

Felipe Fraxe

Documentação Case Meliuz

Manaus, AM
2024

Contextualização

Após analisar diversas métricas de negócio, o CEO da xpto.com.br observou uma tendência preocupante: apesar do aumento contínuo no número de novos usuários, o faturamento mensal vem caindo. Com a volumetria massiva de produtos disponíveis, esses usuários estão tendo dificuldades em descobrir itens relevantes após o cadastro na plataforma.

Em uma reunião de emergência, o CEO apontou que um dos principais problemas é a incapacidade do sistema atual em recomendar produtos de forma eficaz para usuários recém cadastrados. O mecanismo de recomendação não está acompanhando a variedade e o volume dos produtos, resultando em sugestões irrelevantes para novos usuários.

O objetivo deste teste prático é ajudar o CEO da xpto.com.br a conter a queda no faturamento, aumentando o número de compras através do desenvolvimento de um mecanismo de recomendação de produtos. Este mecanismo deve ser capaz de lidar com a grande volumetria de produtos e a alta taxa de acessos diários, oferecendo sugestões relevantes para os usuários novos.

Entendendo os Dados

Como informação base, tem-se os dados de vendas diárias dos meses de março, abril e maio de 2024, dispostos em um arquivo .csv com a seguinte estrutura:

- Data da venda (sale_date)
- ID numérico do produto (product_id)
- Título do produto (product_title)
- Preço (product_price)
- Imagem (product_image_url)
- Nome da loja (store_name)
- ID numérico da loja (store_id)
- Vendas no dia (sales_per_day)

Tais dados serão usados como fonte para o novo algoritmo de recomendação.

Algoritmo de Recomendação

Com base no exposto acima, para atacar o problema, o possível algoritmo de recomendação implementado leva em consideração duas métricas principais, possíveis de extrair da base de dados: Crescimento do volume de vendas de um produto no últimos 3 meses e sua receita total.

O algoritmo pode ser dividido em três etapas principais, sendo elas:

1. Cálculo da taxa de crescimento – Obtida a partir de um algoritmo de regressão linear com o número de vendas dos três meses disponíveis (a escolha desse algoritmo se dá pelo fato de também incorporar meses onde houve queda na receita, diferente de uma taxa de crescimento mensal composta – CMGR, por exemplo).
2. Filtragem de produtos – Retira-se das possíveis sugestões produtos com taxa de crescimento negativa ou flat (0). A partir dos produtos restantes, encontra-se um limiar mínimo de crescimento, calculado pelo quartil de 25%, ou seja, o quarto de produtos com menor taxa de crescimento de vendas também é retirado.
3. Ordenação do ranking – Por fim, ordena-se os produtos restantes pela receita trazida pelas vendas, ou seja, média de preços x quantidade total de vendas nos meses analisados.

Arquitetura do Projeto

O projeto foi desenvolvido seguindo a arquitetura MSC (Model-Service-Controller), uma estrutura organizada que separa claramente as responsabilidades de cada camada, sendo elas:

- Model: Também conhecida como camada de Dados, É responsável pelo carregamento e processamento de informações do banco de dados ou de fontes de dados externas (neste caso, um arquivo CSV com dados de vendas). O modelo manipula os dados, realizando

agregações e transformações necessárias para que as informações possam ser utilizadas pela camada seguinte.

- **Service:** Contém a lógica e regras de negócios. Aqui há a implementação do algoritmo de recomendação. O serviço processa os dados, calcula o crescimento de vendas dos produtos com base na regressão linear, aplica as regras de filtragem e ordena dos produtos.
- **Controller:** É responsável por lidar com as requisições HTTP e fornecer as respostas adequadas aos clientes da API. Sua principal função é receber as requisições dos usuários, chamar os serviços necessários e retornar os resultados formatados via JSON para o cliente.

Instruções de uso da API via endpoints

Requisição: O cliente faz uma requisição HTTP para o endpoint da API.

GET /recommend/<user_id>, onde user_id é um id numérico. (Para simplificação, não há verificação em um banco de dados real se o usuário é válido ou não). A resposta esperada é um array com os 5 produtos recomendados seguindo a seguinte estrutura:

- ID numérico do produto (product_id)
- Título do produto (product_title)
- Preço (product_price) – Menor preço encontrado na base de dados
- Imagem (product_image_url)
- Nome da loja (store_name) – Loja onde ocorreu a venda com menor preço
- ID numérico da loja (store_id)

Que são as informações necessárias tratáveis pelo front end para o usuário final.

OBS: Para simplificação, a requisição GET / já faz um redirecionamento para rota de de recomendação.

OBS2: Instruções de como executar o projeto e comentários mais detalhados de sua estrutura estão presentes no arquivo README.md

Limitações Atuais da Solução e Possíveis Evoluções

1. Dados Limitados a Três Meses:

O algoritmo de recomendação utiliza apenas os dados de vendas de março, abril e maio de 2024, limitando a capacidade do sistema de identificar tendências de longo prazo ou sazonalidades. Produtos que têm um comportamento cíclico ou que se beneficiam de promoções específicas podem não ser corretamente recomendados devido à curta janela de análise.

Expandir o período de análise de vendas para incluir dados de mais meses ou até mesmo anos, ajudaria o algoritmo a detectar tendências sazonais, produtos com crescimento de longo prazo, ou variações de preço mais complexas, possivelmente o tornando mais robusto e capaz de fazer recomendações melhores.

2. Falta de Personalização:

O mecanismo de recomendação não leva em consideração o comportamento individual do usuário. Todos os usuários recebem as mesmas recomendações, baseadas apenas no crescimento de vendas e receita dos produtos. Isso pode gerar sugestões menos precisas para novos usuários, que podem ter interesses variados.

Integrar dados comportamentais dos usuários, como histórico de compras, itens visualizados e preferências, permitiria a personalização das recomendações, oferecendo produtos mais adequados ao gosto de usuários com perfis parecidos.

3. Modelo Simples de Receita Total:

A receita total de cada produto é calculada como a média do preço multiplicada pelo total de vendas. Esse cálculo pode ser uma simplificação inadequada para produtos com variações significativas de preço entre lojas ou que estão sujeitos a promoções pontuais.

A aplicação de modelos de aprendizado de máquina poderia melhorar significativamente as recomendações, especialmente ao lidar com grandes volumes de produtos e dados de comportamento de usuários.

4. Crescimento Flat ou Negativo:

Produtos com vendas estáveis ou em declínio são automaticamente removidos. No entanto, alguns produtos que possuem vendas estáveis ou que tiveram um declínio temporário, mas são populares ou altamente avaliados, podem ser relevantes e, na prática, deveriam ser recomendados.

Em vez de simplesmente remover produtos com crescimento negativo ou flat, o sistema poderia utilizar métricas mais sofisticadas, como a popularidade de um produto, sua nota de avaliação ou seu número de visualizações. Isso traria mais flexibilidade ao sistema e evitaria que produtos potencialmente populares fossem excluídos da recomendação.