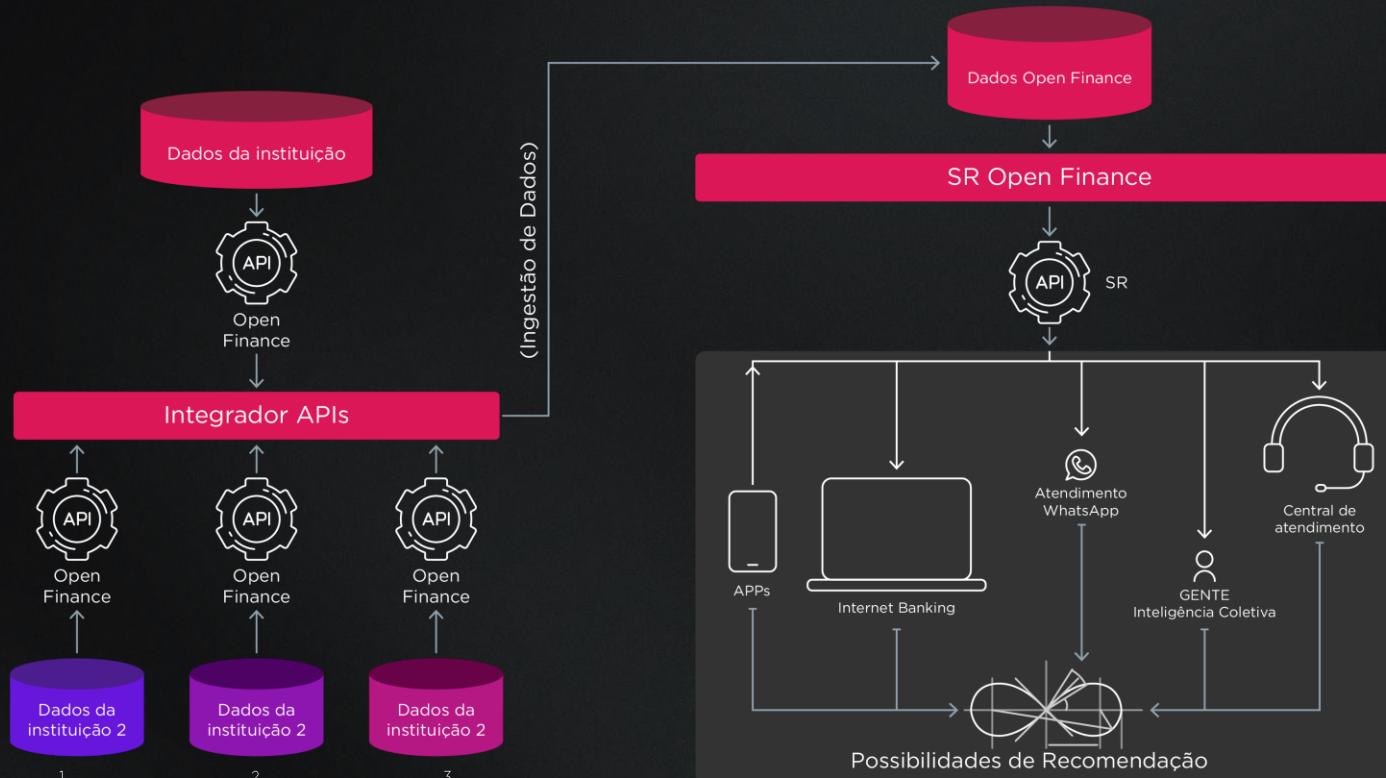
API é a sigla para “Application Programming Interface”. É uma espécie de ponte que conecta diferentes plataformas a uma mesma base. Ou seja, quando os dados passam a ser compartilhados, as instituições autorizadas conseguem trocar informações entre si de maneira segura e padronizada, seguindo as regras e parâmetros estabelecidos pelo órgão regulador.

IMAGEM 32 -OPEN FINANCE CANAIS E API

Importante considerar a disponibilização do SR nos diferentes canais do banco e utilizar o padrão API para integração e comunicação entre as instituições.

ESTUDO DE CASO: ÁREA FINANCEIRA - OPEN FINANCE

Desenho da Arquitetura proposta para o Estudo de Caso:



ESTUDO DE CASO: ÁREA FINANCEIRA **UTILIZANDO OPEN FINANCE**

PROTÓTIPO - CENÁRIO 1

Cliente Silvio Ramos, da Instituição 2, compartilhou dados Open Finance com o Santander;

O cliente já possuía uma conta no Santander.

Na Instituição 2, o cliente possui um cartão de crédito com as seguintes características:

- Limite de crédito de R\$ 3.000,00;
- Taxa de Juros Rotativa de 10% ao mês;
- Programa de pontuação de 1 ponto a cada 2 reais gastos;
- Anuidade no valor de 120,00.

O cliente acessa o Internet Banking do Santander;

Como o SR Open Finance poderia oferecer a melhor recomendação a esse cliente?

ESTUDO DE CASO: ÁREA FINANCEIRA UTILIZANDO OPEN FINANCE

Protótipo - cenário 1 - desenho proposto:

The screenshot shows the homepage of the Santander Van Gogh Open Finance prototype. The top navigation bar includes the Santander Van Gogh logo, a search bar, and user account information for Silvio Ramos. A red circle highlights the 'Cartão de Crédito' section, which displays a message about a pre-approved credit card and its benefits. To the right of this section is a large red banner with text encouraging users to fill out an investor profile analysis. Below these sections are four main service cards: Conta Corrente, Cartões de Crédito, Investimentos, and Empréstimos. The 'Cartões de Crédito' card is circled in red. The sidebar on the left lists various financial services with corresponding icons.

Santander Van Gogh

Digite aqui sua busca

A+ A- Atendimento Silvio Ramos Sair

Shopping

Conta Corrente

Pix e Transferências

Pagamentos e Recargas

Cartões

Investimentos e Poupança

Empréstimos e Financiamentos

Open Finance e Outros Produtos

Utilidades

Tenha o App Santander no seu celular e acompanhe sua conta de onde estiver.

Conta Corrente

Exibir

Cartão de Crédito

Exibir

Você tem um cartão de crédito pré-aprovado !!!
Vantagens do Cartão de Crédito Santander
- Limite de crédito de R\$ 4000,00
- Taxa de Juros Rotativa de 9% ao mês
- Programa Esfera: 1 ponto a cada 1 real gasto
- Isento de Anuidade

Solicitar

Você sabe qual é o seu perfil investidor?
Se ainda não sabe, preencha o API
(Análise de Perfil do Investidor).

PREENCHA AQUI!

Investimentos

Exibir

Empréstimos

Exibir

Favoritos

Transferências
Programar transfe...

Pagamentos
Pagar títulos e con...

Cartões
Cadastrar Bônus E...

Utilidades
2º via de comprov...

Consulte aqui a posição consolidada de seu saldo de conta corrente e seu limite contratado.

Consulte aqui a posição consolidada de seus Investimentos.

Consulte aqui seu limite de empréstimo pré aprovado e aproveite para contratar

IMAGEM 34 -OPEN FINANCE PROTOTIPO 1

ESTUDO DE CASO: ÁREA FINANCEIRA **UTILIZANDO OPEN FINANCE**

PROTÓTIPO - CENÁRIO 2

Cliente Silvia Regina, da Instituição 3, compartilhou dados Open Finance com o Santander;

A cliente já possui no Santander.

Na Instituição 3, a cliente possui uma carteira de investimentos a seguir:

- R\$ 3.000,00 na Poupança;
- R\$ 10.000,00 no CDB DI 80% do CDI;
- R\$ 2.000,00 em um Fundo de ações das empresas mais recomendadas da bolsa;
- Perfil do investidor: Moderado.

Saldo em conta corrente na Instituição 3: R\$ 8.000,00;

O cliente dentro da área logada dele acessa a área de investimentos.

Como o SR Open Finance poderia oferecer a melhor recomendação a esse cliente?

ESTUDO DE CASO: ÁREA FINANCEIRA UTILIZANDO OPEN FINANCE

Protótipo - cenário 2 - desenho proposto



The screenshot shows the "Investimentos" section of the Open Finance prototype. At the top, a dropdown menu is set to "Todas as contas e investimentos". Below it, a section titled "Meus investimentos" shows a circular chart with a small sliver of yellow and red, indicating a total value of R\$ 23.000. To the right, there's a section titled "Produtos" with a large teal circular icon. Further down, a section titled "Carteira recomendada pelo SR Open Finance" displays four investment products in a grid:

Produto	Detalhes	Valor total
CDB	DI Integrado C/C	R\$ 8.000
CDB	Pré-fixado 110% CDI	R\$ 11.000
POUPANÇA		R\$ 1.000
FUNDOS	Carteira Valor Algo+	R\$ 3.000

A large red oval highlights the "Carteira recomendada pelo SR Open Finance" section. Below it, a message states "Simulação de retorno em 1 ano: R\$ 2.000,00" and "Performance 33% melhor que sua carteira atual".

ESTUDO DE CASO: ÁREA FINANCEIRA **UTILIZANDO OPEN FINANCE**

Insights sobre recomendação na área financeira

- **Previdência:** PGBL ou VGBL, tipo de tarifa, adicionar cobertura por morte, invalidez, fundos de renda fixa, variável, mix de fundos.
- **Seguro:** Adicionar serviços de assistência, guincho, franquia de para-brisa e carro reserva.
- **Carteira de investimento:** Sugerir investimentos com base em: Perfil de investidor, histórico de investimentos, patrimônio e outros produtos contratados.
- **Empréstimos e financiamentos:** Sugerir crédito com base em: Scoring, histórico do cliente, renda e comprometimento da renda, produtos contratados, saldo em conta corrente, investimentos e uso do cartão de crédito.

ESTUDO DE CASO: ÁREA FINANCEIRA **UTILIZANDO OPEN FINANCE**

Insights sobre recomendação na área financeira

- Cartões de Crédito: Sugerir cartões, considerando: Programa de benefícios, limites, histórico, uso internacional, uso corporativo e exclusividade em pré-vendas.
- Onboarding (Cadastro de novos clientes): Reaproveitar dados cadastrais do seu cliente (de outra instituição), de forma que ele possa se cadastrar na sua instituição sem precisar preencher formulários longos, facilitando a criação de contas digitais e simplificando a jornada de aprovação de diferentes produtos ofertados.

ESTUDO DE CASO: ÁREA FINANCEIRA **UTILIZANDO OPEN FINANCE**

Conclusão sobre recomendação na área financeira

- Há muitas oportunidades de desenvolvimento para Sistemas de Recomendação em Instituições Financeiras, ainda mais com o Open Finance, que possibilita obter dados do cliente de outras instituições.
- No entanto há, também muitas possibilidades de coleta de novos dados e indicadores de pessoas e empresas que já são clientes da instituição para utilização em SR.
- Com todos os dados que a instituição já possui do cliente (ou ainda pode coletar), associados com novos dados vindos do Open Finance, imaginou quantas possibilidades para desenvolvimento e evolução dos SRs na área financeira?

CAPÍTULO 7

7.1. Hands-on: Recomendação baseada em conhecimento
Case: Open Finance

Hands-on: RECOMENDAÇÃO BASEADA EM CONHECIMENTO

Exemplo de uma aplicação Web para implementar e demonstrar as funcionalidades do SR:

Din Din Fintech
Olá [Ana ▾]. Ag. 1234 CC. 00012345-6 [Atualizar Cliente]

- Recomendação Open Finance
[

nomeCliente	nomeProduto	valor
Marcos	Renda Variável	0,6

]

- Recomendação Popularidade
[

nomeProduto	quantidadeLikes
Conta Corrente	4
Renda Variável	4
Crédito Pessoal	3
Cartão de Crédito	2
Poupança	0
Renda Fixa	0

]

Está curtindo os produtos da Din Din?
[Cartão de Crédito] [Conta Corrente] [Crédito Pessoal]
[Poupança] ; [Renda Fixa] ; [Renda Variável]

Hands-on: RECOMENDAÇÃO BASEADA EM CONHECIMENTO

- Estudo de caso: Modelando e implementando uma prova de conceito de SR para Open Finance

Modelando uma base de dados em SQL-Server - Modelo a ser implementado no exemplo:

- tbCliente: Tabela para cadastro dos clientes;
- tbProduto: Tabela para cadastro dos produtos;
- tbClienteProduto: Relacionamento entre Cliente e Produto, com o valor que o cliente tem aplicado ou financiado no produto
- tbClienteSimilaridade: Auto-relacionamento com tbCliente para associar a similaridade entre clientes.

```
clientes = {
    'Ana': {
        'Cartão de Crédito': 1,
        'Conta Corrente': 2,
        'Poupança': 3,
        'Renda Fixa': 4,
        'Crédito Pessoal': 5,
    },
    'Marcos': {
        'Cartão de Crédito': 2,
        'Conta Corrente': 3,
        'Poupança': 4,
        'Renda Fixa': 5,
        'Renda Variável': 0.6,
    },
    'Pedro': {
        'Cartão de Crédito': 3,
        'Conta Corrente': 4,
        'Poupança': 5,
        'Crédito Pessoal': 7,
    },
    'Claudia': {
        'Cartão de Crédito': 4,
        'Conta Corrente': 5,
        'Poupança': 6
    }
}
```



IMAGEM 52 – MODELO DE DADOS RELACIONAL

Hands-on: RECOMENDAÇÃO BASEADA EM CONHECIMENTO

- Script DDL para criação das tabelas

Para esse hands-on usaremos o SQLite Online [SQL OnLine IDE \(sqliteonline.com\)](https://www.sqliteonline.com)

```
CREATE TABLE [tbCliente] (
    [nomeCliente] [varchar](50) NOT NULL,
    CONSTRAINT [PK_tbCliente] PRIMARY KEY([nomeCliente] ASC));

CREATE TABLE [tbProduto] (
    [nomeProduto] [varchar](50) NOT NULL,
    [quantidadeLikes] [int] NULL,
    CONSTRAINT [PK_tbProduto] PRIMARY KEY([nomeProduto] ASC));

CREATE TABLE [tbClienteProduto] (
    [nomeCliente] [varchar](50) NOT NULL,
    [nomeProduto] [varchar](50) NOT NULL,
    [valor] [float] NULL,
    CONSTRAINT [PK_tbClienteProduto] PRIMARY KEY([nomeCliente] ASC, [nomeProduto] ASC));

CREATE TABLE [tbClienteSimilaridade] (
    [nomeClienteOrigem] [varchar](50) NOT NULL,
    [nomeClienteDestino] [varchar](50) NOT NULL,
    [similaridade] [float] NOT NULL,
    CONSTRAINT [PK_tbClienteDistancia] PRIMARY KEY([nomeClienteOrigem] ASC, [nomeClienteDestino] ASC));
```

Hands-on: RECOMENDAÇÃO BASEADA EM CONHECIMENTO

- Carregando dados pras tabelas tbCliente e tbProduto

nomeCliente
Ana
Claudia
Marcos
Pedro
*
NULL

```
INSERT INTO tbCliente (nomeCliente) VALUES ( 'Ana' );
INSERT INTO tbCliente (nomeCliente) VALUES ( 'Claudia' );
INSERT INTO tbCliente (nomeCliente) VALUES ( 'Marcos' );
INSERT INTO tbCliente (nomeCliente) VALUES ( 'Pedro' );
```

nomeProduto	quantidadeLikes
Cartão de Crédito	1
Conta Corrente	3
Crédito Pessoal	2
Poupança	0
Renda Fixa	0
Renda Variável	0
*	NULL
NULL	NULL

```
INSERT INTO tbProduto (nomeProduto,quantidadeLikes) VALUES ('Cartão de Crédito',1);
INSERT INTO tbProduto (nomeProduto,quantidadeLikes) VALUES ('Conta Corrente',3);
INSERT INTO tbProduto (nomeProduto,quantidadeLikes) VALUES ('Crédito Pessoal',2);
INSERT INTO tbProduto (nomeProduto,quantidadeLikes) VALUES ('Poupança',0);
INSERT INTO tbProduto (nomeProduto,quantidadeLikes) VALUES ('Renda Fixa',0);
INSERT INTO tbProduto (nomeProduto,quantidadeLikes) VALUES ('Renda Variável',0);
```

Hands-on: RECOMENDAÇÃO BASEADA EM CONHECIMENTO

Carregando dados pra tabela tbClienteProduto

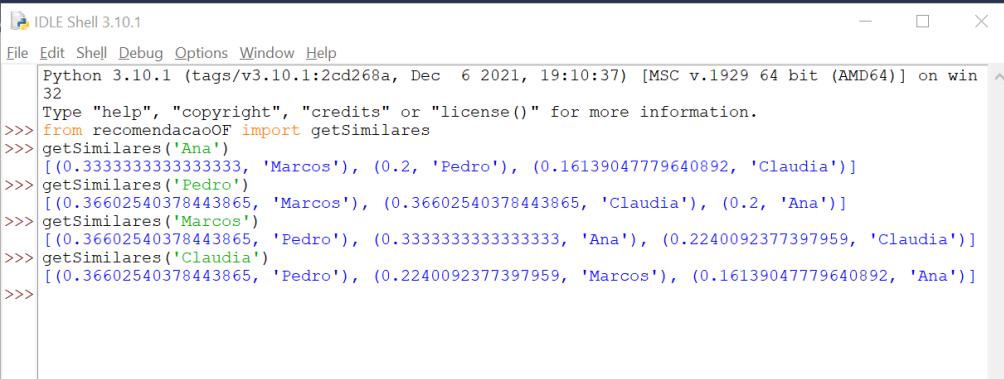
tbClienteProduto

nomeCliente	nomeProduto	valor
Ana	Cartão de Crédito	1
Ana	Conta Corrente	2
Ana	Crédito Pessoal	3
Ana	Poupança	4
Ana	Renda Fixa	5
Marcos	Cartão de Crédito	2
Marcos	Conta Corrente	3
Marcos	Poupança	4
Marcos	Renda Fixa	5
Marcos	Renda Variável	0,6
Pedro	Cartão de Crédito	3
Pedro	Conta Corrente	4
Pedro	Poupança	5
Pedro	Crédito Pessoal	7
Claudia	Cartão de Crédito	4
Claudia	Conta Corrente	5
Claudia	Poupança	6
NULL	NULL	NULL

```
INSERT INTO tbClienteProduto(nomeCliente,nomeProduto,valor) VALUES ('Ana','Cartão de Crédito',1);
INSERT INTO tbClienteProduto(nomeCliente,nomeProduto,valor) VALUES ('Ana','Conta Corrente',2);
INSERT INTO tbClienteProduto(nomeCliente,nomeProduto,valor) VALUES ('Ana','Crédito Pessoal',3);
INSERT INTO tbClienteProduto(nomeCliente,nomeProduto,valor) VALUES ('Ana','Poupança',4);
INSERT INTO tbClienteProduto(nomeCliente,nomeProduto,valor) VALUES ('Ana','Renda Fixa',5);
INSERT INTO tbClienteProduto(nomeCliente,nomeProduto,valor) VALUES ('Marcos','Cartão de Crédito',2);
INSERT INTO tbClienteProduto(nomeCliente,nomeProduto,valor) VALUES ('Marcos','Conta Corrente',3);
INSERT INTO tbClienteProduto(nomeCliente,nomeProduto,valor) VALUES ('Marcos','Poupança',4);
INSERT INTO tbClienteProduto(nomeCliente,nomeProduto,valor) VALUES ('Marcos','Renda Fixa',5);
INSERT INTO tbClienteProduto(nomeCliente,nomeProduto,valor) VALUES ('Marcos','Renda Variável',0.6);
INSERT INTO tbClienteProduto(nomeCliente,nomeProduto,valor) VALUES ('Pedro','Cartão de Crédito',3);
INSERT INTO tbClienteProduto(nomeCliente,nomeProduto,valor) VALUES ('Pedro','Conta Corrente',4);
INSERT INTO tbClienteProduto(nomeCliente,nomeProduto,valor) VALUES ('Pedro','Poupança',5);
INSERT INTO tbClienteProduto(nomeCliente,nomeProduto,valor) VALUES ('Pedro','Crédito Pessoal',7);
INSERT INTO tbClienteProduto(nomeCliente,nomeProduto,valor) VALUES ('Claudia','Cartão de Crédito',4);
INSERT INTO tbClienteProduto(nomeCliente,nomeProduto,valor) VALUES ('Claudia','Conta Corrente',5);
INSERT INTO tbClienteProduto(nomeCliente,nomeProduto,valor) VALUES ('Claudia','Poupança',6);
```

Hands-on: RECOMENDAÇÃO BASEADA EM CONHECIMENTO

Carregando os resultados do algoritmo de similaridade do Python para a tabela na BD



```
IDLE Shell 3.10.1
File Edit Shell Debug Options Window Help
Python 3.10.1 (tags/v3.10.1:2cd268a, Dec 6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> from recomendacaoOF import getSimilares
>>> getSimilares('Ana')
[(0.3333333333333333, 'Marcos'), (0.2, 'Pedro'), (0.16139047779640892, 'Claudia')]
>>> getSimilares('Pedro')
[(0.36602540378443865, 'Marcos'), (0.36602540378443865, 'Claudia'), (0.2, 'Ana')]
>>> getSimilares('Marcos')
[(0.36602540378443865, 'Pedro'), (0.3333333333333333, 'Ana'), (0.2240092377397959, 'Claudia')]
>>> getSimilares('Claudia')
[(0.36602540378443865, 'Pedro'), (0.2240092377397959, 'Marcos'), (0.16139047779640892, 'Ana')]
>>>
```

tbClienteSimilaridade



	nomeClienteOrigem	nomeClienteDestino	similaridade
1	Ana	Claudia	0,16
2	Ana	Marcos	0,33
3	Ana	Pedro	0,2
4	Claudia	Ana	0,16
5	Claudia	Marcos	0,22
6	Claudia	Pedro	0,36
7	Marcos	Ana	0,33
8	Marcos	Claudia	0,22
9	Marcos	Pedro	0,36
10	Pedro	Ana	0,2
11	Pedro	Claudia	0,36
12	Pedro	Marcos	0,36

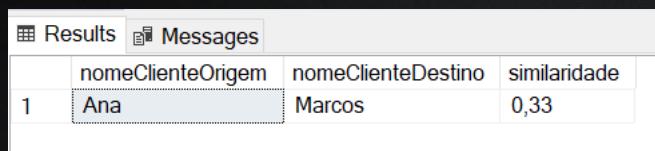
```
INSERT INTO tbClienteSimilaridade(nomeClienteOrigem,nomeClienteDestino,similaridade) VALUES ('Ana','Claudia',0.16);
INSERT INTO tbClienteSimilaridade(nomeClienteOrigem,nomeClienteDestino,similaridade) VALUES ('Ana','Marcos',0.33);
INSERT INTO tbClienteSimilaridade(nomeClienteOrigem,nomeClienteDestino,similaridade) VALUES ('Ana','Pedro',0.20);
INSERT INTO tbClienteSimilaridade(nomeClienteOrigem,nomeClienteDestino,similaridade) VALUES ('Claudia','Ana',0.16);
INSERT INTO tbClienteSimilaridade(nomeClienteOrigem,nomeClienteDestino,similaridade) VALUES ('Claudia','Marcos',0.20);
INSERT INTO tbClienteSimilaridade(nomeClienteOrigem,nomeClienteDestino,similaridade) VALUES ('Claudia','Pedro',0.36);
INSERT INTO tbClienteSimilaridade(nomeClienteOrigem,nomeClienteDestino,similaridade) VALUES ('Marcos','Ana',0.33);
INSERT INTO tbClienteSimilaridade(nomeClienteOrigem,nomeClienteDestino,similaridade) VALUES ('Marcos','Claudia',0.22);
INSERT INTO tbClienteSimilaridade(nomeClienteOrigem,nomeClienteDestino,similaridade) VALUES ('Marcos','Pedro',0.36);
INSERT INTO tbClienteSimilaridade(nomeClienteOrigem,nomeClienteDestino,similaridade) VALUES ('Pedro','Ana',0.20);
INSERT INTO tbClienteSimilaridade(nomeClienteOrigem,nomeClienteDestino,similaridade) VALUES ('Pedro','Claudia',0.36);
INSERT INTO tbClienteSimilaridade(nomeClienteOrigem,nomeClienteDestino,similaridade) VALUES ('Pedro','Marcos',0.36);
```

Hands-on: RECOMENDAÇÃO BASEADA EM CONHECIMENTO

- Vamos ver agora como implementar a técnica de Filtro Colaborativo, aproveitando o cálculo da Similaridade realizada em Python e carregada na nossa base de dados. O primeiro passo é identificar qual é o cliente mais similar em relação ao usuário alvo da recomendação.

Ex. Para a cliente alvo Ana, qual o cliente mais similar ?

```
SELECT * FROM tbClienteSimilaridade  
WHERE nomeClienteOrigem = 'Ana'  
ORDER BY similaridade DESC  
LIMIT 1;
```



The screenshot shows a database query results window. It has two tabs at the top: 'Results' (selected) and 'Messages'. The results table has three columns: 'nomeClienteOrigem', 'nomeClienteDestino', and 'similaridade'. There is one row of data: 'Ana' in the first column, 'Marcos' in the second column, and '0,33' in the third column. The row number '1' is also present in the first column.

	nomeClienteOrigem	nomeClienteDestino	similaridade
1	Ana	Marcos	0,33

Hands-on: RECOMENDAÇÃO BASEADA EM CONHECIMENTO

Agora que conhecemos o cliente com maior similaridade em relação ao cliente alvo, vamos fazer a recomendação de um produto que o cliente mais similar possui e o cliente alvo não possui (estratégia muito utilizada para filtro colaborativo).

```
SELECT b.* FROM tbCliente a, tbClienteProduto b, tbProduto c  
WHERE a.nomeCliente = b.nomeCliente AND b.nomeProduto = c.nomeProduto AND b.nomeCliente = 'Marcos'  
AND b.nomeProduto NOT IN  
(SELECT b.nomeProduto FROM tbCliente a, tbClienteProduto b, tbProduto c  
WHERE a.nomeCliente = b.nomeCliente AND b.nomeProduto = c.nomeProduto AND b.nomeCliente = 'Ana');
```

	nomeProduto	valor
1	Renda Variável	600



Resultado para destino 'Marcos' e origem 'Ana'

SELECIONE um produto do cliente mais similar ('Marcos') em relação ao cliente alvo ('Ana'), desde que o cliente alvo não possua esse produto.

Hands-on: RECOMENDAÇÃO BASEADA EM CONHECIMENTO

Utilizando a propriedade de Popularidade (likes) aplicando uma estratégia de ranqueamento. Neste caso utilizamos o atributo quantidadeLikes como um indicador de popularidade e usamos a ordenação deste atributo de forma Descrescente para criar um ranqueamento por relevância com base na em feedbacks que representam popularidade (ou seja, curtidas).

```
SELECT * FROM tbProduto ORDER BY quantidadeLikes DESC
```

	nomeProduto	quantidadeLikes
1	Conta Corrente	3
2	Crédito Pessoal	2
3	Cartão de Crédito	2
4	Poupança	0
5	Renda Fixa	0
6	Renda Variável	0

Resultado da listagem
dos produtos por
popularidade

Hands-on: RECOMENDAÇÃO BASEADA EM CONHECIMENTO

- Criando uma aplicação Web para implementar e demonstrar as funcionalidades do SR
Métodos para incrementar ou decrementar o atributo quantidadeLikes, com base na ação de like ou unlike do usuário pelos botões disponíveis para interagir com o sistema.



```
UPDATE tbProduto SET quantidadeLikes = quantidadeLikes + 1 WHERE nomeProduto = 'Cartão de Crédito'
```



```
UPDATE tbProduto SET quantidadeLikes = quantidadeLikes - 1 WHERE nomeProduto = 'Cartão de Crédito' +
```

ESPAÇO PARA O PROJETO
INTEGRADO

FIM DA AULA DE HOJE

- Apoveite para evoluir o projeto integrado de SR com seu grupo. Lembre-se de trazer para o início da próxima aula: Objetivo, Arquitetura e Protótipo.
- NÃO SE ESQUEÇA DE PREENCHER A **SURVEY**
- Esteja atento à chamada
- Qualquer dúvida só me acionar pelos contatos fornecidos

CAPÍTULO 8

8.1. Estudo de Caso: Pesquisa Acadêmica – Uma Abordagem
Híbrida para Sistemas de Recomendação de Notícias

ESTUDO DE CASO DE UM SR DE NOTÍCIAS

Interface gráfica do portal de conteúdo criado para o experimento:

[Protótipo de portal de notícias V1.0.3 - Experimento do projeto de pesquisa \(Pagnossim, J. L. M., 2017\)](#)
Usuário: Leitor anônimo ([Cadastrar um novo usuário](#)) [Sair do experimento](#)

Universo: Descoberta de planetas parecidos com a Terra desafia a ciência

Planeta Terra é rochoso, orbita uma estrela chamada sol, tem água em estado líquido, e tem vida. Essas são características que fazem a Terra se diferenciar dos outros planetas do nosso sistema solar, não é mesmo? É...Mas e de outros sistemas solares? Será que existem outras Terras? Olha, 2017 tem sido um ano marcante em relação às descobertas dos exoplanetas - são aqueles encontrados fora do nosso sistema solar. Mas, além disso, da descoberta de exoplanetas, o mais interessante é a descoberta de planetas muito parecidos com o nosso: são os chamados irmãos e até primos da Terra. De fevereiro para cá, dois momentos importantes para a astronomia. O primeiro foi o anúncio feito pela Nasa de que cientistas europeus encontraram em um único sistema solar sete planetas orbitando um tipo de estrela chamada de anã-vermelha. Destes sete, a maioria é rochosa e três têm altas chances de água na superfície. Tudo indica que eles podem ser muito parecidos com a Terra, tanto em relação ao tamanho, quanto a temperatura. E o segundo momento, agora em junho, foi o registro feito pelo telescópio Kepler: Mais de 200 planetas cintilantes, sendo 10 planetas parecidos com a Terra. Quem exalica um pouco melhor pra gente é o...

Avaliação da notícia

 Clique neste ícone se curtiu esta notícia

Avaliação da recomendação

 Clique neste ícone se ficou surpreso positivamente com a recomendação desta notícia

 Forneça uma nota para a recomendação desta notícia

 [Exibir notícias recomendadas](#)

Notícias obtidas do portal [EBC](#) (Empresa Brasil de Comunicação), mediante autorização prévia e indicação da fonte. ©

ESTUDO DE CASO DE UM SR DE NOTÍCIAS

- Exemplo recomendação com lista ranqueada (exibida em ordem aleatória para quebra de viés)

Protótipo de portal de notícias V1.0.3 - Experimento do projeto de pesquisa (Pagnossim, J. L. M., 2017)
Usuário: Leitor anônimo ([Cadastrar um novo usuário](#)) [Sair do experimento](#)

[Exibir notícias recomendadas](#)

Estudantes surdos poderão ter acesso a vídeo com prova do Enem traduzida [Ler](#)

Chuva atrapalha confecção de tapetes de Corpus Christi no Rio de Janeiro [Ler](#)

Idosa é agredida a pedradas no RJ e família denuncia intolerância religiosa [Ler](#)

Desligamento de funcionários do projeto Viver preocupa vítimas de abuso sexual [Ler](#)

Indígenas na cidade: pobreza e preconceito marcam condição de vida [Ler](#)

Maioria dos alunos não entrou em universidade por falta de dinheiro, diz estudo [Ler](#)

Nasa anuncia descoberta de novo sistema solar com sete planetas [Ler](#)

Universo: Descoberta de planetas parecidos com a Terra desafia a ciência [Ler](#)

Em dois anos, Lei do Feminicídio pune apenas uma pessoa na Bahia [Ler](#)

Brasil registra aumento de trabalho infantil entre crianças de 5 a 9 anos [Ler](#)

Notícias obtidas do portal [EBC](#) (Empresa Brasil de Comunicação), mediante autorização prévia e indicação da fonte. ©

ESTUDO DE CASO DE **UM SR DE NOTÍCIAS**

HIPÓTESES:

Se (adotar uma abordagem híbrida) então

é possível equilibrar limitações do uso de métodos isolados, proporcionando uma recomendação baseada na diversidade de propriedades e critérios de recomendação.

Se (conhecer o histórico e as preferências dos usuários) então

é possível alcançar a satisfação da maioria dos usuários.

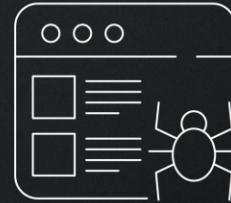
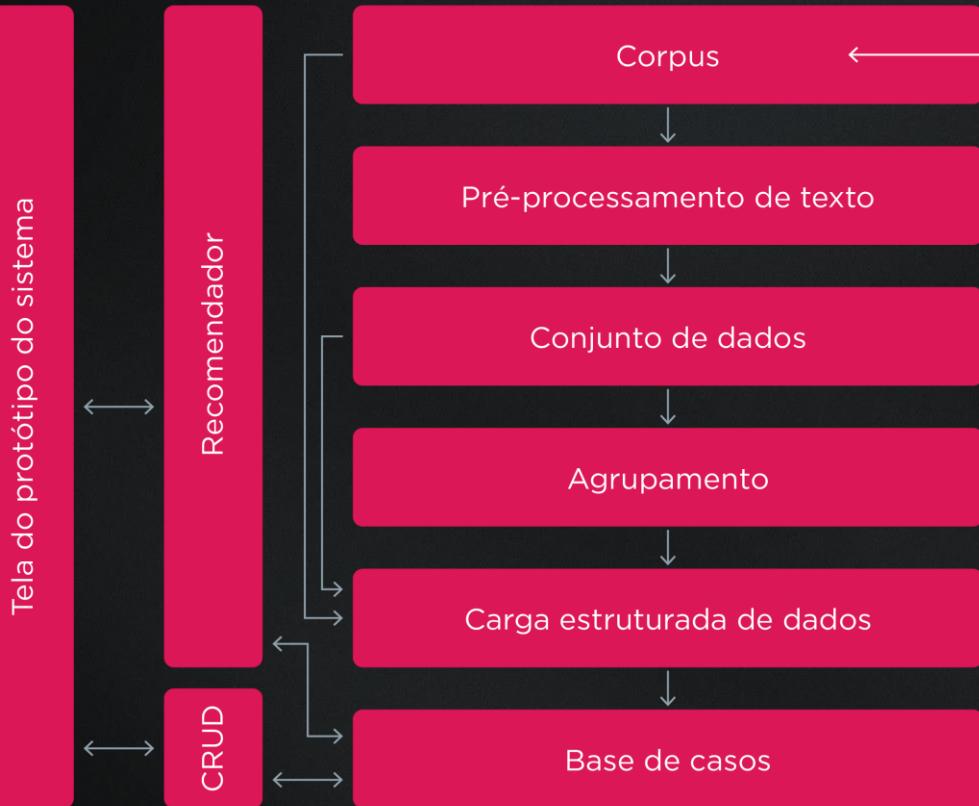
Se (implementar uma fórmula original de ranqueamento) então

é possível encontrar itens mais relevantes aos usuários.

ESTUDO DE CASO DE **UM SR DE NOTÍCIAS**

Leitor ou
Usuário

Tela do protótipo do sistema



ESTUDO DE CASO DE UM SR DE NOTÍCIAS

Do Pré-processamento de texto para a Matriz de Distância:

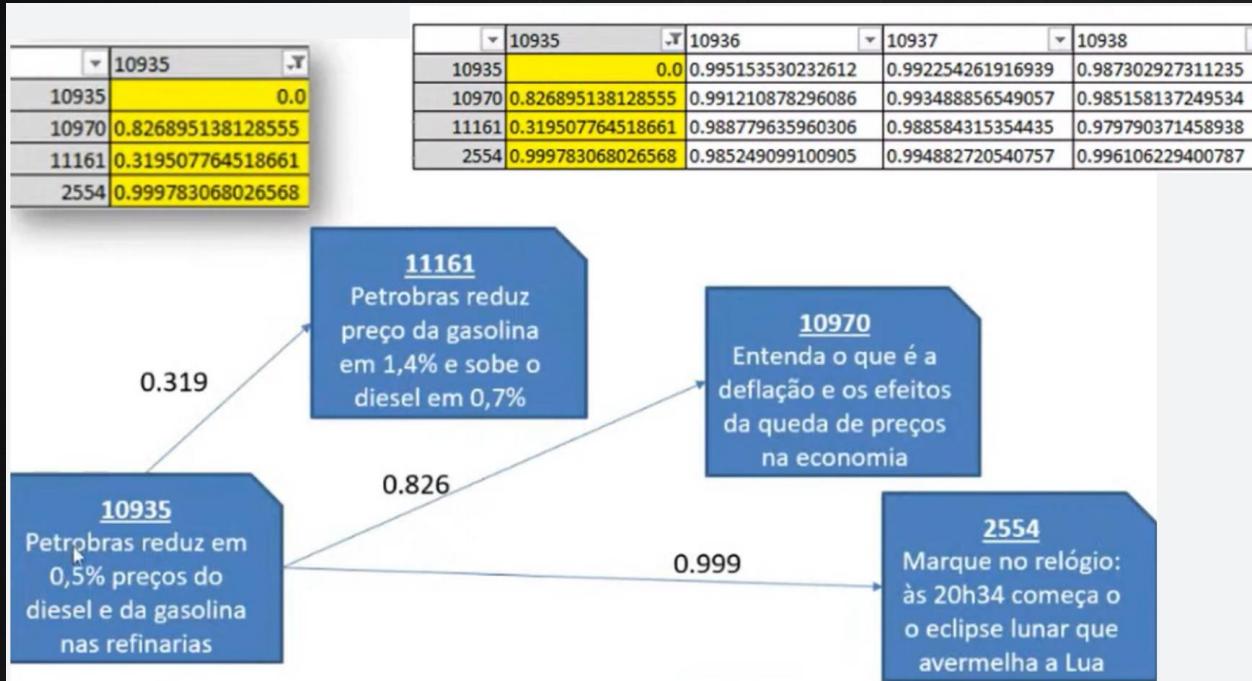
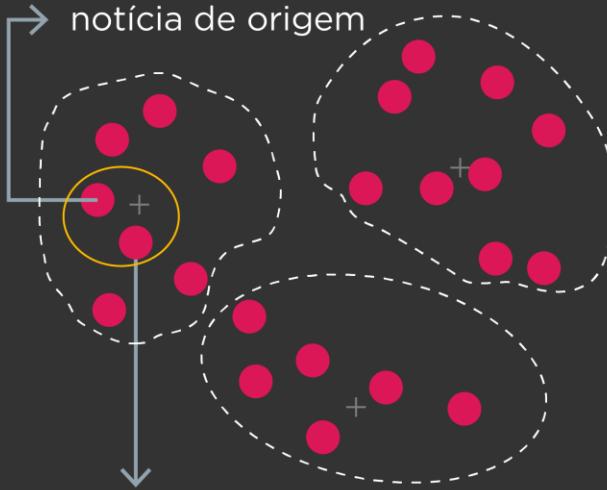


IMAGEM 24 – Matriz Distância

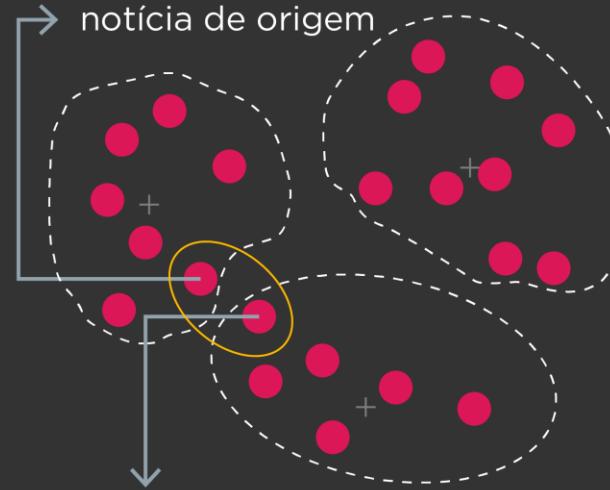
ESTUDO DE CASO DE **UM SR DE NOTÍCIAS**

Do pre-processamento de texto para o Agrupamento de Dados



notícia recomendada

Similaridade intra grupo

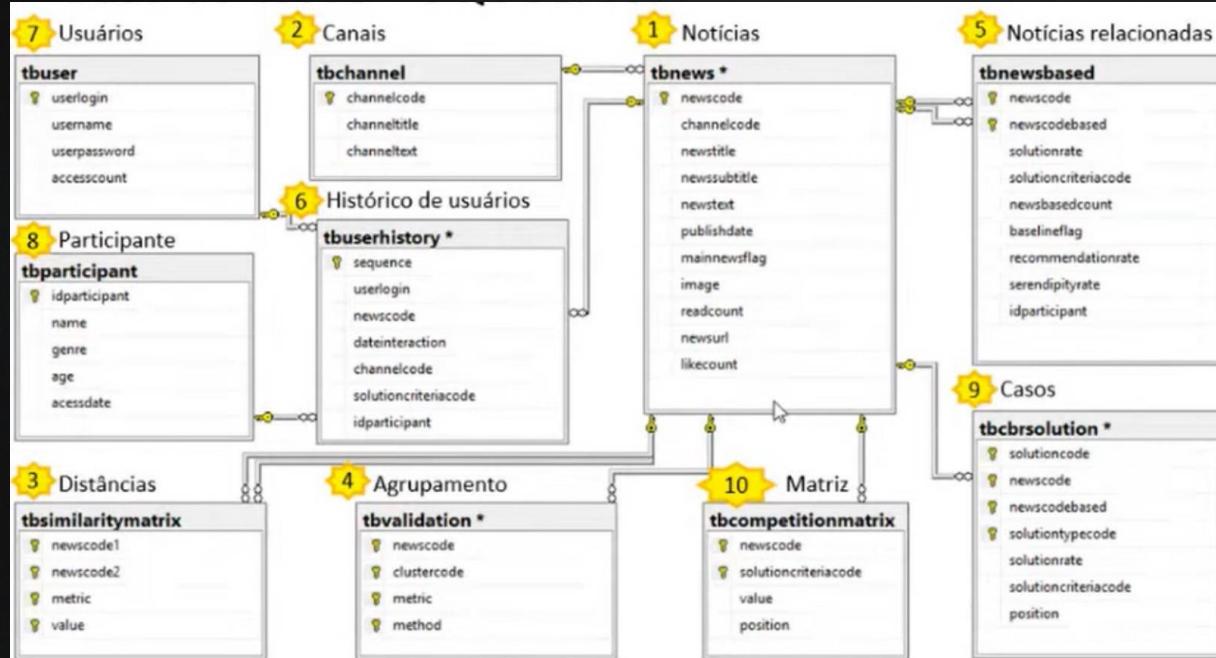


notícia recomendada

Similaridade inter grupo

ESTUDO DE CASO DE UM SR DE NOTÍCIAS

- Base de Casos implementada por meio de modelagem relacional



ESTUDO DE CASO DE UM SR DE NOTÍCIAS

Propriedades, problemas, abordagem e estratégias para recomendação:

Estratégia	Problema associado	Critérios combinados
$\{ES_1\}$	<i>Cold start</i> de item	$\{CR_1\}, \{CR_2\}, \{CR_3\}$ e $\{CR_4\}$
$\{ES_2\}$	Previsibilidade e aleatoriedade	$\{CR_5, \dots, CR_{11}\}$
$\{ES_3\}$	Falta de personalização do usuário	$\{CR_{12}\}$ e $\{CR_{13}\}$

Estratégia	Contexto da utilização
$\{ES_4\}$	Existe uma solução conhecida para o problema
$\{ES_5\}$	Trata-se de um novo problema, busca-se uma solução adaptada

Critério	Descrição
$\{CR_1\}$	Recupera \mathcal{ND} com maior <i>sim</i> de conteúdo em relação à \mathcal{NO}
$\{CR_2\}$	Recupera a \mathcal{N} com maior <i>sim</i> intergrupo
$\{CR_3\}$	Recupera a \mathcal{N} com maior <i>sim</i> intragrupo
$\{CR_4\}$	Recupera uma \mathcal{N} aleatória que não tenha sido recomendada anteriormente
$\{CR_5\}$	Recupera a \mathcal{N} mais lida dentro do canal
$\{CR_6\}$	Recupera a \mathcal{N} mais lida do portal
$\{CR_7\}$	Recupera a \mathcal{N} mais curtida do canal
$\{CR_8\}$	Recupera a \mathcal{N} mais curtida do portal
$\{CR_9\}$	Recupera \mathcal{ND} com mais recomendações aceitas em relação à \mathcal{NO}
$\{CR_{10}\}$	Recupera \mathcal{ND} com maior nota de avaliação em relação à \mathcal{NO}
$\{CR_{11}\}$	Recupera \mathcal{ND} com maior indicador de serendipidade em relação à \mathcal{NO}
$\{CR_{12}\}$	Recupera a última \mathcal{N} lida pelo \mathcal{U} mais similar
$\{CR_{13}\}$	Recupera a \mathcal{N} mais lida do \mathcal{CFU} corrente

ESTUDO DE CASO DE UM SR DE NOTÍCIAS

Rankeamento - matriz de competição.

Fórmula original MURR (Métrica unificada para ranqueamento da recomendação).

	CR1	CR2	CR3	CR4	CR5	CR6	CR7	CR8	CR9	CR10	CR11	CR12	CR13	MURR	Posição
N1	1	2	2	2	12	2	6	12	5	5	12	8	13	82	1
N12	11	4	4	11	4	11	11	4	4	3	2	1	13	83	2
N4	5	9	11	1	2	7	5	2	8	8	9	5	13	85	3
N5	4	10	9	4	1	4	4	7	9	9	8	4	13	86	4
N9	9	5	5	9	5	9	9	5	1	2	4	11	13	87	5
N2	2	1	12	3	11	3	2	11	6	6	11	7	13	88	6
N11	12	3	3	12	3	12	12	3	3	4	1	9	13	90	7
N3	3	12	1	5	10	5	3	10	7	7	10	6	13	92	8
N10	10	6	6	10	6	10	10	6	2	1	3	10	13	93	9
N7	6	8	8	6	8	6	1	8	11	11	6	2	13	94	10
N6	7	11	10	7	9	1	7	9	10	10	7	3	13	104	11
N8	8	7	7	8	7	8	8	1	12	12	5	12	13	108	12
N13	13	13	13	13	13	13	13	13	13	13	13	13	1	157	13

CAPÍTULO 9

9.1. Avaliação / validação em SRs

AVALIAÇÃO DE UM SISTEMA DE RECOMENDAÇÃO

Validação de um SR - ciclo de vida de um SR

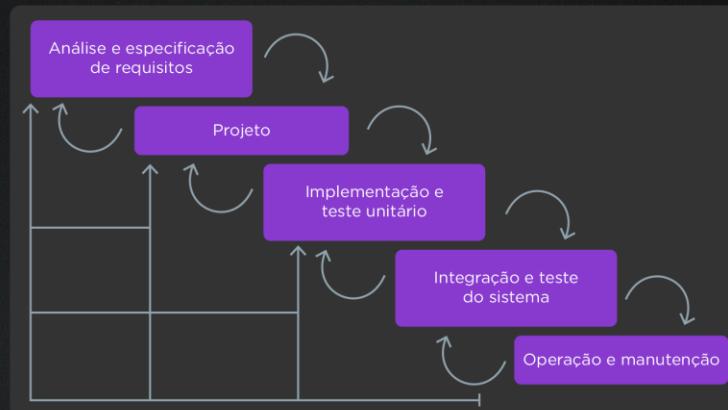
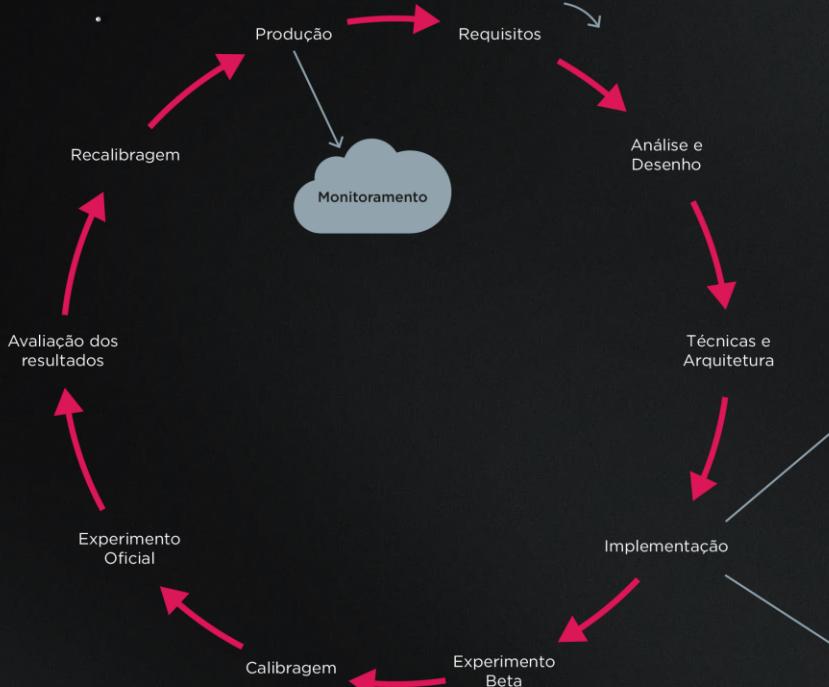


IMAGEM 70 – CICLO DE VIDA DE UM SR

AVALIAÇÃO DE UM **SISTEMA DE RECOMENDAÇÃO**

- Validação de um SR - ciclo de vida de um SR

Definição do tipo de experimento

Conhecimento dos dados

Definição das métricas que serão avaliadas

Definição dos participantes

Organização das sessões dos experimentos

Análise dos dados

Confirmação ou refutação das hipóteses

Lacunas e trabalhos futuros

AVALIAÇÃO DE UM **SISTEMA DE RECOMENDAÇÃO**

Tipos de experimento

- O experimento offline é realizado usando um conjunto de dados pré-coletados de usuários, selecionando ou classificando itens; o estudo do usuário é conduzido por meio de um conjunto de cenários de testes, estimulando os usuários a executarem tarefas de interação com o sistema. Enquanto isso, seus comportamentos são observados e coletados.
- O método de avaliação online (Gunawardana2015) consiste em medir o comportamento do usuário, quando este interage com diferentes sistemas de recomendação.
- Em um experimento online, utiliza-se alguma interface gráfica para interação com o usuário, de forma a capturar os dados de navegação durante a participação deste no experimento.
- Um experimento disponibilizado aos usuários reais, por meio de uma experiência real de interface humano/máquina, que envolve a participação de seres humanos, requer aprovação do comitê de ética “CAAE 68557417.4.0000.5390”.

GUNAWARDANA, S. Evaluating recommender systems. In: RICCI, F. e. a. (Ed.). Recommender Systems Handbook Second Edition. [S.l.]: Springer, 2015.

AVALIAÇÃO DE UM **SISTEMA DE RECOMENDAÇÃO**

Conhecendo os Dados (Corpus)

Tabela 4 – Organização do *corpus* EBC de notícias

Canal da notícia	Total de notícias
Cidadania	96
Cultura	124
Educação	257
Esporte	137
Infantil	14
Notícias	344
Tecnologia	125

Fonte: José Luiz Maturana Pagnossim, 2018

AVALIAÇÃO DE UM **SISTEMA DE RECOMENDAÇÃO**

- Formas de avaliação (métricas)

Formas de avaliação

- {AV₁} Estudo da preferência do usuário
- {AV₂} Ranqueamento de item (implementado por meio da *MURR*)
- {AV₃} Cobertura do item no espaço
- {AV₄} Cobertura do usuário no espaço
- {AV₅} *Cold start* de item
- {AV₆} Novidade
- {AV₇} Diversidade
- {AV₈} Utilidade
- {AV₉} Serendipidade
- {AV₁₀} Similaridade de conteúdo
- {AV₁₁} Similaridade entre usuários
- {AV₁₂} Similaridade intragrupos
- {AV₁₃} Similaridade intergrupos
- {AV₁₄} Popularidade
- {AV₁₅} Estratégias de recomendação

Fonte: José Luiz Maturana Pagnossim, 2018

AVALIAÇÃO DE UM **SISTEMA DE RECOMENDAÇÃO**

Definição dos participantes

- Grupo A - estudantes de graduação do curso de bacharelado de sistemas de informação da EACH/USP.
- Grupo B - estudantes de pós-graduação do curso de mestrado em sistemas de informação da EACH/USP.
- Grupo C - estudantes de graduação dos cursos de bacharelado em ciência da computação e tecnologia em análise e desenvolvimento de sistemas da Faculdade Carlos Drummond de Andrade (campus Tatuapé, São Paulo-SP).
- Grupo D - pessoas das áreas de tecnologia da informação e educação que fazem parte do convívio profissional do pesquisador.

As pessoas foram convidadas pelo pesquisador e pela orientadora desta pesquisa, por meio de uma mensagem eletrônica formal contendo os principais objetivos do experimento e endereço na internet para acesso ao sistema.

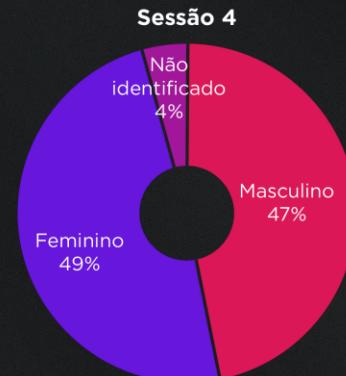
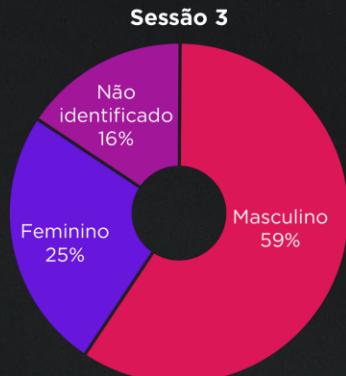
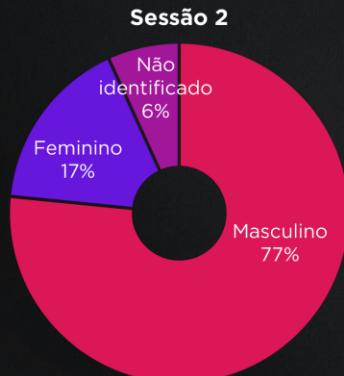
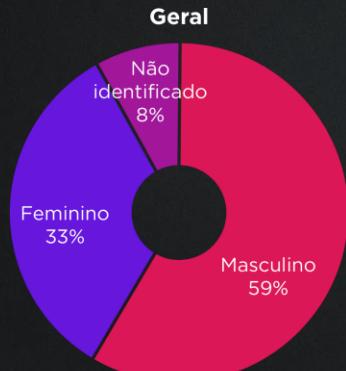
AVALIAÇÃO DE UM **SISTEMA DE RECOMENDAÇÃO**

Organização das sessões do experimento

- Sessão 1 - testes do sistema de recomendação. Para testar o sistema foi selecionado um sub-grupo do grupo B especializado em sistemas de dados. Este sub-grupo contribuiu com a identificação de melhorias. Ao final desta sessão, uma nova versão do sistema foi criada com as correções dos defeitos encontrados. Os dados dos participantes, foram descartados para efeito de apuração.
- Sessão 2 - estabelecimento de uma base inicial de navegação para comparação com as próximas sessões do experimento e análise dos efeitos de *cold start* de itens e *cold start* de usuários. Para esta sessão foi convidado um sub-grupo de pessoas selecionado a partir dos quatro grupos definidos na seção 4.2.1. Ao final do período estabelecido para essa sessão (uma semana), foram registradas trinta participações. Os dados foram separados para uso na sessão seguinte.
- Sessão 3 - comparação entre sessões em um cenário com dados já registrados pelo sistema de recomendação. Para esta sessão, foi selecionado a partir dos quatro grupos definidos (os mesmos convidados anteriormente). Foi definido o período de participação (uma semana), mas após três dias da abertura da sessão, o número de participações havia superado a sessão anterior. Neste momento, os dados para comparação com a sessão anterior foram contabilizados. A partir deste momento foram contabilizadas as participações da sessão 3.
- Sessão 4 - nesta sessão participaram pessoas que acessaram o experimento após o encerramento da sessão 3, ou seja, eram pessoas que foram convidadas para participarem da sessão 2 ou da sessão 3, mas não tinham conseguido acessar o experimento dentro dos prazos estabelecidos nas respectivas sessões. Os dados desta sessão foram separados para serem comparados com as sessões 2 e 3. Apesar do número de participações ter sido maior nesta última sessão (quarenta e nove), foram gerados indicadores, tabelas e gráficos, a partir dos dados coletados, de forma a relativizar a análise dos resultados.

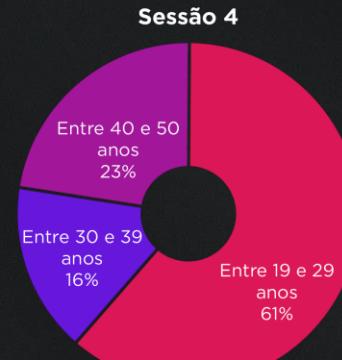
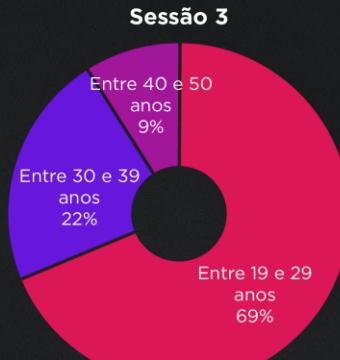
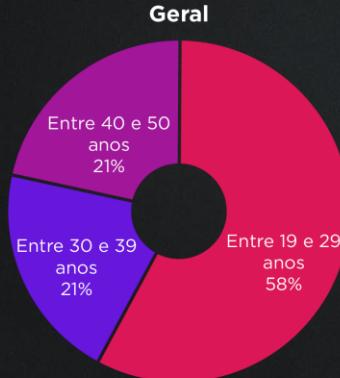
AVALIAÇÃO DE UM **SISTEMA DE RECOMENDAÇÃO**

Análise dos dados (Resultados)



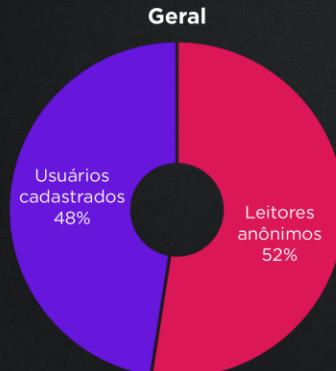
AVALIAÇÃO DE UM **SISTEMA DE RECOMENDAÇÃO**

Análise dos dados (Resultados)



AVALIAÇÃO DE UM **SISTEMA DE RECOMENDAÇÃO**

Análise dos dados (Resultados)



AVALIAÇÃO DE UM **SISTEMA DE RECOMENDAÇÃO**

Análise dos dados (Resultados)

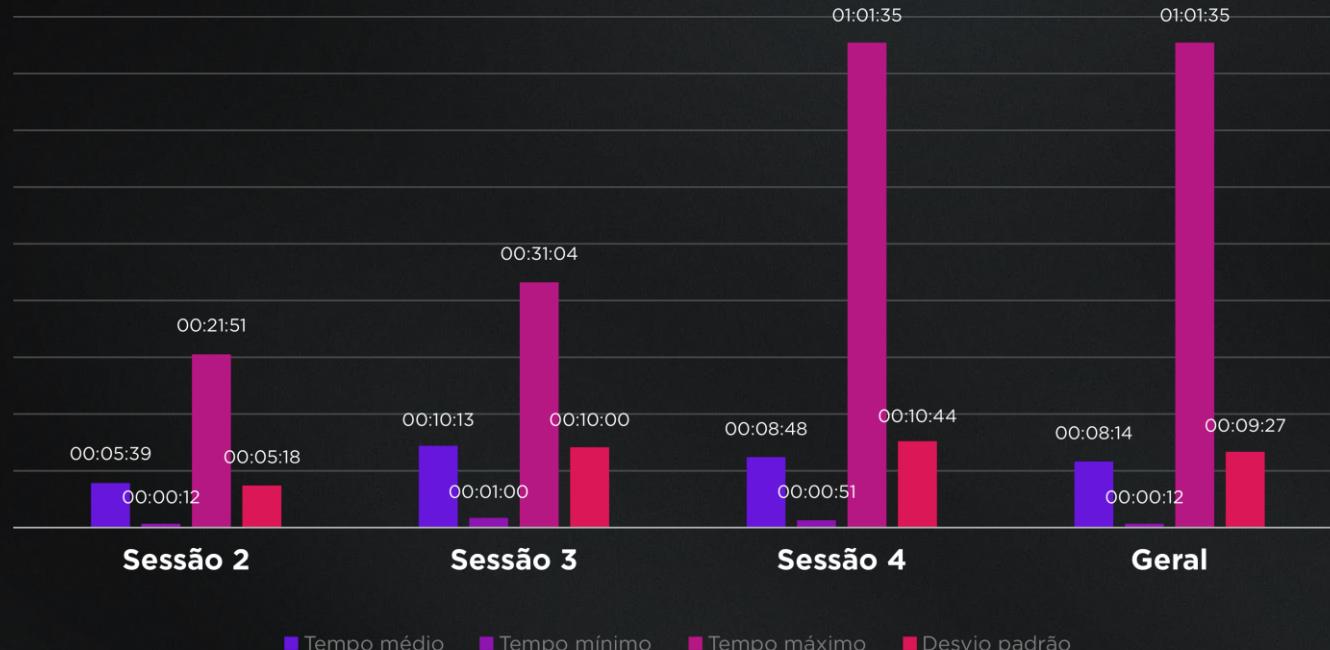


IMAGEM 75 – TEMPO DE PARTICIPAÇÃO POR SESSÃO

AVALIAÇÃO DE UM **SISTEMA DE RECOMENDAÇÃO**

- Análise dos dados (Resultados) - verificação da eficácia

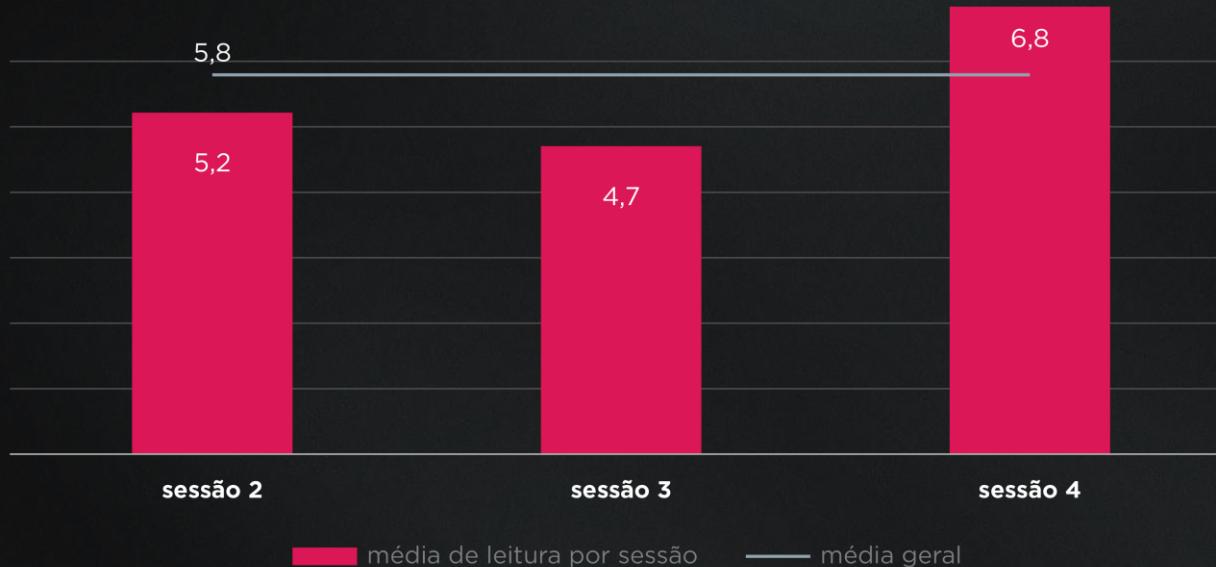


IMAGEM 76 - MÉDIA DE LEITURA POR SESSÃO

AVALIAÇÃO DE UM **SISTEMA DE RECOMENDAÇÃO**

- Análise dos dados (Resultados) - verificação da eficácia

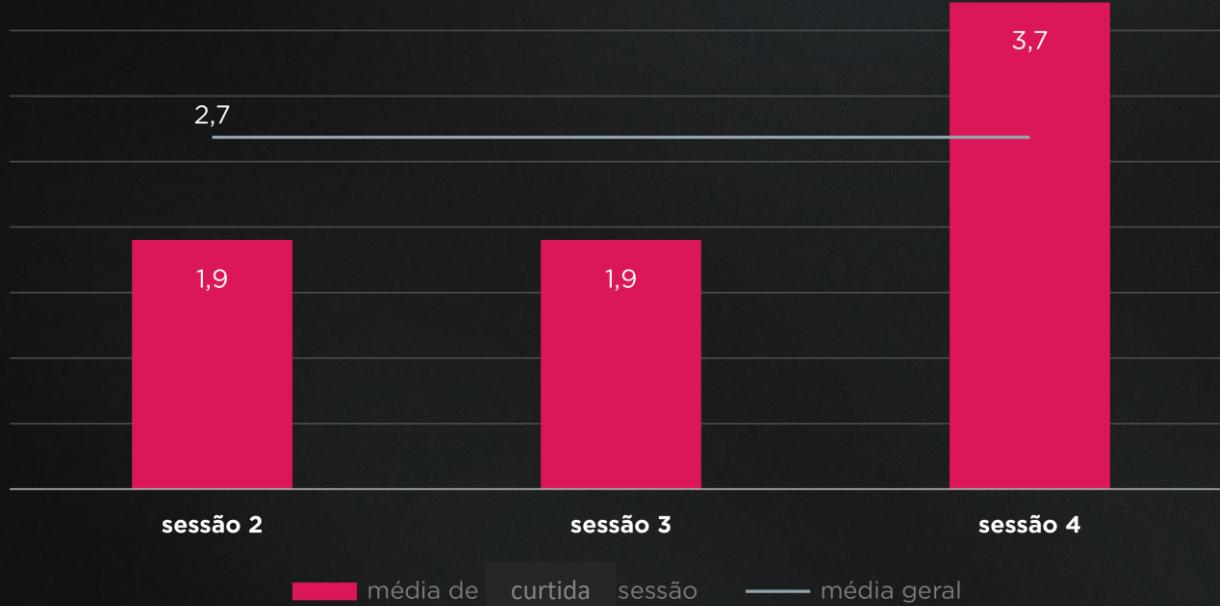


IMAGEM 77 - MÉDIA DE CURTIDA POR SESSÃO

AVALIAÇÃO DE UM **SISTEMA DE RECOMENDAÇÃO**

- Análise dos dados (Resultados) - verificação da eficácia e confirmação da primeira hipótese

Gráfico Serendipidade

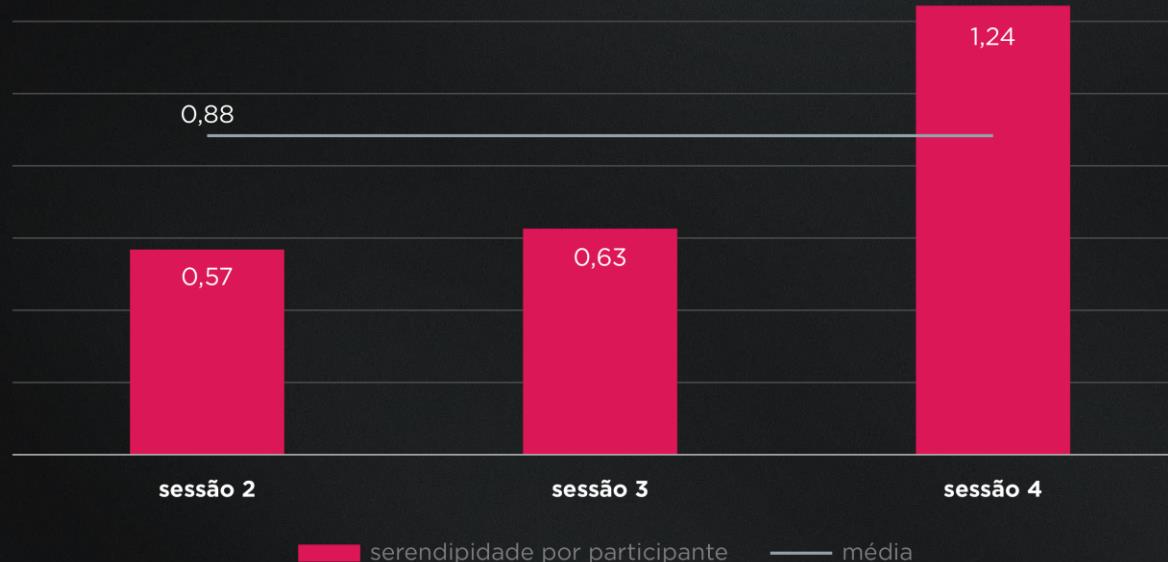


IMAGEM 78 – SERENDIPIDADE POR SESSÃO

AVALIAÇÃO DE UM **SISTEMA DE RECOMENDAÇÃO**

- Análise dos dados (Resultados) - verificação da eficácia e confirmação da primeira hipótese

Se (adotar uma abordagem híbrida) então
é possível equilibrar limitações do uso de métodos isolados



Gráfico efeito diversidade

AVALIAÇÃO DE UM **SISTEMA DE RECOMENDAÇÃO**

- Análise dos dados (Resultados) - verificação da eficácia e confirmação da segunda hipótese

Se (conhecer o histórico e as preferências dos usuários) então
é possível aumentar a satisfação dos usuários

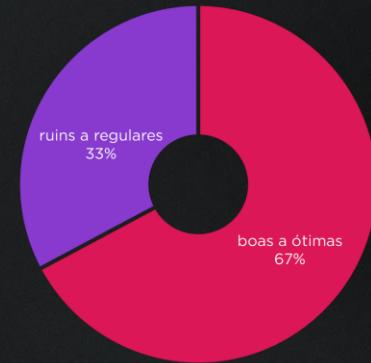
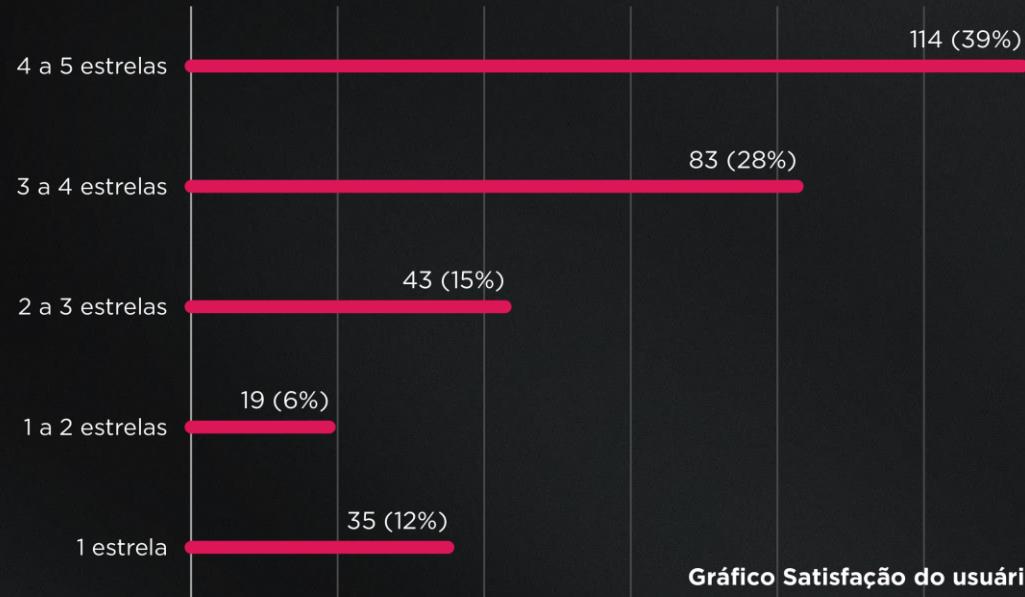
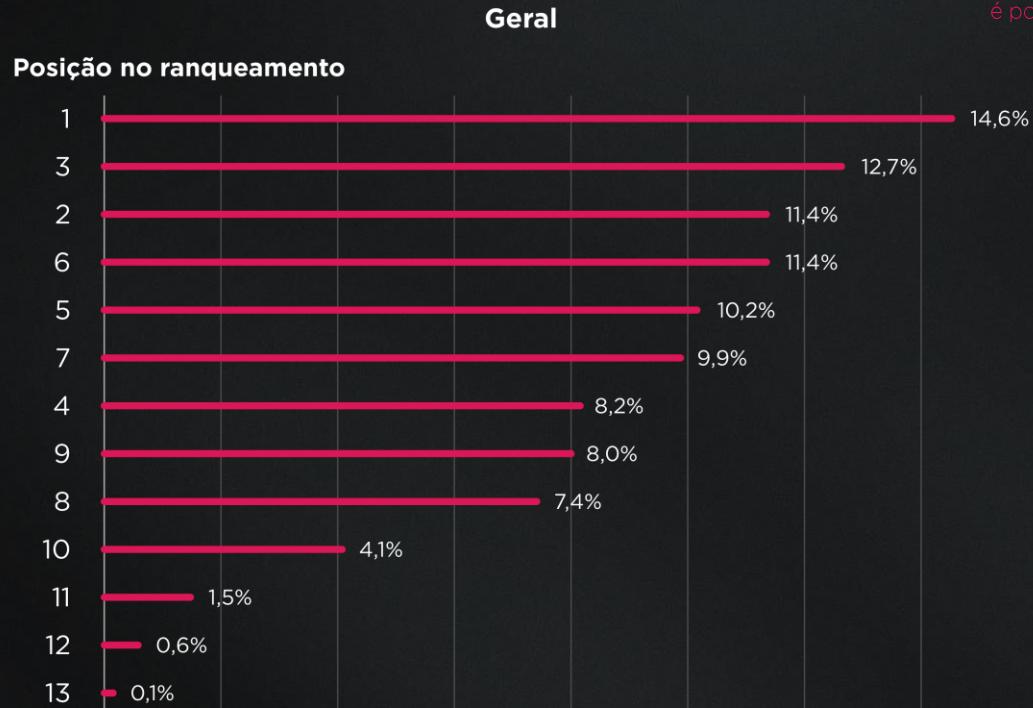


Gráfico Satisfação do usuário

AVALIAÇÃO DE UM **SISTEMA DE RECOMENDAÇÃO**

- Análise dos dados (Resultados) - verificação da eficácia e confirmação da terceira hipótese

Se (usar um ranqueamento por relevância) então
é possível encontrar itens mais relevantes



AVALIAÇÃO DE UM SISTEMA DE RECOMENDAÇÃO - DASHBOARD



PROJETO INTEGRADO SR

Suporte aos Grupos - Objetivo, Arquitetura e Protótipo

CAPÍTULO 10

Material de Apoio (Hands-on): Implementando um chat bot
utilizando inteligência artificial generativa (Gen. A.I.)

HANDS-ON – IMPLEMENTANDO UM CHAT BOT COM GEN AI

Breve contexto

- A Inteligência Artificial Generativa (Gen. AI) é uma sub-área da IA caracterizada pelo uso de grandes modelos de linguagens (LLM), estruturas baseadas em redes neurais capazes de acumular conhecimento de bases de dados e converter tais dados em respostas muito próximas às que seriam elaboradas por seres humanos.
- Uma capacidade importante deste modelo é a de adquirir contexto de uma conversa e moldar respostas que teriam que ser agrupadas e interpretadas pelo ser humano no modo tradicional dos sistemas de busca utilizados antes da Gen AI.
- Exemplos de Gen AI que proporcionam respostas com base em perguntas do tipo prompt: ChatGPT, Gemini, Copilot e DeepSeek, entre outros.

Exemplo



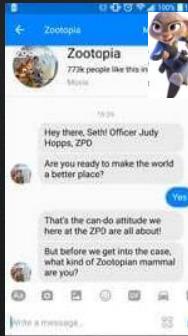
HANDS-ON – IMPLEMENTANDO UM CHAT BOT COM GEN AI

Gen AI em SR

- Uma das aplicações da Gen AI em SRs é a possibilidade de criar bases de conhecimentos específicas de um determinado negócio e fornecer recomendações personalizadas à funcionários ou clientes. Ex:
 - Sistemas de bases de conhecimento para apoio a: Call center, service desk, service support
 - Chat bot para apoio aos clientes seja na resolução de algum problema, esclarecimento de dúvidas ou suporte à venda de um produto ou serviço.
 - Suporte ao desenvolvedor: code quality, code completion, error support, etc.

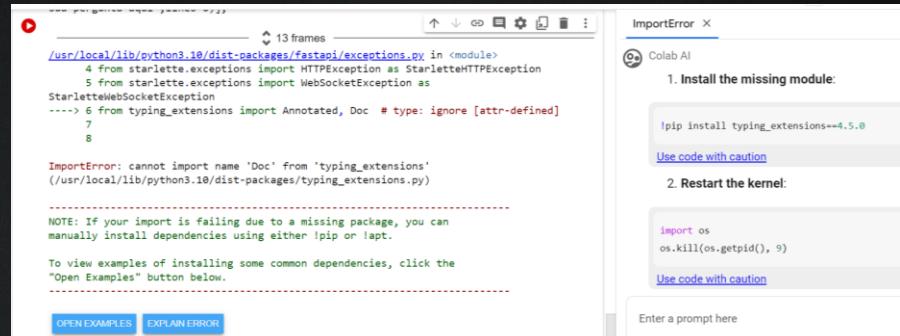
Cases:

- Disney, Uber, Vivo, Casas Bahia



- Link

- Google Colab (Colab AI)



HANDS-ON – IMPLEMENTANDO UM CHAT BOT COM GEN AI

Bases e roteiros para criação do chat bot

- Neste hands-on utilizaremos uma base de conhecimento sobre a disciplina “Física” composta por 3 bilhões de características:
 - [databricks/dolly-v2-3b · Hugging Face](#)
- Essa base possui modelos mais precisos, com mais dados e features, consequentemente, mais pesados para serem utilizados, ex 12b.
 - [databricks/dolly-v2-12b · Hugging Face](#)
- O tutorial que iremos executar foi baseado em um video da Analytics Vidhya: [Run 3 Open-Source LLMs on Google Colab - for FREE 🔥 Top Generative AI Model Hands-on \(Hugging Face\) - YouTube](#)
- Abra sua conta no google Colab para iniciar a implementação.

HANDS-ON – IMPLEMENTANDO UM CHAT BOT COM GEN AI

```
import locale
locale.getpreferredencoding = lambda : "UTF-8"
!pip install transformers
!pip install sentencepiece
!pip install accelerate

import torch
from transformers import pipeline
import os

dolly_pipeline = pipeline(model="databricks/dolly-v2-3b",
                           torch_dtype=torch.bfloat16,
                           trust_remote_code=True,
                           device_map="auto")

def get_completion_dolly(input):
    system = f"""
        Faça sua pergunta sobre física ao AI Clau Clau utilizando frases simples.
        Ele o ajudará na medida do possível, daquele jeitão dele.
    """
    prompt = f"### System: {system}\n### User: \n{input}\n\n### Resposta do AI Clau Clau:"
    print(prompt)
    dolly_response = dolly_pipeline(prompt, max_new_tokens = 500)
    return dolly_response[0]["generated_text"]
```

Change runtime type

Runtime type

Python 3

Hardware accelerator 

CPU

T4 GPU

A100 GPU

L4 GPU

TPU v2-8

Want access to premium GPUs? [Purchase additional compute units](#)

[Cancel](#) [Save](#)

HANDS-ON – IMPLEMENTANDO UM CHAT BOT COM GEN AI

```
prompt = "Explain to me the difference between nuclear fission and fusion."  
print(get_completion_dolly(prompt))
```

System:

Faça sua pergunta sobre física ao AI Clau Clau utilizando frases simples. Ele o ajudará na medida do possível, daquele jeitão dele.

User: Explain to me the difference between nuclear fission and fusion.

Resposta do AI Clau Clau:

Nuclear fission is the breaking up of a nucleus into two or more particles; whereas, nuclear fusion produces a nucleus plus additional particles.

HANDS-ON – IMPLEMENTANDO UM CHAT BOT COM GEN AI

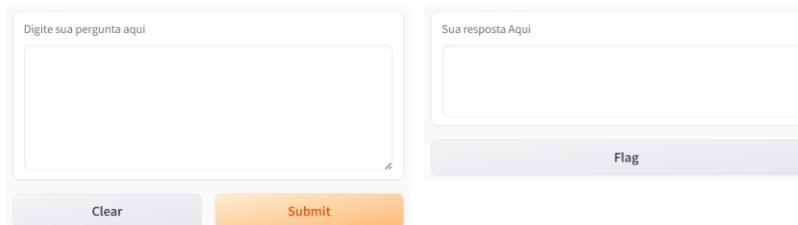
V2: COM INTERFACE GRÁFICA

```
!pip install --upgrade pip
!pip install --upgrade typing-extensions
import locale
locale.getpreferredencoding = lambda : "UTF-8"
!pip install gradio

import gradio as gr

iface = gr.Interface(fn=get_completion_dolly, inputs=[gr.Textbox(label="Digite sua pergunta aqui",lines=6)],
outputs=[gr.Textbox(label="Sua resposta Aqui",lines=3)],
title="AI Clau Clau. Faça uma pergunta sobre Física que o prof. Claudecir (Clau Clau) vai te responder daquele jeitão.",
examples=["Explain to me the difference between nuclear fission and fusion.",
"Why is the Sky blue?"])
iface.launch(share=True)
```

AI Clau Clau. Faça uma pergunta sobre Física que o prof. Claudecir (Clau Clau) vai te responder daquele jeitão.



CAPÍTULO 11

Tendências dos SRs e considerações finais

SISTEMAS DE RECOMENDAÇÃO – Gaps, Oportunidades e TENDÊNCIAS

- Sistemas de Recomendação podem ser muito “chatos” (irrelevantes ou inadequados) sugerindo produtos aleatoriamente ou com base em um histórico de navegação que você já não quer mais consumir.
- Por outro lado, são ferramentas muito úteis (relevantes e adequados) para usuários finais (cliente/consumidor), para usuários de negócio (que usam o SR como ferramenta de apoio para melhores tomadas de decisões) e ainda pelos proprietários dos SRs (com insights obtidos por análise de dados).
- Neste contexto, é importante estarmos atentos às oportunidades de aprimoramento dos SRs, visando proporcionar maior valor agregado ao negócio.

SISTEMAS DE RECOMENDAÇÃO – Gaps, Oportunidades e TENDÊNCIAS

Exemplo de recomendações irrelevantes ou inadequadas:

The screenshot shows a news article from g1 about Black Friday. The headline reads: "Comprou na Black Friday e se arrependeu? Veja o que fazer". Below the headline, there is a list of bullet points: "Filas e busca por TVs: veja FOTOS da Black Friday pelo mundo". The text "Há 7 horas — Em Black Friday" is also present. To the right of the article, there are several recommended images: "berm estar", "desenrola, rio", "prazer, renata", and "bichos na escuta". Below the news article, there is an advertisement for a Samsung Smart TV. The TV is labeled "SAMSUNG 2022 Crystal UHD". The text "Smart TV 55" Crystal UHD 4K Samsung 55BU8000 Painel Dynamic Crystal Color Design Slim T..." is displayed, along with a link "Saiba mais". The logo "Americana.com" is visible at the bottom of the ad. To the right of the TV ad, there is another advertisement for the same TV model on the website "americanas.com". The TV is labeled "Smart TV 55" Crystal UHD 4K...". The price "R\$ 3.449,99" is shown, along with a discount offer "-10% R\$ 3.100,00". Below this, another TV model is listed with the price "R\$ 2.496,75".

IMAGEM 83 – RECOMENDAÇÃO IRRELEVANTE E-COMMERCE

SISTEMAS DE RECOMENDAÇÃO – Gaps, Oportunidades e TENDÊNCIAS

Exemplo de recomendações irrelevantes ou inadequadas:



IMAGEM 84 - RECOMENDAÇÃO IRRELEVANTE E-COMMERCE 2

SISTEMAS DE RECOMENDAÇÃO – Gaps, Oportunidades e TENDÊNCIAS

Exemplo de recomendações irrelevantes ou inadequadas:

The image consists of two side-by-side screenshots from the Mercado Livre e-commerce platform.

Left Screenshot: Histórico de navegação (Browsing History)

- Tênis Nike Casual - R\$ 65
6x R\$ 10,83 sem juros
- Bota - R\$ 59
6x R\$ 9,83 sem juros

Right Screenshot: Recomendações baseadas no último visto (Based on last viewed)

- Cela Mangalarga Completa + Acessórios Pronta Para Uso - R\$ 497
10x R\$ 49,70 sem juros
Frete grátis
- Cela Completa Mangalarga Arreada + Manta Porta Sela ... - R\$ 497
10x R\$ 49,70 sem juros
Frete grátis
- Cabresto Quarto De Milha Para Cavalo Reforçado Nylon... - R\$ 30
- Sela Mangalarga Gel De Cabeça Completa E Aces... - R\$ 559
10x R\$ 55,90 sem juros
Frete grátis

Both screenshots show a significant disconnect between the user's previous purchases (shoes) and the recommended products (equestrian equipment like saddle pads and girths), which is considered an irrelevant recommendation.

IMAGEM 85 – RECOMENDAÇÃO IRRELEVANTE E-COMMERCE 3

SISTEMAS DE RECOMENDAÇÃO – Gaps, Oportunidades e TENDÊNCIAS

Caso George Floyd: tropas ocupam ruas dos EUA após protestos violentos contra o racismo



pressionou o pescoço da vítima com o joelho durante uma abordagem na última segunda-feira.



RELACIONADAS



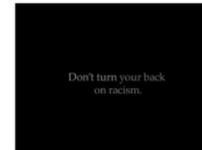
De Obama a Kim Kardashian, personalidades protestam por George Floyd



A longa história de segregação e conflito racial em Minneapolis, onde George Floyd foi morto pela polícia



Don't turn your back on racism.



Nike lança manifesto contra o racismo e ganha apoio da Adidas



Swift



SISTEMAS DE RECOMENDAÇÃO – Gaps, Oportunidades e TENDÊNCIAS

Caso George Floyd: tropas ocupam ruas dos EUA após protestos violentos contra o racismo



última segunda-feira.



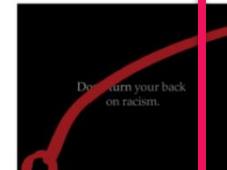
RELACIONADAS



De Obama a Kim Kardashian, personalidades protestam por George Floyd



A longa história de segregação e conflito racial em Minneapolis, onde George Floyd foi morto pela polícia



Nike lança manifesto contra o racismo e conta com apoio da Adidas



Publicidade relacionada com similaridade => maior valor agregado ao negócio

IMAGEM 84 – RECOMENDAÇÃO RELEVANTE

SISTEMAS DE RECOMENDAÇÃO – Gaps, Oportunidades e TENDÊNCIAS

- Estar atento ao lançamento de features geradas por algum tipo de recomendação e nunca parar de recomendar, como Spotify

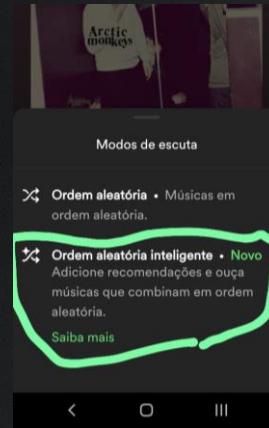
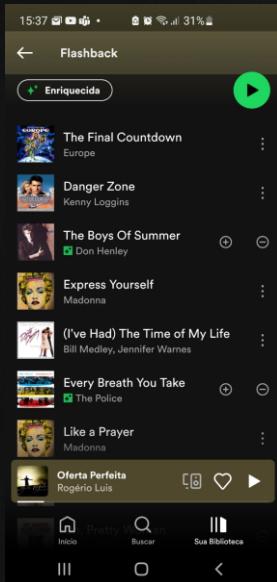


IMAGEM 85 – FEATURES EM SR

SISTEMAS DE RECOMENDAÇÃO - **TENDÊNCIAS**

- Tornar a recomendação algo natural na experiência do usuário, mesmo não sendo o principal fator de decisão do cliente ao escolher uma plataforma em detrimento de outra. Isto faz com que ele fidelize, se sinta satisfeito e passe cada vez mais tempo na sua plataforma.

Fatores explícitos que levam os usuários a preferir uma plataforma:

- Preço;
- Acervo;
- Performance.

Fatores implícitos que levam os usuários a fidelizar em uma plataforma:

- Usabilidade;
- Recomendação.



IMAGEM 86 - FATORES PREFERENCIA STREAMING

SISTEMAS DE RECOMENDAÇÃO - **TENDÊNCIAS**

- Os SRs tendem a ser cada vez mais:



IMAGEM 87 – TENDÊNCIAS - HER

SISTEMAS DE RECOMENDAÇÃO - TENDÊNCIAS

- Apesar de alguns problemas que este tipo de recomendação pode causar, por exemplo:

Privacidade

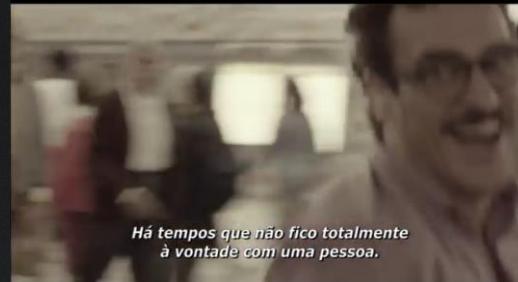
Bolhas sociais

Viés ou
manipulação

Isolamento
social

Resumindo, os SRs devem estar cada vez mais:

Integrados
naturalmente
à vida das
pessoas



SISTEMAS DE RECOMENDAÇÃO - **CONSIDERAÇÕES FINAIS**

- Iniciar a construção de um SR é **relativamente simples**. Com apenas uma técnica ou controle já se implementa alguma forma de recomendação, como click, timer, contadores, ou ainda filtros baseados em algum conhecimento prévio dos dados.
- Por outro lado, vimos também que técnicas **isoladas** podem trazer **problemas** ao SR, como Cold-start, aleatoriedade e até previsibilidade;
- Para atacar esses problemas e construir SRs mais robustos e **sofisticados**, estuda-se **propriedades, técnicas e algoritmos**. Em posse dessas informações, define-se a melhor **estratégia** para aplicar as ferramentas mais adequadas para cada necessidade e dessa forma, utiliza-se então, uma **abordagem híbrida**.

SISTEMAS DE RECOMENDAÇÃO - CONSIDERAÇÕES FINAIS

- Importante também conhecermos **aplicações de mercado** e como as principais plataformas da atualidade utilizam esse poderoso recurso, que é a recomendação para alavancar seus negócios. Adicionalmente, ter acesso a estudos de casos que implementam SRs é sempre uma boa ideia, pois nos mostra alguns dos possíveis caminhos a seguir, utilizando-se de boas práticas do mercado e da **acadêmica**.
- Por fim, os SRs possuem características **particulares** em relação ao desenvolvimento de um sistema **tradicional**, principalmente na fase de **Validação (Avaliação)**. Por este motivo, definir as métricas, organizar um experimento, analisar os resultados e confirmar/refutar as hipóteses são essenciais para uma boa avaliação do SR. Os resultados podem nos indicar também, necessidade de **re-calibragem** do sistema, **lacunas** encontradas e **oportunidades** de trabalhos futuros que devem ser considerados como processo de **melhoria contínua do SR**.

PROJETO INTEGRADO SR

Suporte aos Grupos - Objetivo, Arquitetura e Protótipo

OBRIGADO

Preencher Survey,
navegar pelas refs,
ver anexo e manter
contato

FIAP

PROFESSOR PAGNOSSIM



profjose.pagnossim@fiap.com.br



linkedin.com/in/jose-luiz-maturana-pagnossim-3454898



(11) 980587052

Referências

RICCI, F. et al. Recommender Systems Handbook. [S.I.]: Springer, 2011

GUNAWARDANA, S. Evaluating recommender systems. In: RICCI, F. e. a. (Ed.). Recommender Systems Handbook Second Edition. [S.I.]: Springer, 2015.

A. Diaz, A. Lima, A. Silva, F. Costa, J. Pagnossim, S. Peres. Uma análise comparativa das ferramentas de pré-processamento de dados textuais: NLTK, PreTeXT e R. Relatórios Técnicos, EACH-USP. São Paulo-SP, 2018.

PAGNOSSIM, José Luiz Maturana. Uma abordagem híbrida para sistemas de recomendação de notícias. 2018. 119 f. Dissertação (Mestrado em Ciências) – Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, São Paulo, 2018.

LAURIA, Rayana de Souza Aplicação de sistemas de recomendação em finanças – FGV, Rio de Janeiro, RJ, 2020.

Netflix TechBlog. Netflix Recommendations: Beyond the 5 stars (Part 1). 2012. Disponível em: <<https://netflixtechblog.com/netflix-recommendations-beyond-the-5-stars-part-1-55838468f429>>. Acesso em: 26 nov. 2022.

Escola de E-commerce. O motivo por trás do sucesso do Sistema de Recomendação da Amazon. 2020. Disponível em: <<https://www.escoladeecommerce.com/artigos/sucesso-do-sistema-de-recomendacao-da-amazon/>>. Acesso em: 26 nov. 2022.

EBC. Portal EBC de notícias. 2022. Disponível em: <<https://www.ebc.com.br/>>. Acesso em: 26 nov. 2022.

iMasters. Principais técnicas de machine learning. 2022. Disponível em: <<https://imasters.com.br/back-end/principais-tecnicas-de-machine-learning>>. Acesso em: 26 nov. 2022.

Diogo Nogare. Explicando o algoritmo de regra de associação. 2020. Disponível em: <<https://diegonogare.net/2020/05/explicando-o-algoritmo-de-regra-de-associacao/>>. Acesso em: 26 nov. 2022.

Santander. Open Finance. 2022. Disponível em: <<https://www.santander.com.br/hotsite/open-finance/>>. Acesso em: 26 nov. 2022.

Python. Downloads. 2022. Disponível em: <<https://www.python.org/downloads/>>. Acesso em: 26 nov. 2022.

Referências

Hashtag Treinamentos. Como Criar um Gráfico de Dispersão no Excel. 2020. Disponível em: <<https://www.youtube.com/watch?v=IJLKkr4M9ko>>. Acesso em: 26 nov. 2022.

Warner Bros. Her (Filme). 2020. Disponível em: <<https://www.warnerbros.com/movies/her>>. Acesso em: 26 nov. 2022.

Youtube. Her - Trailler oficial. 2014. Disponível em: <<https://www.youtube.com/watch?v=hX09Kz7BAIU>>. Acesso em: 26 nov. 2022.

Canal Tech – LLM <https://canaltech.com.br/inteligencia-artificial/o-que-e-llm-large-language-model/>. Acesso em 29 jan 2024

Hamper – Exemplos de Chatbot

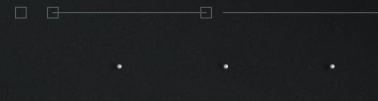
<<https://www.google.com.br/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjdqaeF04QEAxVhqZUCHckMDuYQFnoECBQQAw&url=https%3A%2F%2Framper.com.br%2Fblog%2Fexemplos-de-chatbot%2F&usg=AOvVaw0mdkmvtHo5HcGGvoGj7EL&opi=89978449>>

Acesso em 29 jan 2024

DataBricks – Dataset Dolly 3b <[databricks/dolly-v2-3b](https://databricks.com/dolly-v2-3b) · Hugging Face>. Acesso em 29 jan 2024

DataBricks – Dataset Dolly 3b <[databricks/dolly-v2-12b](https://databricks.com/dolly-v2-12b) · Hugging Face >. Acesso em 29 jan 2024

Youtube – Tutorial chatbot Gen AI Dolly 3b https://youtu.be/3fyazg9-n2o?si=a6-MaHxu6M_tvUNa. Acesso em 29 jan 2024



ANEXO I

7.1. Material de Apoio: Modelando e implementando uma prova de conceito de SR para Open Finance

DESENVOLVENDO UM SISTEMA DE RECOMENDAÇÃO

- Estudo de caso: Modelando e implementando uma prova de conceito de SR para Open Finance
 - Modelando uma base de dados em SQL-Server - Modelo a ser implementado no exemplo:
 - tbCliente: Tabela para cadastro dos clientes;
 - tbProduto: Tabela para cadastro dos produtos;
 - tbClienteProduto: Relacionamento entre Cliente e Produto, com o valor que o cliente tem aplicado ou financiado no produto
 - tbClienteSimilaridade: Auto-relacionamento com tbCliente para associar a similaridade entre clientes.

```
clientes = {  
    'Ana':  
        {'Cartão de Crédito': 1,  
        'Conta Corrente': 2,  
        'Poupança': 3,  
        'Renda Fixa': 4,  
        'Crédito Pessoal': 5},  
  
    'Marcos':  
        {'Cartão de Crédito': 2,  
        'Conta Corrente': 3,  
        'Poupança': 4,  
        'Renda Fixa': 5,  
        'Renda Variável': 0.6},  
  
    'Pedro':  
        {'Cartão de Crédito': 3,  
        'Conta Corrente': 4,  
        'Poupança': 5,  
        'Crédito Pessoal': 7},  
  
    'Cláudia':  
        {'Cartão de Crédito': 4,  
        'Conta Corrente': 5,  
        'Poupança': 6}  
}
```

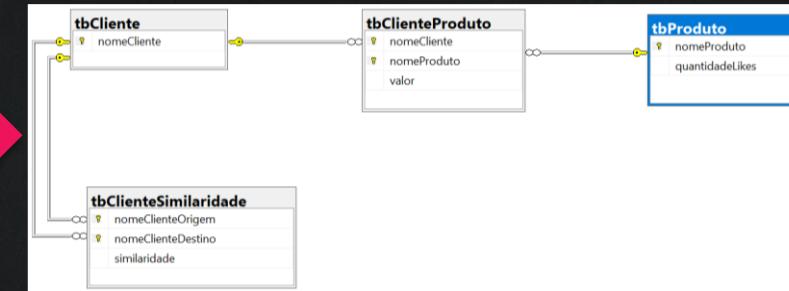


IMAGEM 52 – MODELO DE DADOS RELACIONAL

DESENVOLVENDO UM **SISTEMA DE RECOMENDAÇÃO**

- Estudo de caso: Modelando e implementando uma prova de conceito de SR para Open Finance

nomeCliente
Ana
Claudia
Marcos
Pedro
* NULL

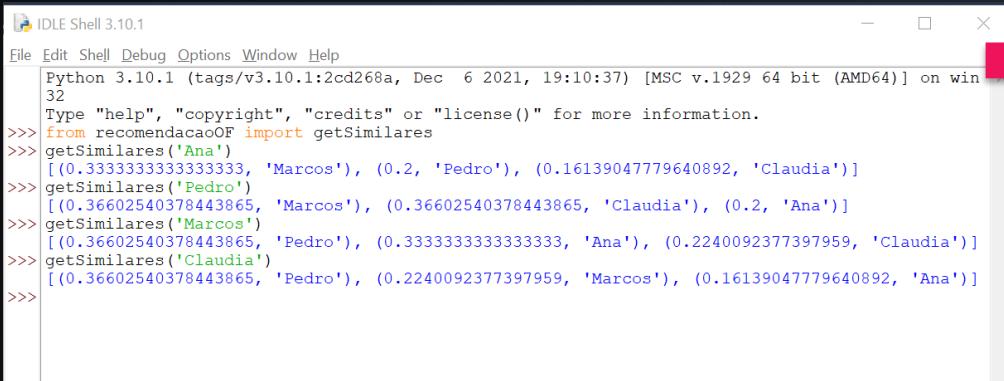
nomeCliente	nomeProduto	valor
Ana	Cartão de Crédito	1
Ana	Conta Corrente	2
Ana	Crédito Pessoal	3
Ana	Poupança	4
Ana	Renda Fixa	5
Marcos	Cartão de Crédito	2
Marcos	Conta Corrente	3
Marcos	Poupança	4
Marcos	Renda Fixa	5
Marcos	Renda Variável	0,6
Pedro	Cartão de Crédito	3
Pedro	Conta Corrente	4
Pedro	Poupança	5
Pedro	Crédito Pessoal	7
Claudia	Cartão de Crédito	4
Claudia	Conta Corrente	5
Claudia	Poupança	6
* NULL	NULL	NULL

nomeProduto	quantidadeLikes
Cartão de Crédito	1
Conta Corrente	3
Crédito Pessoal	2
Poupança	0
Renda Fixa	0
Renda Variável	0
* NULL	NULL

IMAGEM 53 – CARGA DOS DADOS NAS TABELAS

DESENVOLVENDO UM SISTEMA DE RECOMENDAÇÃO

- Carregando os resultados do algoritmo de similaridade do Python para a tabela na BD



```
IDLE Shell 3.10.1
File Edit Shell Debug Options Window Help
Python 3.10.1 (tags/v3.10.1:2cd268a, Dec  6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)] on win32
32
Type "help", "copyright", "credits" or "license()" for more information.
>>> from recomendacao import getSimilarares
>>> getSimilarares('Ana')
[(0.3333333333333333, 'Marcos'), (0.2, 'Pedro'), (0.16139047779640892, 'Claudia')]
>>> getSimilarares('Pedro')
[(0.36602540378443865, 'Marcos'), (0.36602540378443865, 'Claudia'), (0.2, 'Ana')]
>>> getSimilarares('Marcos')
[(0.36602540378443865, 'Pedro'), (0.3333333333333333, 'Ana'), (0.2240092377397959, 'Claudia')]
>>> getSimilarares('Claudia')
[(0.36602540378443865, 'Pedro'), (0.2240092377397959, 'Marcos'), (0.16139047779640892, 'Ana')]
>>>
```



	nomeClienteOrigem	nomeClienteDestino	similaridade
1	Ana	Claudia	0,16
2	Ana	Marcos	0,33
3	Ana	Pedro	0,2
4	Claudia	Ana	0,16
5	Claudia	Marcos	0,22
6	Claudia	Pedro	0,36
7	Marcos	Ana	0,33
8	Marcos	Claudia	0,22
9	Marcos	Pedro	0,36
10	Pedro	Ana	0,2
11	Pedro	Claudia	0,36
12	Pedro	Marcos	0,36

IMAGEM 54 – CARGA TABELA SIMILARIDADE

tbClienteSimilaridade

DESENVOLVENDO UM **SISTEMA DE RECOMENDAÇÃO**

- Criando uma aplicação Web para implementar e demonstrar as funcionalidades do SR

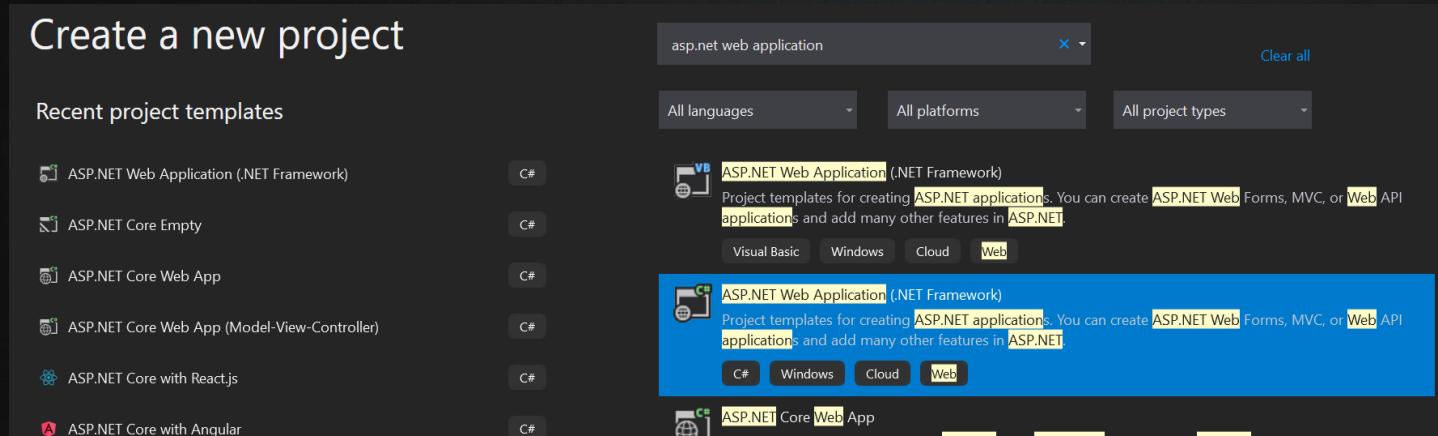


IMAGEM 55 – CRIANDO PROJETO WEB

DESENVOLVENDO UM **SISTEMA DE RECOMENDAÇÃO**

- Criando uma aplicação Web para implementar e demonstrar as funcionalidades do SR
Defina um nome para sua aplicação e o local em que o projeto será salvo.

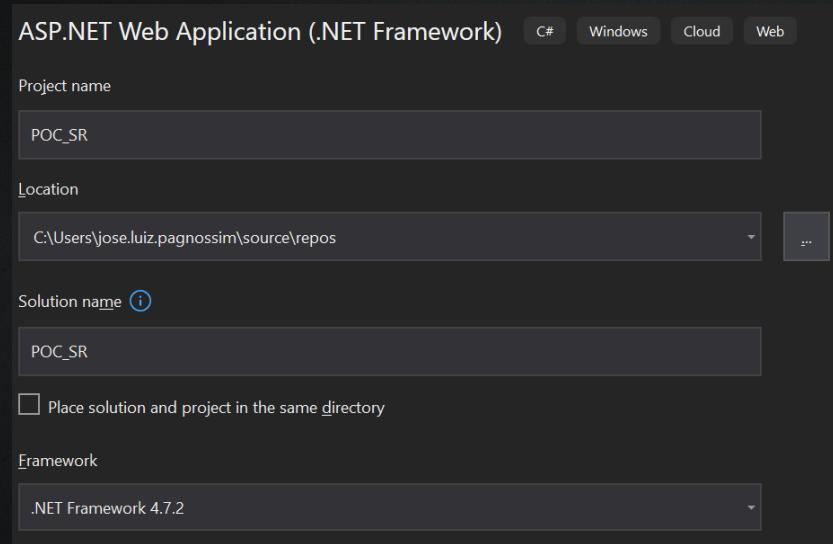


IMAGEM 56 -PROJETO WEB NOME

DESENVOLVENDO UM **SISTEMA DE RECOMENDAÇÃO**

- Criando uma aplicação Web para implementar e demonstrar as funcionalidades do SR
Escolha a opção Empty e desmarque a opção Advanced/Configure HTTPS.

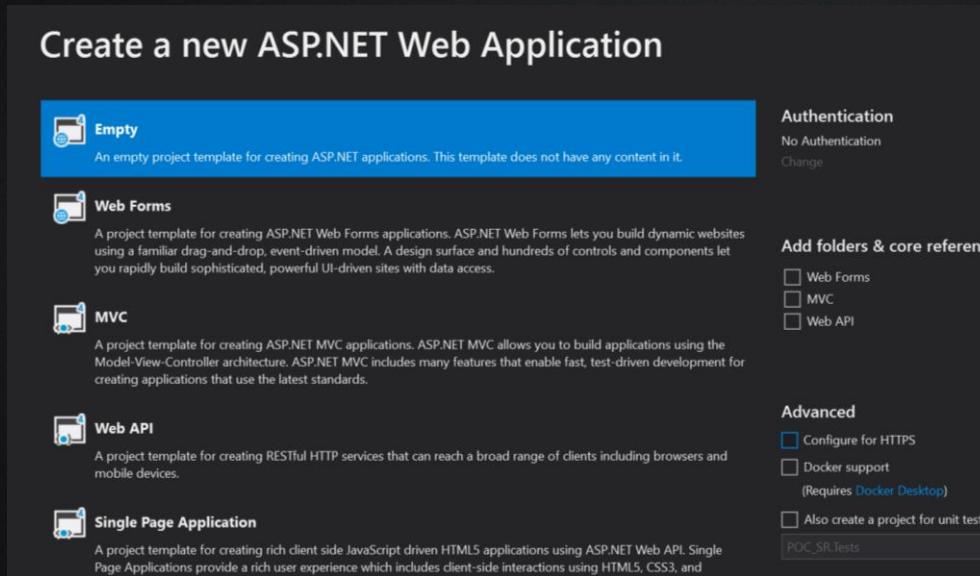


IMAGEM 57 - NOVO PROJETO VAZIO

DESENVOLVENDO UM **SISTEMA DE RECOMENDAÇÃO**

- Criando uma aplicação Web para implementar e demonstrar as funcionalidades do SR
Crie um novo webform para iniciar o desenvolvimento.

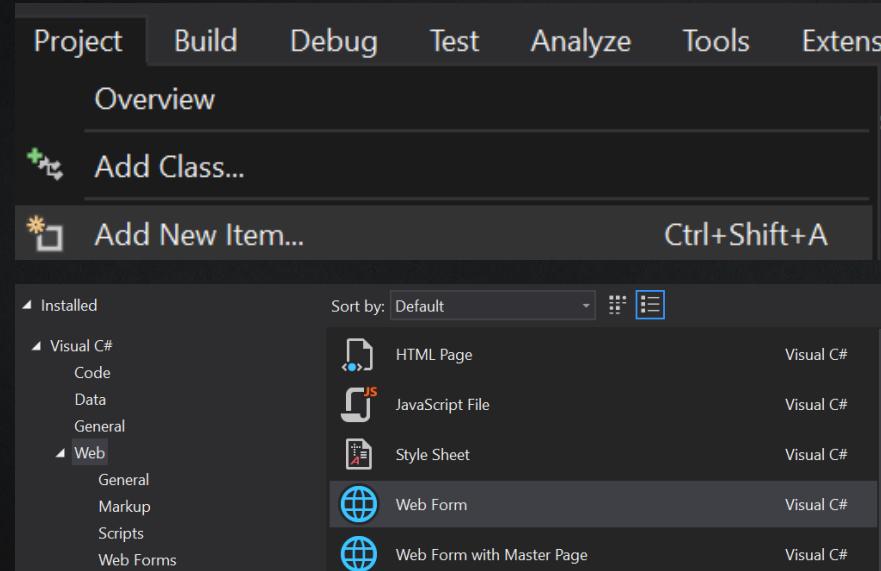


IMAGEM 58 – NOVO WEB FORM

DESENVOLVENDO UM **SISTEMA DE RECOMENDAÇÃO**

- Criando uma aplicação Web para implementar e demonstrar as funcionalidades do SR
Defina o WebForm1.aspx como página de início. Na sequência, execute o projeto com IIS Express.

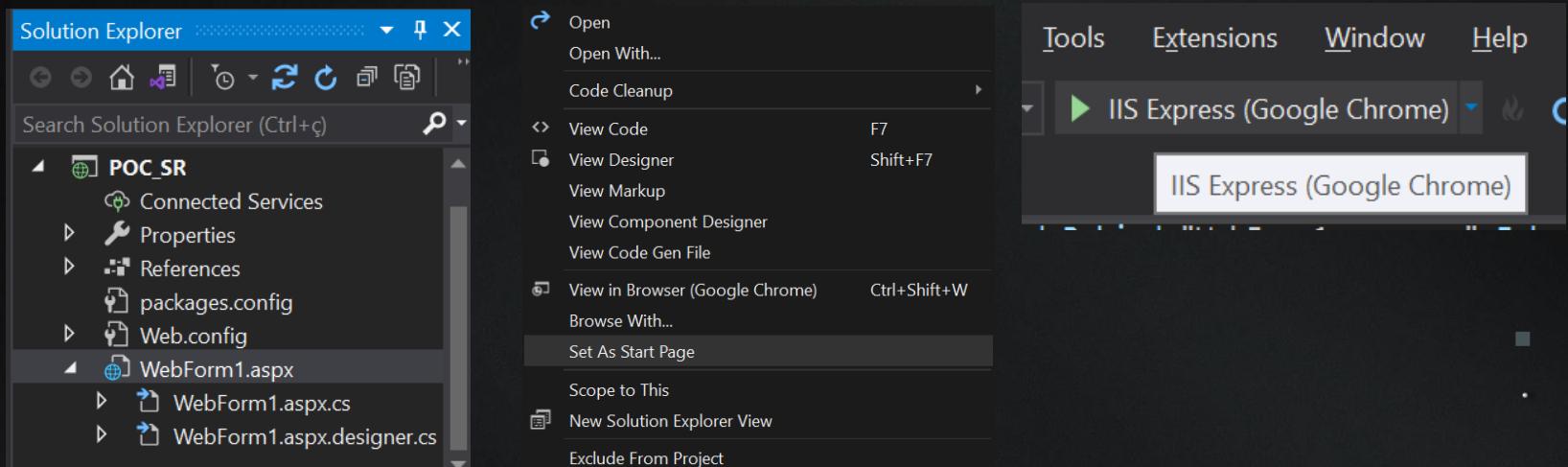


IMAGEM 59 – SET START PAGE

DESENVOLVENDO UM **SISTEMA DE RECOMENDAÇÃO**

- Criando uma aplicação Web para implementar e demonstrar as funcionalidades do SR
Seu projeto será executado no navegador indicado na execução. Neste caso, no Google Chrome.

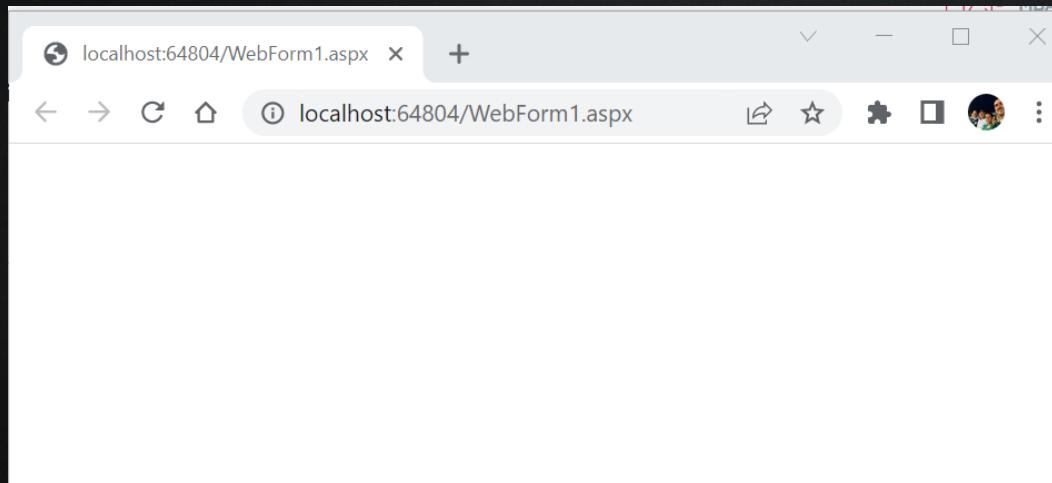


IMAGEM 60 - EXECUÇÃO PÁGINA WEB EM BRANCO

DESENVOLVENDO UM SISTEMA DE RECOMENDAÇÃO

- Criando uma aplicação Web para implementar e demonstrar as funcionalidades do SR
A seguir, o layout do frontend e as funcionalidades a serem implementadas.

The screenshot shows a web application interface for 'Din Din Fintech'. At the top, there's a header with the company name and a dropdown menu. Below the header, a message says 'Olá [Ana #2 v], Ag. 1234 CC. 00012345-6'. There are two sections: 'Recomendação Open Finance' and 'Recomendação Popularidade'. The first section displays a GridView of client products with columns: nomeCliente, nomeProduto, and valor. The second section displays a GridView of products with columns: nomeProduto and quantidadeLikes. At the bottom, there's a section for liking products, with icons for Cartão de Crédito, Conta Corrente, Crédito Pessoal, Poupança, Renda Fixa, and Renda Variável, each with a 'Like' button and a count of likes. A red box highlights the 'Actualizar Cliente' button and the 'Recomendação Popularidade' section, with a detailed description of their functions.

Din Din Fintech #1 #3

Olá [Ana #2 v], Ag. 1234 CC. 00012345-6

Atualizar Cliente #4

Recomendação Open Finance

nomeCliente	nomeProduto	valor
Marcos	Renda Variável	0,6

Recomendação Popularidade

nomeProduto	quantidadeLikes
Conta Corrente	3
Crédito Pessoal	2
Cartão de Crédito	2
Poupança	0
Renda Fixa	0
Renda Variável	0

Está curtindo os produtos da Din Din? #7

Cartão de Crédito

Conta Corrente

Crédito Pessoal

Poupança

Renda Fixa

Renda Variável

[Resultado] #8

#1 - Cabecalho da aplicação (fixo)
#2 - DropDownList com a listagem de todos os clientes
#3 - Agência e Conta (fixos)
#4 - Botão para atualizar os dados de recomendação, após seleção de um outro cliente no DropDownList
#5 - GridView com o resultado da recomendação por filtro colaborativo (Encontrar o cliente mais similar em relação ao cliente selecionado no DropDownList e selecionar os produtos que este cliente de destino possui e que o cliente de origem não possua)
#6 - GridView com o resultado da listagem dos produtos da tabela de produtos em ordem decrescente de quantidade de Likes
#7 - Espaço de colaboração que o usuário pode curtir ou descurtir um produto. Ao curtir, incrementar o campo quantidadeLikes da tabela de produto, ao descurtir, este campo deverá ser decrementado.
#8 - Label de resultado para apresentar mensagens de exceção geradas pela aplicação

IMAGEM 61 - LAYOUT FRONTEND

DESENVOLVENDO UM **SISTEMA DE RECOMENDAÇÃO**

- Criando uma aplicação Web para implementar e demonstrar as funcionalidades do SR

No evento Load (carga do web form), são executados 4 blocos: O primeiro para preencher o dropdown com os nomes dos clientes. O segundo busca o usuário mais similar com base no cliente selecionado no dropdown. O terceiro popula o grid de recomendação com base na técnica de filtro colaborativo. Por fim, o último grid carrega os produtos em ordem de maior popularidade. Esta é uma camada que integra o frontend ao backend, que por enquanto está abstraído.

```
using System.Data;
Dados objDados = new Dados();
protected void Page_Load(object sender, EventArgs e) {
if (Page.IsPostBack == false) {
    //dropdown cliente
    DataTable dt = objDados.listarClientes();
    DropDownList1.DataSource = dt;
    DropDownList1.DataTextField = "nomeCliente";
    DropDownList1.DataBind();
    //usuário mais similar
    string strClienteMaisSimilar = objDados.consultarClienteMaisSimilar(DropDownList1.Text);
    //gridview recomendacao
    dt = objDados.listarRecomendacaoOF(DropDownList1.Text, strClienteMaisSimilar);
    GridView1.DataSource = dt;
    GridView1.DataBind();
    //gridview popularidade
    dt = objDados.listarProdutosPopularidade();
    GridView2.DataSource = dt;
    GridView2.DataBind(); } }
```

DESENVOLVENDO UM **SISTEMA DE RECOMENDAÇÃO**

- Criando uma aplicação Web para implementar e demonstrar as funcionalidades do SR

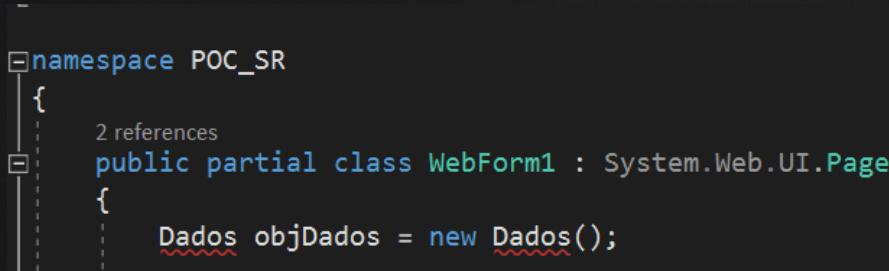
Os botões de Curtiu/Descurtiu trabalham com uma lógica relacionada à imagem. Dessa forma, a imagem padrão indica não curtiu, ao clicar no botão o evento verifica se a imagem é de Descurtiu, se for, muda para a imagem de Curtiu e contabiliza o indicador (incrementa quantidadeLikes). O inverso funciona de forma análoga, decrementando o indicador de Likes e voltando a imagem para Descurtiu. O exemplo abaixo funciona para o botão de Cartão de Crédito. Para os demais botões, basta usar o mesmo código alterando apenas o nome do componente de imagem e o produto que deseja incrementar ou decrementar.

```
protected void ImageButton1_Click(object sender, ImageClickEventArgs e)
{
    if (ImageButton1.ImageUrl == "~/10_likeNao.png")
    {
        ImageButton1.ImageUrl = "~/10_likeSim.png";
        objDados.IncrementarLike("Cartão de Crédito");
    }
    else
    {
        ImageButton1.ImageUrl = "~/10_likeNao.png";
        objDados.DecrementarLike("Cartão de Crédito");
    }
}
```

DESENVOLVENDO UM **SISTEMA DE RECOMENDAÇÃO**

- Criando uma aplicação Web para implementar e demonstrar as funcionalidades do SR

Com a parte do Frontend implementada e as chamadas aos métodos do objeto objDados que estávamos abstraindo até o momento, agora podemos desenvolver a camada de Backend, responsável pelas consultas e atualização dos dados na Base de Dados. Para resolver o erro abaixo, precisamos criar a classe Dados.



```
namespace POC_SR
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        Dados objDados = new Dados();
```

IMAGEM 62 - CLASSE DADOS

DESENVOLVENDO UM **SISTEMA DE RECOMENDAÇÃO**

Os métodos de acesso a dados devem ser implementados em uma classe que vamos chamar de 'Dados'. Para criar uma nova classe com esse nome, faça o seguinte:

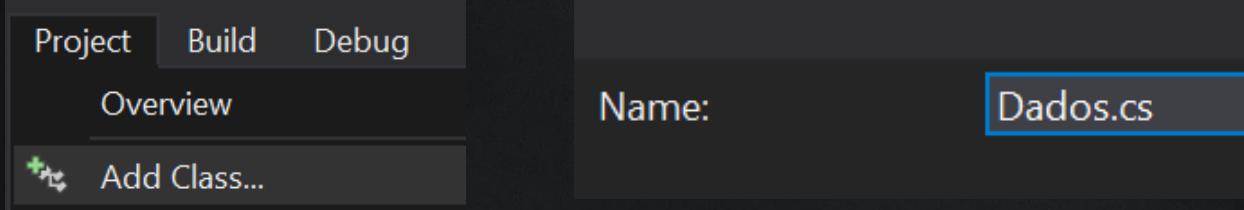
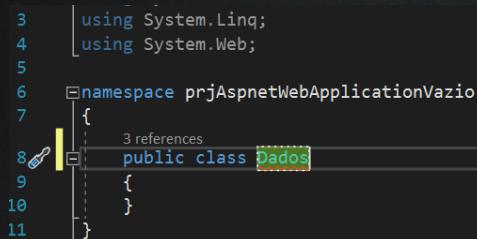


IMAGEM 62 - CLASSE DADOS

DESENVOLVENDO UM **SISTEMA DE RECOMENDAÇÃO**

Criando uma aplicação Web para implementar e demonstrar as funcionalidades do SR

Note que foi criada uma classe vazia.



```
3  using System.Linq;
4  using System.Web;
5
6  namespace prjAspnetWebApplicationVazio
7  {
8      public class Dados
9      {
10  }
11 }
```

IMAGEM 63 – CLASSE DADOS VAZIA

Agora, vamos declarar as bibliotecas para acesso aos métodos de Dados e do Client do SQL-Server.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data;
using System.Data.SqlClient;
namespace prjAspnetWebApplicationVazio{
    public class Dados
    {
    }
}
```

DESENVOLVENDO UM **SISTEMA DE RECOMENDAÇÃO**

Criando uma aplicação Web para implementar e demonstrar as funcionalidades do SR

Agora, vamos implementar o método listarClientes, que consulta todos os clientes da tabela tbClientes e retorna um objeto do tipo DataTable (dt) para ser populado no dropdown.

```
SqlConnection con; //objeto responsável pela conexão com a BD
SqlCommand cmd; //objeto que implementa o comando SQL e associa à conexão
public DataTable listarClientes(){
    conectar(); //Conecta na BD do SQL-Server
    con.Open(); //Abre a conexão com a BD
    string sql = "SELECT nomeCliente FROM tbCliente"; //Busca os clientes da tabela tbCliente
    SqlCommand adoCmd = new SqlCommand(sql, con); //Associa o comando SQL com a conexão
    SqlDataAdapter da = new SqlDataAdapter(adoCmd); //Cria um adaptador associado ao comando SQL, esse adaptador é necessário para
    //preencher os dados retornados no comando SQL para um objeto DataTable
    DataTable dt = new DataTable(); //Cria um objeto DataTable para receber o resultado do comando SQL
    da.Fill(dt); //Preenche o DataTable utilizando o método da.Fill do objeto SqlDataAdapter
    return dt; //Retorna o objeto dt (DataTable preenchido)
```

The screenshot shows a SQL Server Management Studio (SSMS) interface. At the top, it displays the connection path: CPX-BF030SWZ5XN... -> dbo.tbProduto -> CPX-BF030SWZ5XN..NDIN - Diagram_0*. Below this, a query window contains the following code:

```
CPX-BF030SWZ5XN... -> dbo.tbProduto -> CPX-BF030SWZ5XN..NDIN - Diagram_0*
/*
***** Seleciona todos os clientes da tabela tbCliente *****/
SELECT nomeCliente FROM tbCliente
```

Under the results tab, the output is shown in a table:

nomeCliente
1 Ana
2 Claudia
3 Marcos
4 Pedro

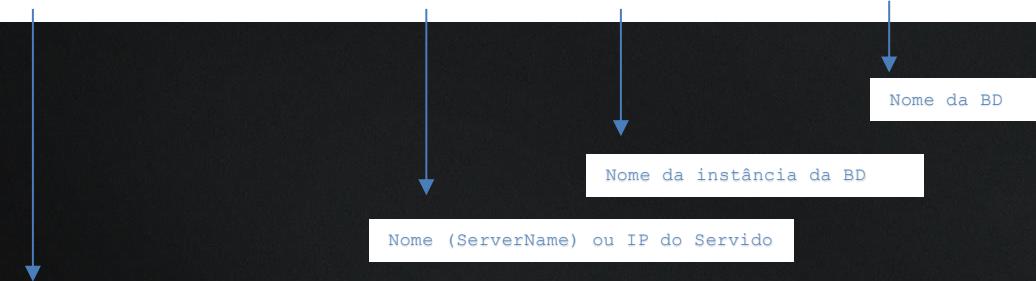
Este é o resultado que esperamos obter no retorno do banco de dados.

DESENVOLVENDO UM **SISTEMA DE RECOMENDAÇÃO**

Criando uma aplicação Web para implementar e demonstrar as funcionalidades do SR

No método anterior, utilizamos um procedimento chamado conectar(), responsável por definir os dados de conexão com a BD (string de conexão).

```
private void conectar()
{
    con = new SqlConnection("Data Source=CPX-8F030SWZ5XN\SQLEXPRESS01;Initial Catalog = DBDINDIN; Integrated Security = True");
}
```



Autenticação do Windows.
Há outras formas de se definir a string de conexão, caso seja necessário especificar usuário e senha do SQL-Server. Exemplo:

```
"Persist Security Info=False;User ID=*****;Password=*****;Initial Catalog=AdventureWorks;Server=MySqlServer"
```

DESENVOLVENDO UM SISTEMA DE RECOMENDAÇÃO

- Criando uma aplicação Web para implementar e demonstrar as funcionalidades do SR

Vamos ver agora como implementar a regra de Filtro Colaborativo, aproveitando o cálculo da Similaridade realizada em Python e carregada na nossa BD da aplicação. O primeiro passo é identificar qual é o cliente mais similar em relação ao usuário logado ou selecionado no nosso dropdown.

```
//Backend - classe Dados
public string consultarClienteMaisSimilar(string clienteOrigem){
    conectar();
    con.Open();
    string sql = "select top 1 a.* from tbClienteSimilaridade a where a.nomeClienteOrigem = '" + clienteOrigem + "' order by similaridade desc;";
    //Seleciona 1 registro da tabela que armazena a distância entre os clientes, passando como critério o cliente de origem (selecionado na tela),
    ordenado pela maior similaridade
    SqlCommand adoCmd = new SqlCommand(sql, con);
    SqlDataReader adoDR = adoCmd.ExecuteReader(); //Cria um objeto DataReader para armazenar o resultado da BD
    if (adoDR.HasRows){//Se retornou algum registro
        if (adoDR.Read()) //Posiciona no registro retornado
            return adoDR["nomeClienteDestino"].ToString(); //Retorna o nome do cliente obtido
        else
            return "";
    }
    else
        return "";
}
```

Resultado 'Marcos'
para a cliente 'Ana'

	nomeClienteOrigem	nomeClienteDestino	similaridade
1	Ana	Marcos	0,33

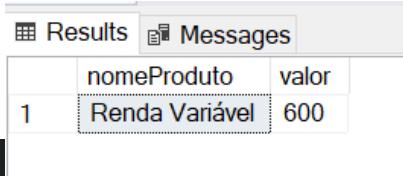
IMAGEM 65 -CLIENTE MAIS SIMILAR

DESENVOLVENDO UM SISTEMA DE RECOMENDAÇÃO

Criando uma aplicação Web para implementar e demonstrar as funcionalidades do SR

Agora que temos o nome do cliente com maior similaridade em relação ao cliente de origem, vamos fazer a recomendação do(s) produto(s) que o cliente de destino possui e o cliente de origem não possui (filtro colaborativo).

```
public DataTable listarRecomendacaoOF(string nomeClienteOrigem, string nomeClienteDestino){  
    conectar();  
    con.Open();  
    //Seleciona todos os campos da tabela tbClienteProduto onde o nomeCliente = nomeClienteDestino (mais similar) e o produto não exista para o cliente de origem  
    string sql = "select b.* from tbCliente a, tbClienteProduto b, tbProduto c ";  
    sql += "where a.nomeCliente = b.nomeCliente and b.nomeProduto = c.nomeProduto and b.nomeCliente = '" + nomeClienteDestino + "' ";  
    sql += "and b.nomeProduto not in (select b.nomeProduto from tbCliente a, tbClienteProduto b, tbProduto c ";  
    sql += "where a.nomeCliente = b.nomeCliente and b.nomeProduto = c.nomeProduto ";  
    sql += "and b.nomeCliente = '" + nomeClienteOrigem + "')";  
    SqlCommand adoCmd = new SqlCommand(sql, con);  
    SqlDataAdapter da = new SqlDataAdapter(adoCmd);  
    DataTable dt = new DataTable();  
    da.Fill(dt);  
    return dt;  
}
```



The screenshot shows a results grid with two columns: 'nomeProduto' and 'valor'. The first row contains the values 'Renda Variável' and '600'. A red arrow points from the text 'Resultado para destino 'Marcos' e origem 'Ana'' to the 'Renda Variável' cell.

Resultado para destino 'Marcos' e origem 'Ana'

IMAGEM 66 - RECOMENDAÇÃO FILTRO COLABORATIVO

DESENVOLVENDO UM **SISTEMA DE RECOMENDAÇÃO**

Criando uma aplicação Web para implementar e demonstrar as funcionalidades do SR

Neste método, utilizamos o atributo quantidadeLikes como um indicador de popularidade e usamos a ordenação deste atributo de forma Decrescente para criar um ranqueamento por relevância com base na popularidade (curtidas).

```
//Backend
public DataTable listarProdutosPopularidade() {
conectar();
con.Open();
//Lista os produtos da tbProduto em ordem da maior para a menor popularidade
string sql = "select a.* from tbProduto a order by a.quantidadeLikes desc;";
SqlCommand adoCmd = new SqlCommand(sql, con);
SqlDataAdapter da = new SqlDataAdapter(adoCmd);
DataTable dt = new DataTable();
da.Fill(dt);
return dt;
}
```

	nomeProduto	quantidadeLikes
1	Conta Corrente	3
2	Crédito Pessoal	2
3	Cartão de Crédito	2
4	Poupança	0
5	Renda Fixa	0
6	Renda Variável	0

Resultado da
listagem dos
produtos por
popularidade

IMAGEM 67 - LISTAR POPULARIDADE

DESENVOLVENDO UM SISTEMA DE RECOMENDAÇÃO

Criando uma aplicação Web para implementar e demonstrar as funcionalidades do SR

Métodos para incrementar ou decrementar o atributo quantidadeLikes, com base na ação de like ou unlike do usuário pelos botões disponíveis para interagir com o sistema.

```
//Frontend - incrementa/decrementa Likes
if (ImageButton1.ImageUrl == "~/likeNao.png"){
    ImageButton1.ImageUrl = "~/likeSim.png";
    objDados.IncrementarLike("Cartão de Crédito");}
else{
    ImageButton1.ImageUrl = "~/likeNao.png";
    objDados.DecrementarLike("Cartão de Crédito");}
```

```
//Backend
public bool IncrementarLike(string produto){
    conectar();
    cmd = new SqlCommand("UPDATE tbProduto SET quantidadeLikes =
= quantidadeLikes + 1 WHERE nomeProduto = '" + produto +
"', con");
    con.Open();
    cmd.ExecuteNonQuery();
    con.Close();
    return true;
}
```

```
//Backend
public bool DecrementarLike(string produto){
    conectar();
    cmd = new SqlCommand("UPDATE tbProduto SET quantidadeLikes =
quantidadeLikes - 1 WHERE nomeProduto = '" + produto + "'",
con);
    con.Open();
    cmd.ExecuteNonQuery();
    con.Close();
    return true;
}
```

IMAGEM 68 - INCREMENTA DECREMENTA

DESENVOLVENDO UM SISTEMA DE RECOMENDAÇÃO

Criando uma aplicação Web para implementar e demonstrar as funcionalidades do SR

Executando o projeto e validando os resultados:

The screenshot shows the Microsoft Visual Studio IDE interface. On the left, the code editor displays the file `WebForm1.aspx.cs` with several lines of C# code. Lines 40 and 41 are highlighted in red, indicating a syntax error or warning. Line 40 contains the code `string strClienteMaisSimilar = objDados.consultarClienteMaisSimilar(DropDownList1.Text, strClienteMaisSimilar);`. Line 41 contains the code `DataTable dt = objDados.listarRecomendacaoOF(DropDownList1.Text, strClienteMaisSimilar);`. The code editor also shows a tooltip for the `objDados` variable, which is defined as `POC_SR.POC_SR`.

On the right, the execution results are displayed in a browser window. The title bar says "Din Din Fintech". The page content includes a header "Olá [Ana] Ag. 1234 CC. 00012345-6 [Atualizar Cliente]". Below it, there are two tables: "Recomendação Open Finance" and "Recomendação Popularidade".

Recomendação Open Finance

nomeCliente	nomeProduto	valor
Marcos	Renda Variável	0,6

Recomendação Popularidade

nomeProduto	quantidadeLikes
Conta Corrente	4
Renda Variável	4
Crédito Pessoal	3
Cartão de Crédito	2
Poupança	0
Renda Fixa	0

Below the tables, a message asks "Está curtindo os produtos da Din Din?". It lists five products with thumbs-up icons: "Cartão de Crédito", "Conta Corrente", "Crédito Pessoal", "Poupança", "Renda Fixa", and "Renda Variável".

IMAGEM 69 – EXECUTANDO O PROJETO