Projeto MPEI

Sistema de Recomendação de Produtos

Pedro Marques nº 118895

Catarina Ribeiro nº 119467

2024/2025

---

1. Introdução

Este primeiro projeto insere-se no âmbito da disciplina de Métodos Probabilísticos para Engenharia Informática e tem como objetivo desenvolver um sistema de recomendação de produtos para plataformas de comércio eletrónico. A solução implementada combina métodos probabilísticos e técnicas de aprendizagem, como Naïve Bayes, Filtro de Bloom e MinHash, para prever categorias de interesse, evitar recomendações redundantes e personalizar sugestões com base em padrões de compra de utilizadores similares. Como ponto de partida, foram utilizados os conceitos abordados nas aulas teóricas e práticas, bem como as ferramentas recomendadas no enunciado do trabalho.

2. Descrição de como correr os vários programas

2.1. Base de Dados

Para a realização deste projeto tivemos alguma dificuldade a encontrar um dataset que se encaixasse na nossa ideia, encontramos um mas tinha poucos produtos e avaliações então tivemos de completá-lo com mais produtos e mais categorias:

- Dataset.csv: este ficheiro contém a avaliação de vários produtos de várias categorias feitas por vários utilizadores. Para a implementação deste dataset neste projeto vamos focar-nos mais nos utilizadores, nas categorias e na avaliação média de cada produto.

2.2. Naïve Bayes

O objetivo do Naïve Bayes é **prever** a **categoria "preferida"** de um utilizador específico com base nos dados de compras (produtos comprados e as suas avaliações).

- As categorias de produtos (por exemplo, "Eletrônicos", "Roupas", etc.) são **transformadas em índices numéricos** para serem utilizadas por este modelo.
- Filtra-se o conjunto de dados para obter apenas os **produtos comprados** pelo utilizador selecionado.

- As avaliações (ratings) associadas a esses produtos são usadas como características, e as categorias são os valores a serem previstos.
- Calcula-se a **probabilidade a priori** de cada categoria, ou seja, a proporção de produtos em cada categoria.
- Calcula-se também a **probabilidade condicional** de um valor de avaliação (rating) dado uma categoria específica. Para evitar probabilidades nulas, é adicionando um pequeno valor.
- Utiliza-se um valor de avaliação hipotético (**4.5**) para atualizar as **probabilidades logarítmicas** de cada categoria, que transforma as multiplicações em somas o que melhora a **estabilidade numérica**, a **eficiência** e a **robustez** deste método.
- Com base nas probabilidades calculadas, o programa determina a **categoria mais provável** para um produto associado à avaliação fornecida.

Para testar este método basta correr o *script Naive_Bayes.m* que irá pedir um um utilizador dentro dos *UserID* disponíveis no ficheiro *Dataset.csv* e fazer a previsão da categoria "preferida" desse utilizador em específico.

2.3. Filtro de Bloom

O objetivo do Filtro de Bloom é **eliminar** da lista de recomendações os produtos que o utilizador **já comprou**, de forma eficiente, e selecionar os **10 produtos com as melhores avaliações** (ratings) na categoria que foi prevista no Naïve Bayes.

- Inicializa-se um vetor binário (filtro) de tamanho fixo de 1540, onde inicialmente todos os bits são zero.
- Para cada produto que o utilizador já comprou, calcula-se um **índice hash** (posição no vetor Bloom Filter) usando a função hash_function.
- O **índice hash** é obtido somando os valores ASCII dos caracteres do ID do produto e calculando o módulo do tamanho do filtro.
- A posição correspondente no vetor (filtro) é marcada como **verdadeira** (indicando que o produto foi "inserido" no Bloom Filter).
- Filtra-se o conjunto de dados para incluir apenas os produtos pertencentes à categoria prevista.
- Para cada produto na lista filtrada, o código verifica se ele **já foi comprado** pelo utilizador consultando o **Bloom Filter.**

Para testar este método basta correr o *script Bloom_Filter.m* que através de algumas informações dadas pelo Naïve Bayes vai retornar uma tabela com recomendações de produtos da categoria "preferida" do utilizador priorizando os produtos com melhores avaliações.

2.4. Similaridade utilizando Minhash

O objetivo do Minhash é calcular a similaridade entre utilizadores e recomendar produtos relevantes com base nos produtos que os utilizadores mais semelhantes compraram dentro da categoria que foi prevista no Naïve Bayes.

- Cada utilizador é representado por um **conjunto de produtos comprados**.

- O método MinHash é usado para criar **assinaturas compactas** (um vetor de valores mínimos de hash) para cada utilizador.
- As assinaturas servem como **resumos** dos conjuntos de produtos, permitindo a comparação eficiente entre utilizadores.
- A **similaridade de Jaccard** aproximada entre utilizadores é calculada comparando suas **assinaturas MinHash**. A similaridade representa a sobreposição entre os conjuntos de produtos comprados pelos utilizadores.
- O algoritmo seleciona os **5 utilizadores mais semelhantes** ao utilizador alvo, com base nas suas assinaturas MinHash.
- Os produtos comprados pelos utilizadores mais semelhantes são recolhidos.
- Os produtos que o utilizador alvo **já comprou** são removidos da lista.
- A lista resultante é filtrada para incluir apenas os produtos da **categoria prevista**.

Para testar este método basta correr o *script Minhash.m* que com as informações fornecidas pelos outros métodos vai retornar uma tabela com recomendações de produtos da categoria "preferida" do utilizador com os produtos dessa categoria que já foram comprados pelos utilizadores mais semelhantes, e também retorna a similaridade que os utilizadores escolhidos têm com o utilizador alvo.

2.5. Aplicação Conjunta

Para testar este programa basta correr o *script ap_conjunta.m* que vai gerar um utilizador aleatoriamente através dos *UserIDs* presentes no ficheiro *Dataset.csv* (de 100 a 149). Na parte do Naïve Bayes vai retornar a categoria prevista como sendo a "preferida" do utilizador escolhido. Esta informação vai ser utilizada para fazer o Filtro Bloom e assim retornar uma tabela com os produtos dessa categoria que ainda não foram comprados pelo utilizador priorizando os produtos com as melhores avaliações. Vai ser também utilizada para o Minhash que vai retornar uma tabela com os produtos que os utilizadores mais semelhantes ao utilizador escolhido compraram dentro da categoria prevista e retorna também a similaridade dos utilizadores mais semelhantes ao utilizador inicialmente escolhido.

3. Análise dos resultados obtidos nos vários testes

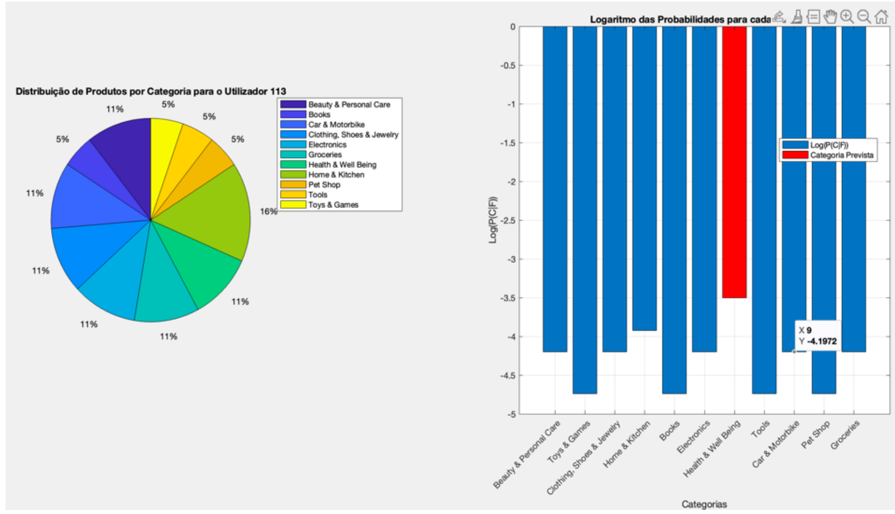
Foi criado um *script Testes.m* com o objetivo de reforçar a veracidade das informações dadas pelos métodos utilizados para este projeto, que basta simplesmente corre-lo.

3.1. Teste do Naïve Bayes

Para fazer este teste vai ser corrido o *script Naive_Bayes.m* e vai ser pedido no terminal para inserir o *UserId* de um utilizador (de 100 a 149) e só para quando é inserido um número válido. Após isto, é retornada a categoria prevista e dois gráficos. O primeiro gráfico mostra a percentagem de compras feitas pelo utilizador inicialmente escolhido por categoria, o segundo gráfico mostra as probabilidades logarítmicas calculadas para mostrar a justificação de ter sido prevista aquela categoria. Este exemplo feito para o utilizador 113 mostra que o utilizador fez mais compras na categoria *Home & Kitchen* mas a categoria prevista foi *Health & Well Being* pois as avaliações (ratings) dadas aos produtos da categoria *Health & Well Being* foram, em média, **mais altas**.

```
>> Testes
Insira o UserID de utilizador (de 100 a 149): 1234
Erro: O UserID deve ser um número inteiro entre 100 e 149.
Insira o UserID de utilizador (de 100 a 149): 98
Erro: O UserID deve ser um número inteiro entre 100 e 149.
Insira o UserID de utilizador (de 100 a 149): 119.7
Erro: O UserID deve ser um número inteiro entre 100 e 149.
Insira o UserID de utilizador (de 100 a 149): 113

Predicted Category: Health & Well Being
```



3.2. Teste do Filtro Bloom

Utilizando as informações dadas pelo Naïve Bayes o *sricpt Bloom_Filter.m* é corrido e retorna uma tabela com recomendações e é criada uma lista (lista_de_produtos) que contém todos os produtos comprados pelo utilizador inicialmente escolhido. Este teste consiste em verificar se as duas listas (lista_de_produtos e a lista com as recomendações) têm algo em comum, caso não tenham, confirma que as recomendações dadas pelo Filtro Bloom não contém nenhum produto que já tenha sido comprado pelo utilizador. Se o teste for bem-sucedido é impresso no terminal "Passou no teste." se não é impresso "Não passou no teste". Continuando com o exemplo do utilizador 113 podemos ver que passou no teste.

```
                              Name
_____________________________________________________________________________

{'Amazon Essentials Men's Slim-Fit Long-Sleeve T-Shirt'                     }
{'wet n wild MegaGlo Dual-Ended Contour Stick, Light Medium, Cruelty-Free'}
{'King of Sloth (Kings of Sin, 4)'                                          }
{'Under Armour Women's Ignite Select Slide Sandal'                          }
{'Câmera de Segurança TP-Link'                                              }
{'Lâmpada Inteligente Wi-Fi Positivo Casa'                                  }
{'Batedeira Planetária Oster'                                               }
{'Batedeira Planetária Oster'                                               }
{'Conjunto de Potes Herméticos de Vidro'                                    }
{'Escova Alisadora Philco'                                                  }
{'Jogo de Tabuleiro Monopoly Hasbro'                                        }
{'Termômetro Digital Infravermelho'                                         }
{'Almofada Ortopédica para Lombar'                                          }
{'Alicate de Corte Vonder'                                                  }
{'Aspirador de Carro Portátil'                                              }
{'Câmera de Ré com Monitor'                                                 }
{'Bebedouro Fonte para Gatos'                                               }
{'Café em Cápsulas Nespresso Ristretto'                                     }
{'Café em Cápsulas Nespresso Ristretto'                                     }
```

Produtos da categoria Health & Well Being com as melhores avaliações.

Nº	Nome do Produto	ID Produto	Avaliação	Preço	Disponibilidade
1	Massageador Elétrico Portátil	99	4.9	40.00	Out of Stock
2	Travesseiro de Espuma da NASA	103	4.6	20.00	In Stock
3	Balança Digital Corporal Xiaomi	100	4.4	35.00	In Stock
4	Aparelho de Pressão Automático Omron	101	3.6	50.00	In Stock
5	Suplemento de Vitamina C Sundown	95	3.4	10.00	In Stock
6	Oxímetro de Dedo Portátil	97	3.2	15.00	In Stock
7	Colchão Ortopédico Casal	102	3.2	300.00	In Stock

```
--------------------------

Passou no teste!

--------------------------
```

3.3. Teste do Minhash

Utilizando as informações dadas pelo Naïve Bayes o *sricpt Minhash.m* é corrido e retorna uma tabela com recomendações e é criada uma lista (elementos) que contém todos os produtos comprados pelos 5 utilizadores mais semelhantes ao utilizador inicialmente escolhido. Este teste consiste em verificar se os elementos da lista de recomendações estão dentro da lista "elementos", se estão, confirma que o Minhash só recomenda produtos que foram comprados pelos utilizados mais semelhantes. Se o teste for bem-sucedido é impresso no terminal "Passou no teste." se não é impresso "Não passou no teste". Continuando com o exemplo do utilizador 113 podemos ver que passou no teste.

```
                Produtos recomendados com base no utilizador mais semelhante da categoria Health & Well Being.
=================================================================================================================
| Nº  | Nome do Produto                                | ID Produto | Avaliação | Preço     | Disponibilidade |
-----------------------------------------------------------------------------------------------------------------
| 1   | Colchão Ortopédico Casal                       | 102        | 3.2       | 300.00    | In Stock
| 2   | Faixa Abdominal Pós-Cirúrgica                  | 104        | 3.1       | 25.00     | Out of Stock
| 3   | Massageador Elétrico Portátil                  | 99         | 4.9       | 40.00     | Out of Stock
| 4   | Travesseiro de Espuma da NASA                  | 103        | 4.6       | 20.00     | In Stock
-----------------------------------------------------------------------------------------------------------------

===== Utilizadores Semelhantes =====
Usuário selecionado: 113
Similaridade com usuário 120: 0.18
Similaridade com usuário 123: 0.16
Similaridade com usuário 141: 0.15
Similaridade com usuário 148: 0.14
Similaridade com usuário 105: 0.13

--------------------------

Passou no teste!

--------------------------
```

4. Vantagens e limitações das soluções propostas

Vantagens:

1. **Eficiência e Escalabilidade**:
   - Métodos como Naïve Bayes, Bloom Filter e MinHash são rápidos, econômicos e lidam bem com grandes volumes de dados.
2. **Uso Eficiente de Recursos**:
   - Naïve Bayes transforma as categorias em índices para ser mais fácil na implementação do código.
   - Bloom Filter economiza memória ao evitar produtos já comprados.
   - MinHash reduz o custo de comparação entre utilizadores.
3. **Recomendações Personalizadas**:
   - Combina previsões baseadas em categorias (Naive Bayes) e recomendações colaborativas (MinHash), garantindo resultados relevantes.
4. **Resultados Práticos**:
   - Gera listas claras de produtos recomendados e destaca utilizadores similares.

Limitações:

1. **Dependência de Parâmetros**
   - O tamanho do Bloom Filter afeta a taxa de falsos positivos.
   - O número de funções hash em MinHash influencia a precisão da similaridade calculada.
2. **Custo Computacional Inicial**
   - Apesar de eficiente na execução, o cálculo inicial das **assinaturas MinHash** e a construção do **Bloom Filter** podem ser custosos em grandes conjuntos de dados.
3. **Dependência de Dados de Qualidade**
   - O sistema depende da disponibilidade de dados precisos e consistentes. Ratings incompletos, erros nos dados ou pouca sobreposição entre utilizadores podem comprometer as recomendações.

5. Conclusão

Em suma, o sistema de recomendação de produtos desenvolvido neste projeto integra três métodos complementares Naïve Bayes, Filtro de Bloom e MinHash para prever, filtrar e refinar sugestões personalizadas. Os testes realizados confirmaram a eficácia e eficiência da solução, que apresentou recomendações precisas, evitou redundâncias e identificou padrões de compra de forma eficiente. Apesar de algumas limitações, a solução proposta é escalável e representa uma ferramenta robusta para sistemas de recomendação em plataformas de comércio eletrónico.