Universidade Presbiteriana Mackenzie

Desenvolvimento de um Algoritmo de Recomendação de Receitas a partir das compras realizadas no mercado.

Aparecida Vânia de Jesus **RA:**10407484

Lucas Gomes Porfírio da Silva **RA**: 10370475

Vanessa Hacklauer de Aguiar **RA**:10407324

Wagner de Mendonça Trindade **RA:** 10407917

São Paulo

2024

Resumo

Este estudo desenvolveu um algoritmo de recomendação de receitas, que utiliza bibliotecas de mineração de dados e análise estatística para oferecer recomendações personalizadas baseadas nas compras de ingredientes dos usuários. O algoritmo analisa o histórico de compras para identificar as preferências de ingredientes do usuário, compara essas informações com um banco de dados de receitas, e recomenda as três receitas mais relevantes que utilizam esses ingredientes. O processo é projetado para se aprimorar iterativamente, adaptando-se às preferências do usuário à medida que mais dados são coletados.

Contudo, o desenvolvimento enfrentou desafios como a ausência de uma API para acessar dados de receitas e avaliações e a falta de dados de avaliação dos usuários, que limitavam a implementação de algoritmos de recomendação mais avançados e a avaliação de desempenho. O estudo alcançou o objetivo proposto mostrando que existem alternativas eficientes de recomendação por meio de filtragem baseada em conteúdo, permitindo personalização significativa sem a necessidade de dados de outros usuários.

Palavras-chave: Algoritmo de Recomendação, Banco de Dados, Aprendizado de Máquina, Mineração de dados

São Paulo
2024

Sumário

Introdução

No contexto atual, marcado por desafios globais relacionados à sustentabilidade ambiental, a questão do consumo responsável emerge como um dos pilares essenciais para o equilíbrio entre as necessidades humanas e a capacidade de regeneração do planeta. A Agenda 2030 para o Desenvolvimento Sustentável, estabelecida pelas Nações Unidas, destaca entre seus Objetivos de Desenvolvimento Sustentável (ODS), a meta de "Consumo e Produção Responsáveis" (item 12), com um enfoque particular na redução do desperdício de alimentos. Dentro deste objetivo, o subitem 12.3 desafia o mundo a "reduzir pela metade o desperdício de alimentos per capita mundial, nos níveis de varejo e do consumidor, e reduzir as perdas de alimentos ao longo das cadeias de produção e abastecimento, incluindo as perdas pós-colheita" até 2030.

Este objetivo ressalta a importância de abordagens inovadoras que possam contribuir significativamente para a redução do desperdício de alimentos, especialmente no âmbito doméstico, onde as decisões individuais de consumo têm um impacto direto no problema em escala global.

Neste cenário, o desenvolvimento de tecnologias que promovam práticas de consumo mais conscientes e eficientes torna-se crucial. Uma das abordagens promissoras nesse sentido é a implementação de algoritmos de recomendação personalizada, especialmente aqueles voltados para a recomendação de receitas baseadas nos hábitos de compra dos usuários.Hoje, o que estas interfaces oferecem ultrapassa as opções que o usuário escolheria, como quem manuseia um hipertexto capaz de remeter a caminhos desconhecidos (LANDOW, 1992). Esses sistemas de recomendação têm o potencial de alinhar as preferências culinárias individuais com práticas de consumo responsável, contribuindo para a minimização do desperdício de alimentos nas residências.

Os algoritmos de recomendação de receitas operam coletando e analisando dados sobre as compras dos usuários, identificando padrões de consumo e preferências alimentares. Essa análise permite a criação de perfis de usuários que refletem seus interesses culinários e restrições dietéticas.

Com base nesses perfis e num banco de dados de receitas que detalha os ingredientes necessários para cada prato, o sistema é capaz de sugerir receitas que utilizem os ingredientes já adquiridos pelos consumidores. Estas recomendações avaliam os interesses do futuro através do passado, recuperando - o como um tipo de recordação. Uma sugestão remete à lembrança, repetindo - se um instante anterior, acessível como reminiscências que transformam em um todo coerente o comportamento, este conjunto caótico de ações pregressas. O cotidiano parece carente de justificativa, mas a mecanização

visa a certa ordem (ALEXANDER, 2016). Esta abordagem não apenas personaliza a experiência culinária para cada usuário, mas também incentiva o uso integral dos alimentos comprados, reduzindo as chances de que estes acabem desperdiçados.

A relevância desses sistemas de recomendação estende-se além da conveniência pessoal, tocando diretamente na questão do desperdício alimentar doméstico. Ao promover o uso eficiente dos ingredientes disponíveis em casa, esses algoritmos contribuem para uma redução significativa do volume de alimentos descartados, alinhando-se com as metas do ODS 12.3.

O papel dessas tecnologias é especialmente crítico em um momento em que a produção alimentar mundial enfrenta pressões crescentes devido ao aumento da população global, à escassez de recursos naturais e às mudanças climáticas.

Este trabalho propõe analisar o impacto potencial dos algoritmos de recomendação de receitas na promoção do consumo e produção responsáveis, com foco na redução do desperdício de alimentos no contexto doméstico.

Será explorada a união entre tecnologia, comportamento do consumidor e sustentabilidade, visando compreender como soluções inovadoras podem contribuir para os esforços globais de combate ao desperdício alimentar e, por extensão, para a construção de um futuro mais sustentável para todos.

1. Dataset

Após pesquisas em dados abertos, não conseguimos localizar dados que possam alimentar o treinamento do algoritmo, desta forma, os dois Datasets foram construídos pelo grupo.

O primeiro é composto por 1.000 itens que simulam as compras de uma pessoa em um supermercado, excluídos os itens de higiene, já que estes não são importantes para o treinamento e execução. Os itens foram selecionados entre os components da sexta básica do brsileiro e nos itens mais comprados: compostos por "laranja", "banana", "maça", "pera","abacaxi", "manga", "tangerina", "melão", "limão", "abóbora", "repolho", "alface", "chuchu", "batata-doce", "pimentão", "batata", "tomate", "cebola", "quiabo", "couve-flor", "berinjela", "melancia", "ervilha", "jiló", "leite condensado","creme de leite", "chocolate","chocolate branco","chocolate granulado", "leite de coco", "coco ralado", "ovos", "farinha de trigo", "farinha de milho", "farinha de mandioca", "massas", "pão francês", "leite integral", "leite em pó", "iogurte", "leite fermentado", "queijos", "leite integral", "manteiga", "arroz", "feijão", "café", "açúcar", "óleo de soja", "óleo vegetal", "macarrão", "molho de tomate", "sal de cozinha", "fubá de milho", "carne bovina","carne suína", "frango", "peixe", "linguiça", "salsicha". A seleção foi feita de forma aleatória, sendo todas as compras com 15 destes itens.

O segundo é um dataset com 49 receitas variadas distribuídas entre 50 receitas tradicionais brasileiras, abrangendo saladas, sobremesas, doces, tortas, pratos principais e lanches. Composto pelo nome da receita e pelos ingredientes de cada receita e o valor total, simulado sem base em valores reais.

Desta forma criamos os dados de entrada para treinamento e teste e os dados para consulta e efetivação da recomendação.

2. Análise Exploratória

2.1 Base Compras

Utilizamos a biblioteca Pandas-Profiling que mudou para Pandas Ydata, e dentro desta biblioteca utilizei o *ProfileReport* para gerar uma análise de dados.

Temos três variáveis - duas numéricas e uma texto, sem duplicatas nem valores nulos em um total de mil observações:

Overview

Overview | Alerts 3 | Reproduction

Dataset statistics

Number of variables	3
Number of observations	1000
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	23.6 KiB
Average record size in memory	24.1 B

Variable types

Numeric	2
Text	1

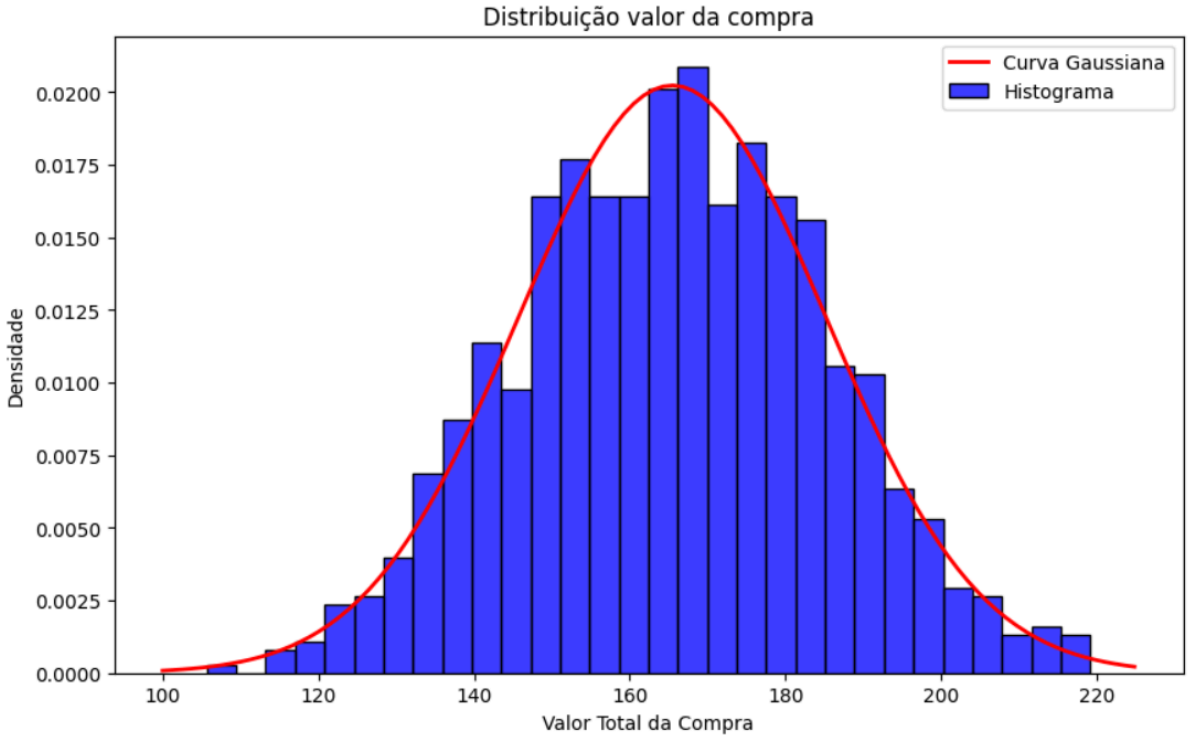
A variável ‘valor da compra’ apresenta uma distribuição Gaussiana, na qual podemos verificar que a maioria das observações estão entre os valores 150 e 180.

valor_total_da_compra

Real number (ℝ)

Distinct	941	Minimum	105.67
Distinct (%)	94.1%	Maximum	219.13
Missing	0	Zeros	0
Missing (%)	0.0%	Zeros (%)	0.0%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	165.45133	Memory size	7.9 KiB

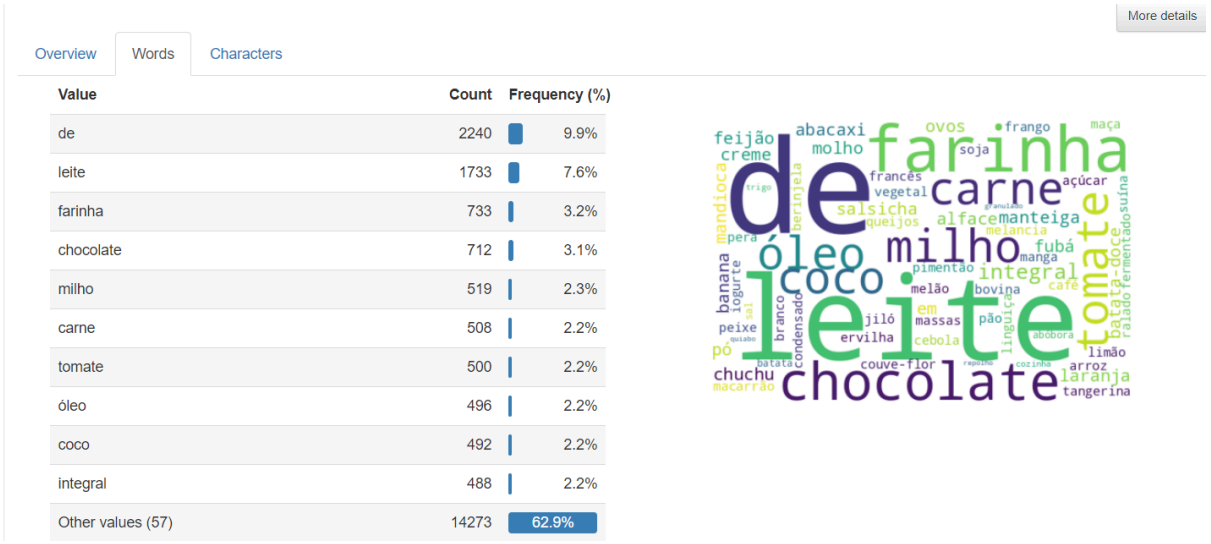
Histograma:



Dados gerados pelos autores

A coluna Itens comprados, que é uma coluna trabalhada e agregou todas as colunas itens em uma só é composta com valores únicos.

O Report gerou uma gráfico de nuvem de palavras onde podemos ver as palavras que mais aparecem são farinha, leite, chocolate, milho, carne, óleo, coco e a palavra de:



As análises de correlação e interação não geraram dados que possam gerar insights.

2.2 Base Receitas

Neste dataset temos três variáveis sendo uma numérica e duas texto. Sem duplicatas nem valores nulos em um total de 49 observações:

Overview

Overview | Alerts 2 | Reproduction

Dataset statistics

Number of variables	3
Number of observations	49
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	1.3 KiB
Average record size in memory	26.6 B

Variable types

Numeric	1
Text	2

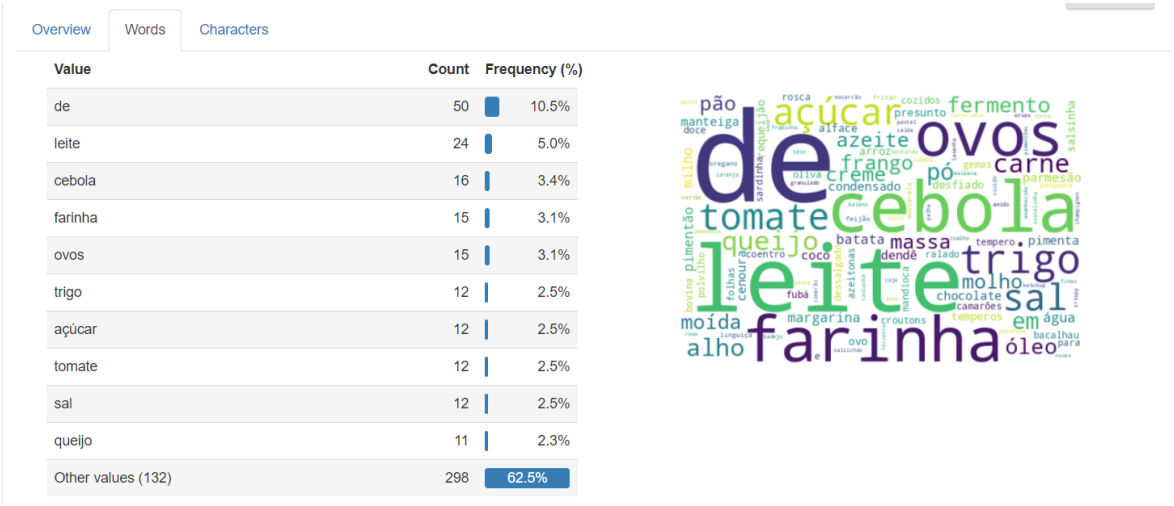
A coluna ‘nome da receita’ composta de valores únicos, foi gerado um gráfico de nuvem de palavras na qual identificamos a predominância das palavras de, carne, com, bolo, bolinho, sanduíche, panqueca e arroz:

More details

Overview | Words | Characters

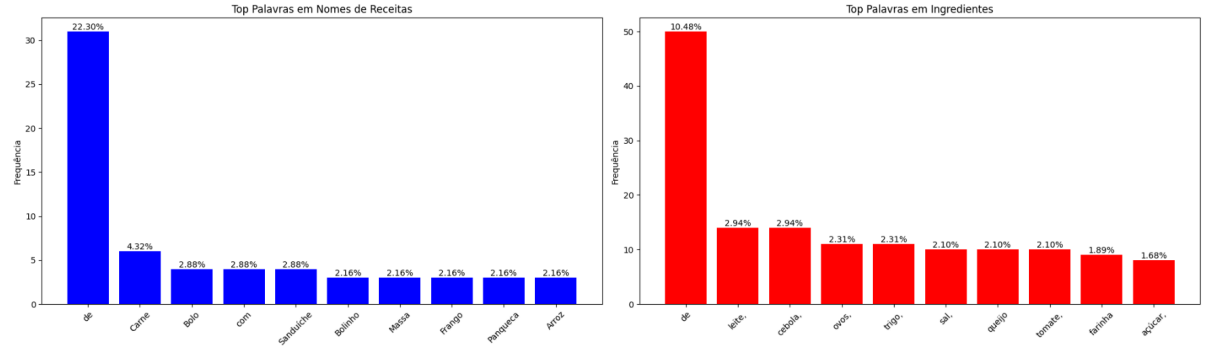
Value	Count	Frequency (%)
de	31	22.3%
carne	6	4.3%
com	4	2.9%
bolo	4	2.9%
sanduíche	4	2.9%
bolinho	3	2.2%
arroz	3	2.2%
frango	3	2.2%
massa	3	2.2%
panqueca	3	2.2%
Other values (63)	75	54.0%

A coluna ‘ingredientes’ também composta de variáveis do tipo string gerou um gráfico de nuvem de palavras onde vemos a predominância das palavras farinha, leite, ovos, cebola, trigo, sal, de e tomate:



Overview | Words | Characters

Value	Count	Frequency (%)
de	50	10.5%
leite	24	5.0%
cebola	16	3.4%
farinha	15	3.1%
ovos	15	3.1%
trigo	12	2.5%
açúcar	12	2.5%
tomate	12	2.5%
sal	12	2.5%
queijo	11	2.3%
Other values (132)	298	62.5%



Todos os gráficos gerados pelos autores ou pela ferramenta *ReportProfiling.*

3. Algoritmos de Recomendação

O algoritmo de recomendação é uma ferramenta útil e muito utilizada atualmente em sites e aplicativos para indicar produtos que possam atrair os clientes. Os algoritmos de recomendação são multidisciplinares, envolvendo conhecimentos em software, ciência de dados, matemática, estatística, sociologia e psicologia, com as técnicas certas, eles podem aprimorar significativamente a experiência do usuário em plataformas online.

Existem vários tipos de Algoritmos de recomendação, eles podem utilizar várias técnicas.

A técnica de recomendação baseada no conteúdo onde as informações existentes no perfil do usuário ou as características dos itens são utilizados para procurar uma correlação entre eles e os novos itens ainda desconhecidos dos usuários para que ocorra a recomendação. Ele combina os atributos do perfil do usuário com os atributos dos itens e obtém um critério de relevância que varia entre 0 e 1, este critério é utilizado para representar o grau de interesse do usuário pelo item.

A técnica de filtragem colaborativa é baseada em encontrar padrões nas respostas dos usuários em comparação às respostas do usuário que receberá a recomendação. Quando se obtém um histórico do seu usuário, é possivel fazer recomendações com base nas escolhas/consumo de usuários com gostos semelhantes.

A técnica de filtragem híbrida é a combinação das duas técnicas acima(Baseada em conteúdo e a filtragem colaborativa).

A utilização de Redes Neurais para Recomendações Baseadas em Conteúdo envolve recomendar itens com base em suas características, como descrições de produtos, gêneros, entre outros. As redes neurais são empregadas para aprender representações latentes desses itens a partir de suas características.

As Redes Neurais Convolucionais (CNNs) são um tipo de rede neural especialmente eficaz no processamento de dados com uma estrutura de grade, como imagens. CNNs utilizam filtros convolucionais para capturar padrões espaciais dentro dos dados, tornando-as excelentes para analisar visuais de produtos ou características espaciais em dados.

As Redes Neurais Recorrentes (RNNs) são projetadas para lidar com sequências de dados, como texto ou séries temporais. Elas têm a capacidade única de manter informações de estados anteriores, o que é útil para entender o contexto em

descrições de produtos ou avaliações textuais, permitindo que a rede faça previsões baseadas em sequências de palavras ou ações ao longo do tempo. Os modelos específicos, como CNNs e RNNs, podem ser aplicados de maneira eficaz para capturar e representar as nuances dessas características.

3.1 Referencial teórico

A técnica de recomendação empregada neste estudo é sofisticadamente centrada na análise dos padrões de compra dos usuários, visando selecionar receitas que não apenas correspondam aos seus gostos, mas também atendam às suas necessidades culinárias. Para alcançar esse objetivo, exploramos uma abordagem híbrida que combina metodologias baseadas em filtragem colaborativa com técnicas avançadas de aprendizado de máquina e processamento de linguagem natural (PLN).

A filtragem colaborativa é fundamentada na premissa de que usuários com preferências semelhantes em avaliações passadas tendem a exibir interesses congruentes no futuro. Esta técnica, ao utilizar as avaliações e interações coletivas de todos os usuários, permite prever com eficácia o interesse de um usuário específico em determinados itens, como receitas. Tal abordagem é enriquecida pela implementação de algoritmos de aprendizado de máquina, que aprimoram a capacidade de identificar padrões complexos nas preferências dos usuários, sugerindo assim receitas altamente relevantes.

Adicionalmente, o PLN desempenha um papel crucial ao analisar as descrições das receitas e os ingredientes contidos nelas. Essa análise permite uma compreensão mais profunda das preferências culinárias do usuário, facilitando recomendações que são não apenas relevantes, mas também personalizadas de acordo com os ingredientes preferidos ou disponíveis. Utilizamos o `CountVectorizer` para transformar descrições de texto em representações numéricas que são comparadas utilizando a métrica de similaridade de cosseno. Esse processo possibilita a identificação de receitas que compartilham características semânticas com os itens presentes na lista de compras do usuário.

Ao combinar essas técnicas de filtragem colaborativa, aprendizado de máquina, e PLN, nossa técnica de recomendação transcende a simples análise de padrões de compra. Garantimos recomendações de receitas que são não apenas precisas, mas também altamente personalizadas, promovendo uma experiência culinária enriquecedora e adaptada às preferências individuais de cada usuário.

4. Metodologia

O código para preparação dos dados é composto de vários passos que serão definidos como processos que serão utilizados para adaptar os dois datasets de maneira que sejam entendidos de forma igual e facilite a comparação dos ingredientes nas duas listas. O processo envolve várias etapas de pré-processamento para deixar os dados em um formato adequado:

- **Normalizar e Tokenizar os Ingredientes das Receitas**

Normalizamos a coluna de ingredientes no dataset, que contém listas de ingredientes em formato de texto.

Em seguida realizamos o processo de tokenização dessa coluna, ou seja, separar os ingredientes em listas de strings individuais, como converter tudo para minúsculas para garantir a consistência.

- **Preparar o Dataset de Compras/Receitas**

No dataset de Compras, transformamos os dados de forma que, todas as colunas chamadas de ‘item..’ foram agregadas em uma só coluna, utilizando a biblioteca Panda e viraram strings dentro de listas.

No dataset de Receitas, como todos os ingredientes já estavam na mesma coluna, foi necessário somente transformálos em strings dentro de listas.

- **Converter os dados para o mesmo padrão**

Utilizamos uma função na qual empregamos a biblioteca do python *re* para remover caracteres especiais dos ingredientes das duas tabelas de dados, como por exemplo traços, espaços extras, parênteses, a palavra e também foi removida, acentos e vírgulas no meio de palavras. Esta função foi aplicada nos dois datasets.

Criamos uma coluna ingredientes_str com os dados normalizados.

- **Tokenizar os dados**

O tokenizer foi utilizado para que o texto fosse dividido em tokens. Utiliza a função lambda para dividir o texto pelo espaço em branco, para que cada palavra seja tratada como um token individual.

Utilizamos o *CountVectorizer* , uma classe do scikit-learn usada para converter uma coleção de documentos de texto em uma matriz de contagens de tokens. Utilizada para transformar texto em um formato que pode ser manipulável por algoritmos de machine learning.

Ele é usado para criar um vetor para cada documento, onde cada dimensão do vetor corresponde a uma palavra no vocabulário total do corpus e cada valor nesse vetor representa a frequência da palavra correspondente no documento.

Utilizamos o método *'vectorizer.fit()'* para aprender o vocabulário de todos os documentos no conjunto de dados fornecido pela coluna ingredientes_str, este método analisa todos os ingredientes listados em todas as receitas e constroi um vocabulário interno. Desta forma podemos realizar análises ou qualquer outra operação que necessite de uma representação numérica dos dados textuais.

Definimos vetores para os dois datasets.

Em seguida implementamos o *cosine_similarity* , uma função que calcula a similaridade de cosseno entre duas matrizes. A similaridade de cosseno é utilizado para medir a similaridade entre itens ou usuários. A similaridade de cosseno entre esses vetores nos dirá quão semelhantes são os documentos em termos de seu conteúdo textual, independentemente do comprimento dos documentos.

O próximo passo do processo é a função de recomendação, função chamada 'recomendar_receitas_para_compra', que é usada para recomendar receitas com base numa lista de compras fornecida pelo usuário. A função usa processamento de linguagem natural e técnicas de similaridade de cossenos para encontrar receitas que melhor combinem com os itens comprados

O 'Vectorizer.transform' é utilizado para converter as strings da lista em features, utilizada após a .fit(), com a finalidade de utilizar o vocabulário existente e convertê-lo em um vetor.

O método *'similaridades.argsort()*' retorna os índices que ordenam o array de similaridades. A parte [0] seleciona a primeira linha da matriz de similaridades (que contém as similaridades entre a lista de compras e cada receita), [::-1] inverte a

ordem para que os índices com maior similaridade venham primeiro, e [:3] seleciona os três primeiros índices, ou seja, as três receitas mais similares.

A lista de índices (indices_similares) é usada para acessar as receitas no DataFrame receitas_1. Para cada índice em indices_similares, o nome da receita é obtido junto com seu valor de similaridade associado, formando um par (nome da receita, similaridade).

O ‘return recomendacoes’ retorna a lista de recomendações, a função efetivamente usa a lista de compras do usuário para encontrar e recomendar as três receitas mais relevantes com base na similaridade dos ingredientes.

- **Técnica do processo de recomendação**

O algoritmo utiliza a similaridade de cosseno para encontrar itens no dataset receitas que são semelhantes a uma entrada do usuário, uma lista de compras do usuário.

A lista de compras fornecida pelo usuário é transformada em um vetor utilizando o mesmo método usado para vetorizar os dados das receitas, converte a string de texto em uma representação numérica que pode ser comparada com as receitas vetorizadas.

Em seguida calcula a similaridade de cosseno entre o vetor da lista de compras do usuário e todos os vetores de receitas previamente vetorizados e armazenados.O resultado é uma matriz de similaridades que quantifica o quão semelhante cada receita é em relação à lista de compras do usuário. retorna os índices que ordenaram a matriz de similaridades.

Neste resultado, o índice do maior valor, ou seja, da receita mais similar, aparece por último, portanto precisamos inverter essa ordem para ter os índices das maiores similaridades primeiro, em seguida, limita a seleção aos três índices de maior similaridade, correspondendo às três receitas mais similares à lista de compras.

Teremos, então, uma lista das três principais receitas recomendadas com base na lista de compras do usuário, juntamente com as pontuações de similaridade que indicam quão relevantes são essas receitas para a lista de compras fornecida.

A função aproveita a representação vetorial de textos e a medida de similaridade de cosseno para recomendar receitas que utilizam ingredientes semelhantes aos presentes na lista de compras do usuário, potencializando uma experiência

personalizada e relevante de recomendação. Em seguida, o comando ‘print’ mostra ao usuário as receitas recomendadas.

Assim, construiremos o nosso sistema, fazendo com que o algoritmo consiga identificar e recomendar conteúdos relevantes com base nas preferências do usuário ou no conteúdo que eles compraram.

Ex:

```
Recomendações para a lista de compras 2:
- Danoninho Caseiro
- Bobó de Camarão
- Moqueca de Peixe


Recomendações para a lista de compras 3:
- Danoninho Caseiro
- Massa de Panqueca
- Nhoque à bolonhesa
```

4.1 Diagrama de Atividades

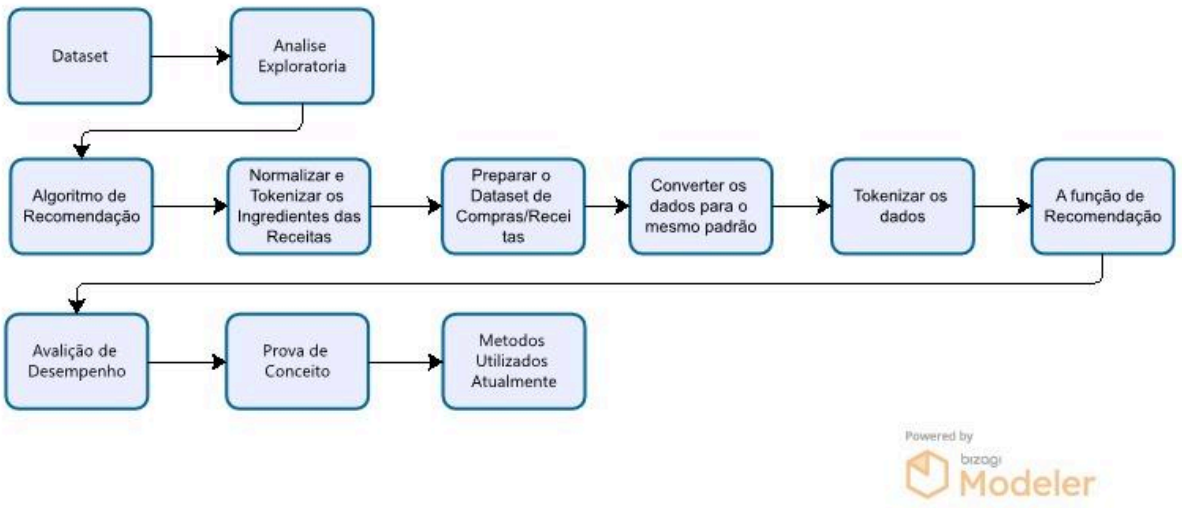


Imagem criada pelos autores

4.2 Avaliação de Desempenho

Alguns métodos de avaliação de desempenho como Acurácia, F1-Score e outras mais utilizadas não podem ser aplicadas neste algoritmo devido a especificidade dos dados.

Poderíamos verificar falsos positivos e falsos negativos no sistema de recomendação, porém seria necessário uma "verdade de base" (ground truth) que definisse quais são as recomendações corretas, precisaria saber antecipadamente quais receitas seriam consideradas como recomendações verdadeiramente relevantes para cada lista de compras. Uma vez que não temos este dado, este tipo de avaliação não será possível.

Esses conceitos são comumente aplicados em tarefas de classificação. No contexto de sistemas de recomendação, a implementação dessas métricas pode ser um pouco diferente, pois sistemas de recomendação geralmente não oferecem uma não-recomendação explícita, o que significa que o conceito de verdadeiros negativos é bem improvável de definir e calcular.

Outras métricas e métodos que você pode considerar para avaliar o sistema indiretamente ou melhorar a qualidade das recomendações são:

- *Cobertura de Catálogo*, que avalia a percentagem do catálogo de receitas que é efetivamente coberto pelas recomendações ao longo do tempo. Se apenas um pequeno subconjunto de receitas é sempre recomendado, você pode estar perdendo a diversidade nas recomendações. Esta medida foi coberta pela aplicação do ‘threshold’ que deu um percentual de receitas recomendadas.
- Diversidade, que mede quão diferentes são as receitas recomendadas entre si. Diversidade alta significa que o sistema está sugerindo uma ampla variedade de receitas, o que pode ser positivo para manter o interesse do usuário, porém como não temos variedades de tipos para uma mesma receita, esta medida não se aplica.
- *Novidade*, que mede a tendência do sistema em recomendar itens menos conhecidos ou populares. Uma alta pontuação em novidade significa que o sistema está recomendando itens que o usuário não está propenso a já conhecer. Porém como não temos um ‘ranking’, esta forma de avaliação não se aplica aos dados.
- *Serenidade*, que mede a capacidade do sistema de recomendação de sugerir itens que o usuário ainda não considerou ou descobriu, oferecendo surpresas agradáveis, que não se aplicam aos dados.

- *Satisfação do Usuário*, que gera avaliações subjetivas coletadas diretamente dos usuários podem fornecer insights valiosos sobre a eficácia do sistema de recomendação. Isso pode ser feito por meio de pesquisas ou análise de feedback porém também não se aplica aos dados.

4.3 Prova de Conceito

Obtivemos sucesso com o algoritmo de recomendação, ele realizou a recomendação de receitas para as 1000 entradas.

Devido à falta de uma coluna de avaliação na base de dados, os algoritmos de recomendação que trabalham com separação de dados entre treino e teste não puderam ser testados nem aplicados, pois os algoritmos que utilizam este tipo de abordagem precisam de um item que eles chamam de *'avaliação da recomendações'* ou da *'nota de relevância'* da receita que seria dada pelo usuário.

Como não temos estes dados, foi empregado o método de funções, uma para definir as recomendações através da similaridade do cosseno e outra para escolher os 3 resultados mais relevantes e para imprimir os resultados para o usuário.

Os dados obtidos pelo algoritmo estão na cessão de conclusão.

4.4 Métodos utilizados atualmente

Nos últimos anos, houve um avanço significativo na aplicação de algoritmos de recomendação para uma variedade de domínios, incluindo a recomendação personalizada de receitas culinárias.

Esses algoritmos visam aprimorar a experiência do usuário ao sugerir receitas que não apenas correspondam às suas preferências individuais, mas também promovam o uso eficiente dos ingredientes disponíveis, contribuindo para a redução do desperdício de alimentos. Os algoritmos de recomendação funcionam para o oferecimento de sugestões com base em algum critério e nos gostos dos usuários.

Os sistemas de recomendação prometem ajudar o usuário a administrar o problema de excesso de informação, recomendando-o itens que correspondam, ao mesmo tempo, à exaustão e à precisão de sua busca, ou seja, aos critérios pessoais de relevância que incluem um gosto prévio(Santini,2020).

Utilizados para personalizar a experiência do usuário e para aumentar as conversões, eles também são utilizados em automação de marketing e atendimento personalizado ao consumidor. Pode-se recomendar de tudo, ofertas, lançamentos, produtos mais comprados, conteúdos mais assistidos, notícias mais lidas, I.As mais utilizadas, qualquer coisa.

Alguns exemplos de utilização em grande escala são as empresas Spotify, Netflix, Amazon e TikTok.

O Spotify baseado no que é buscado, compartilhado e pulado pelo usuário, o Spotify aprende sobre as preferências de cada usuário e passa a recomendar conteúdos. O site cria uma lista chamada “No Repeat” baseada nas músicas mais ouvidas pelo usuário e o “Daily Mix” que mistura as músicas mais ouvidas com músicas recomendadas(mais ouvidas por outras pessoas) de conteúdo similar.

A Netflix utiliza *machine learning* para aprender o comportamento e as preferências de seus usuários e recomenda conteúdo personalizado, com uma tela inicial totalmente personalizada e diferenciada para cada cliente. Além disso, traz o top 10, com os mais assistidos. Método reproduzido por várias empresas de streaming que vieram depois como Disney+, Globo play, Prime e afins.

A Amazon já utiliza um sistema de filtragem híbrida, que gera uma personalização mais apurada, com abas ‘recomendados para você’ além dos mais vendidos e de ofertas.

Outra indicação deste tipo de filtragem é a parte na tela onde aparece “frequentemente comprados juntos” que utiliza o modelo colaborativo que utiliza a informação dos itens que outros usuários compraram juntos quando compraram um determinado que o usuário atual colocou no carrinho.

O Tiktok utiliza o mesmo método de recomendação que o Twitter e o Instagram, que sugerem conteúdos personalizados em uma aba “For You” - Tiktok, onde mostra recomendações baseadas em curtidas, iterações e localização dos usuários.

Muitas empresas atualmente utilizam algoritmos de recomendação e geram lucro e fidelização através desta tecnologia.

5. Biblioteca

No desenvolvimento de algoritmos de recomendação, o tratamento dos dados é um passo fundamental para assegurar a precisão das sugestões geradas. Neste processo, empregamos várias bibliotecas e ferramentas especializadas em análise e manipulação de dados.

A biblioteca Pandas é usada para realizar a manipulação e análise de dados em Python de maneira eficiente. Ela nos permite realizar uma série de operações essenciais, como a limpeza de dados, seleção de colunas específicas, e agregação de informações, sendo instrumental no pré-processamento de dados.

NumPy -> Oferece suporte na manipulação de arrays e matrizes multidimensionais, além de prover um vasto conjunto de operações matemáticas. Isso é particularmente útil para realizar cálculos numéricos complexos e operações de alta performance, fundamentais no tratamento eficaz dos dados.

Matplotlib -> Para a visualização de dados, utilizamos a biblioteca Matplotlib para gerar gráficos, como demonstrado no código de distribuição dos valores de similaridade de cosseno. Isso nos permite visualizar a distribuição e entender melhor os resultados do nosso algoritmo de recomendação.

Pyplot -> É um módulo do pacote Matplotlib, que fornece uma interface que imita a do MATLAB, facilitando a criação de visualizações gráficas como gráficos de linhas, barras, histogramas, nuvens de pontos, de maneira simples, amplamente utilizada para visualização de dados e análise exploratória.

Seaborn -> Trabalha em conjunto com *Matplotlib*, poderia ser igualmente utilizado para melhorar a estética e a compreensão dos gráficos gerados.

Scikit-Learn -> O uso do `*CountVectorizer*` e da função `*cosine_similarity*`, da biblioteca Scikit-Learn, oferece ferramentas para pré-processamento de dados, aprendizado de máquina, e métricas de avaliação, como a similaridade de cosseno, essencial para calcular a proximidade entre os vetores de itens e recomendar os mais relevantes.

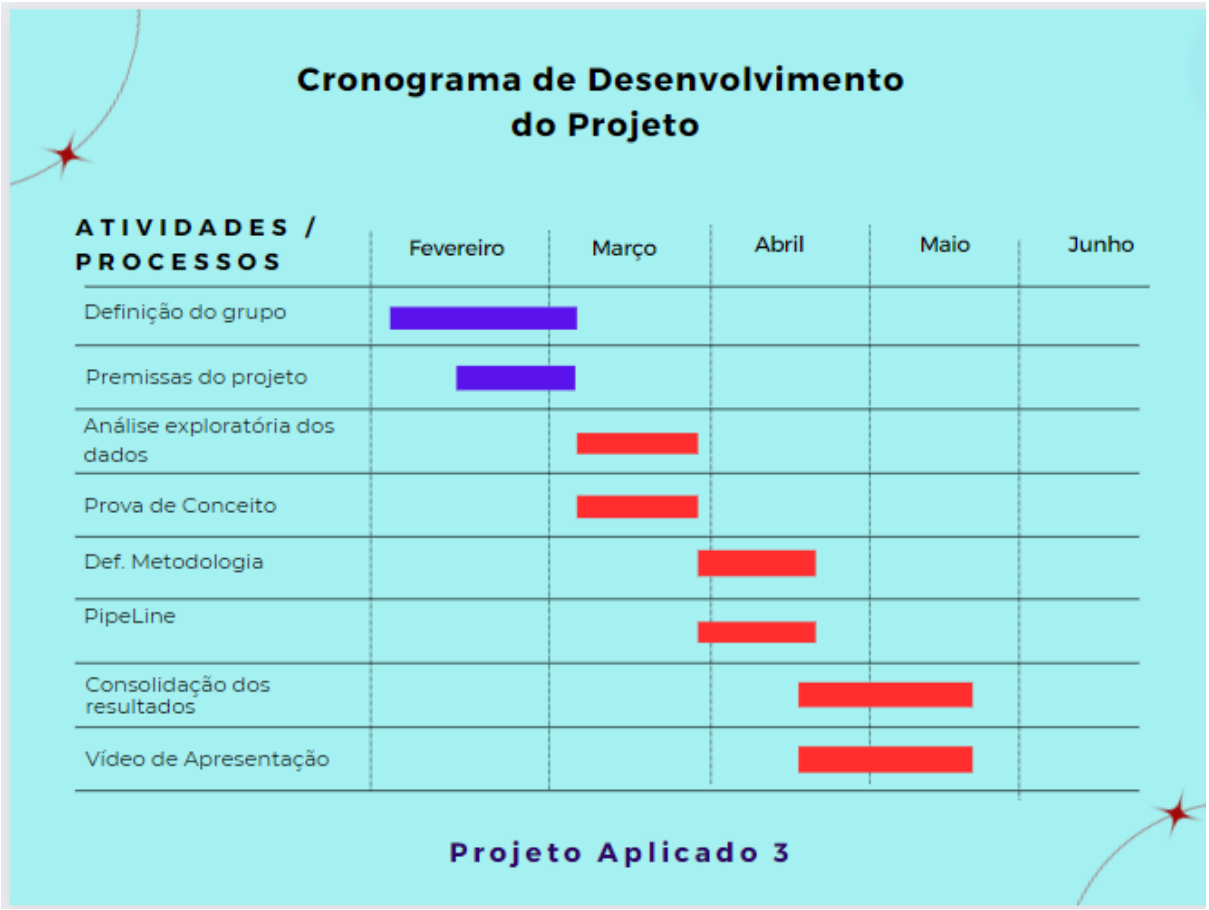
O uso de um limiar de aceitação `threshold` para filtrar recomendações com base na similaridade de cosseno, ressaltamos a flexibilidade e a robustez do NumPy em operações condicionais e matemáticas.

Métodos matemáticos como a Similaridade do cosseno utilizada para medir o cosseno do ângulo entre dois vetores. É útil quando a magnitude dos vetores não é relevante.

Métricas como *Average Recall@k* que mede a porcentagem de relevantes encontrados entre as top K recomendações em relação ao total de relevantes possíveis, *Average Precision@k* que mede a porcentagem de recomendações corretas entre as top K recomendações e *Cobertura de catálogo* utilizada para medir a diversidade das recomendações em termos da porcentagem do catálogo total que é coberto pelas recomendações durante um determinado período ou em um determinado conjunto de interações.

Essas operações são vitais para refinar o processo de recomendação e garantir a relevância das sugestões oferecidas aos usuários. A combinação dessas bibliotecas forma a espinha dorsal do nosso algoritmo de recomendação, possibilitando desde o tratamento e análise dos dados até a implementação de métodos de machine learning e a visualização de resultados, assegurando assim recomendações precisas e de alta qualidade.

6. Cronograma das Atividades



7. Resultados

Obtivemos dois tipos de resultados, antes de fixar um threshold e um depois de fixar um threshold.

Seguem os resultado antes de fixar um threshold de aceitação:

Calculando os percentuais das recomendações baseadas nas seguintes faixas de similaridade de cosseno, abaixo de 0.5, Igual ou acima de 0.5.

Percentual de recomendações com similaridade abaixo de 0.5: 100.00%
Percentual de recomendações com similaridade igual ou acima de 0.5: 0.00%

Um resultado geral, obtido sem a definição de uma métrica de aceitação, em uma análise manual e visual podemos verificar que as recomendações estão corretas e que as receitas realmente tem em sua lista de ingredientes os mesmo ingredientes das compras de referência:

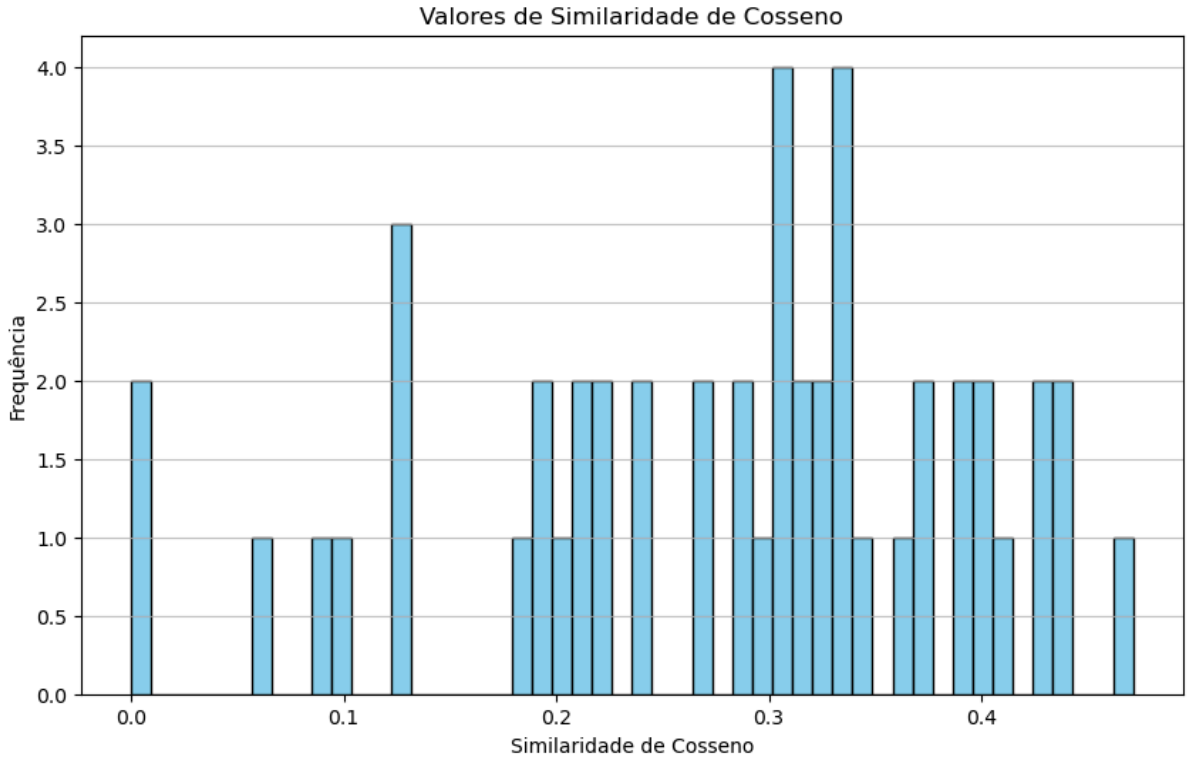
```
Recomendações para a lista de compras 1:
- Nhoque à bolonhesa
- Massa de Pizza
- Strogonoff de Carne


Recomendações para a lista de compras 2:
- Danoninho Caseiro
- Bobó de Camarão
- Moqueca de Peixe


Recomendações para a lista de compras 3:
- Danoninho Caseiro
- Massa de Panqueca
- Nhoque à bolonhesa
```

O sistema de recomendação é um algoritmo de filtragem baseada em conteúdo, ele utiliza vetores de características para que cada compra e cada receita seja representada por um vetor de características. No caso das receitas, essas características são os ingredientes. Para as compras, são os itens adquiridos.

A similaridade entre os vetores de características das compras e das receitas é calculada usando a similaridade de cosseno, um método comum para determinar o quão semelhantes dois vetores são, sendo particularmente útil quando a magnitude dos vetores não é relevante. Os valores de similaridade de cosseno geralmente não são negativos, especialmente no contexto da recomendação de receitas, onde as

similaridades são calculadas a partir de características como a presença ou ausência de ingredientes (geralmente representadas por vetores não negativos).

Gráfico:



Dados gerados pelos autores

O gráfico de barras indica que os valores de similaridade de cosseno estão agrupados em intervalos distintos, o que é comum em sistemas de recomendação quando existem padrões específicos ou limitados de combinações de ingredientes.

Os resultado após fixar um threshold de aceitação:

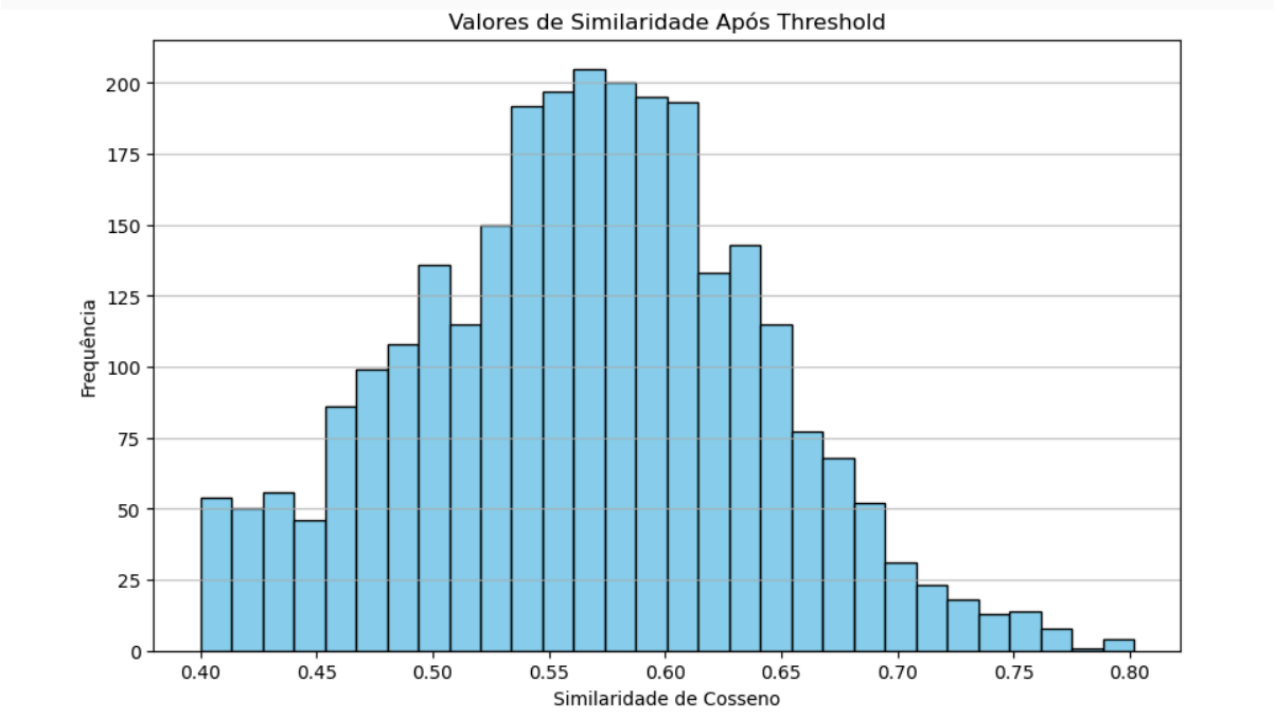
Após definir um limiar de aceitação para melhoria da relevância das receitas recomendadas, em 0.4 que representa 40% de relevância, obtivemos resultados diferentes:

```
Recomendações para a lista de compras 1:
- Nhoque à bolonhesa  (Similaridade: 0.47140452079103173)
- Massa de Pizza (Similaridade: 0.44194173824159216)
- Strogonoff de Carne (Similaridade: 0.44194173824159216)


Recomendações para a lista de compras 2:
- Danoninho Caseiro (Similaridade: 0.6481812160876688)
- Bobó de Camarão (Similaridade: 0.6000991981489792)
- Moqueca de Peixe (Similaridade: 0.5940885257860046)


Recomendações para a lista de compras 3:
- Danoninho Caseiro (Similaridade: 0.6316139407998893)
- Massa de Panqueca (Similaridade: 0.5954913341754138)
- Nhoque à bolonhesa  (Similaridade: 0.5844720395466498)
```

Dados gerados pelos autores

Além das Receitas, o código mostra também o percentual de similaridade que representa a relevância da receita em relação aos itens comprados.

Gráfico:



Dados gerados pelos autores

Podemos visualizar uma distribuição inclinada (Distribuição Gaussiana) para a direita, com a maioria das pontuações de similaridade agrupadas entre 0.55 e 0.65. Isso sugere que o sistema está encontrando um nível moderado de similaridade entre as receitas em relação à lista de compras do usuário, o que é bom para a variedade.

A frequência de pontuações acima do limite é alta, o que significa que o sistema é bastante flexível no que considera similar. No entanto, sem contexto sobre a escala de pontuações ou a granularidade das diferenças entre as receitas, temos dificuldades em avaliar a eficácia do limite como citado anteriormente, devido a falta de avaliações ou de um ranking de pontuação.

A realização de cálculos de medidas de desempenho - Métricas como a acurácia, F1-Score e recall são baseados na interseção entre as receitas recomendadas e as receitas consideradas relevantes para cada usuário. Em um sistema de recomendação a acurácia é calculada como a proporção de predições corretas (tanto verdadeiros positivos quanto verdadeiros negativos) sobre o total de casos, já a matriz de confusão, por sua vez, mostra verdadeiros positivos, falsos positivos, verdadeiros negativos e falsos negativos. No entanto, esses conceitos podem ser um pouco mais complexos devido à natureza das recomendações.

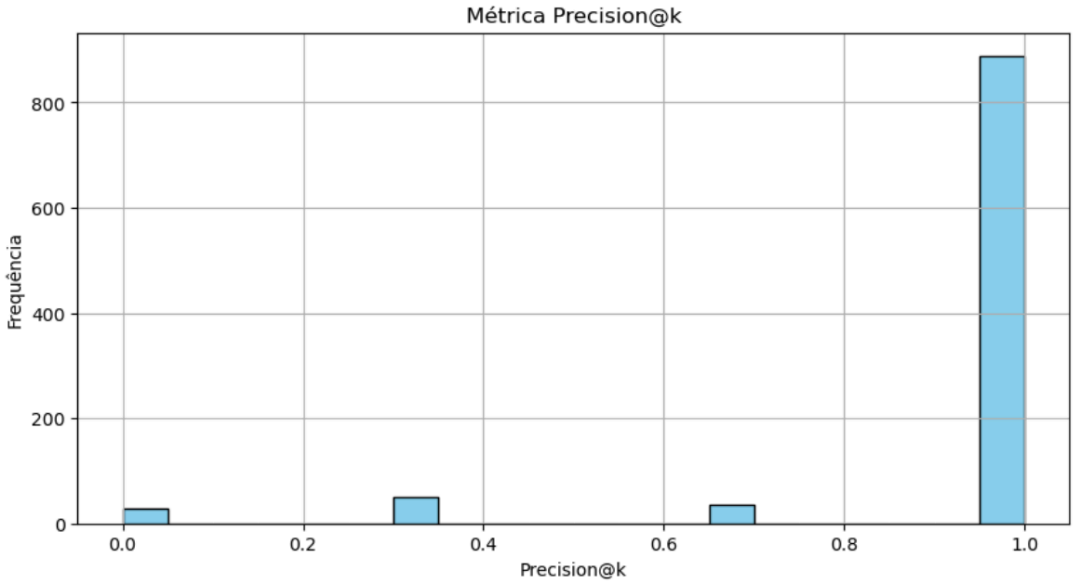
Aqui, consideramos uma recomendação como positiva se estiver entre as top recomendações e relevante, e negativa se não estiver entre as recomendações ou não for relevante. Neste tipo de sistemas, baseados em conteúdo, cada item não recomendado não é necessariamente um verdadeiro negativo, pois pode simplesmente não ser relevante o suficiente para estar nas top recomendações.

Se considerarmos que para cada compra há múltiplos itens que poderiam ser vistos como relevantes, e esses itens não foram recomendados, o número de falsos negativos pode ser substancialmente maior que o número de compras.

Devido a falta de um rating, utilizaremos métricas mais específicas como Precision@k, Recall@k e a cobertura do catálogo.

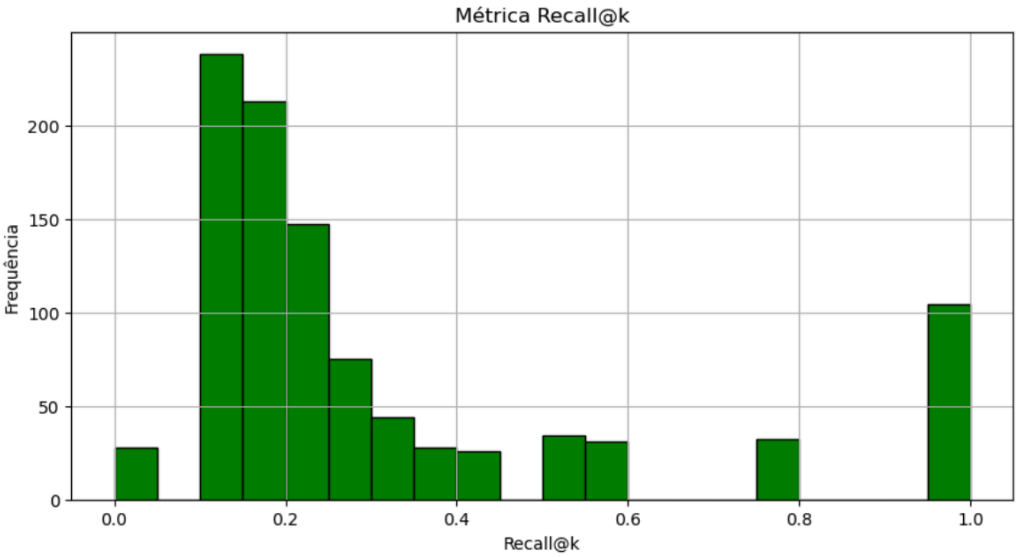
A métrica **Precision@k** é utilizada para avaliar como as receitas recomendadas são relevantes para o usuário. Uma recomendação é relevante se contém pelo menos metade dos ingredientes de compras do usuário.

A Precisão média foi de 92,77%. Um resultado excelente, que indica que a maioria das recomendações que o seu sistema faz nos top-k, assumindo k como 3 - já que o algoritmo recomenda as 3 receitas, são realmente relevantes para os usuários. Isso sugere que o sistema é muito eficaz em identificar e sugerir receitas aos usuários, baseando-se na lista de compras.



A métrica **Recall@k** é utilizada para medir quantos dos ingredientes comprados foram cobertos pelas receitas recomendadas. Calcula o recall das recomendações, ou seja, quão bem as recomendações cobrem os ingredientes que o usuário realmente comprou. Recall é o número de itens relevantes e recomendados dividido pelo total de relevantes.

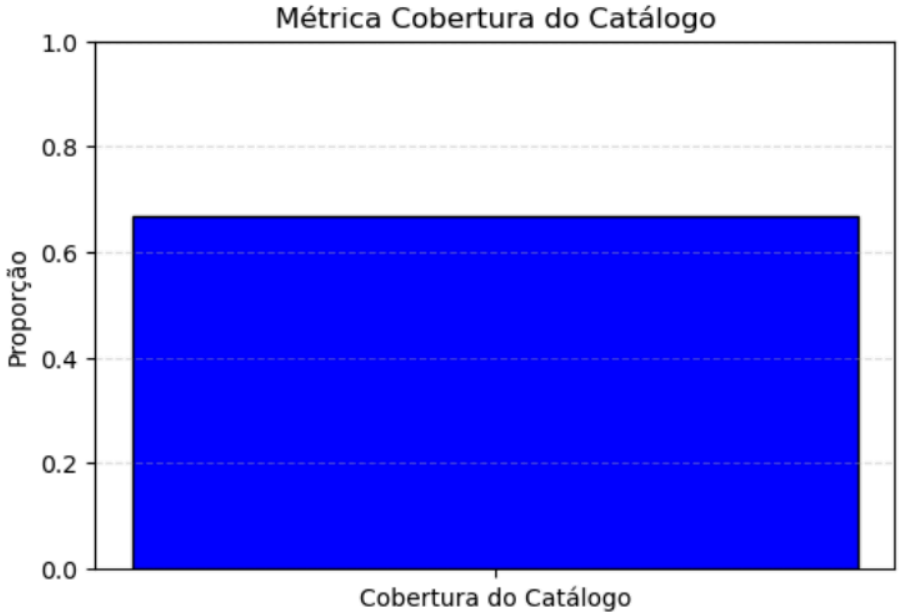
Com 32,13% o recall foi relativamente baixo. Isso significa que, embora as recomendações que seu sistema faz são relevantes, ele está capturando apenas uma pequena fração das receitas que seriam consideradas relevantes pelos usuários, ou seja, muitas receitas potencialmente relevantes não estão sendo recomendadas.



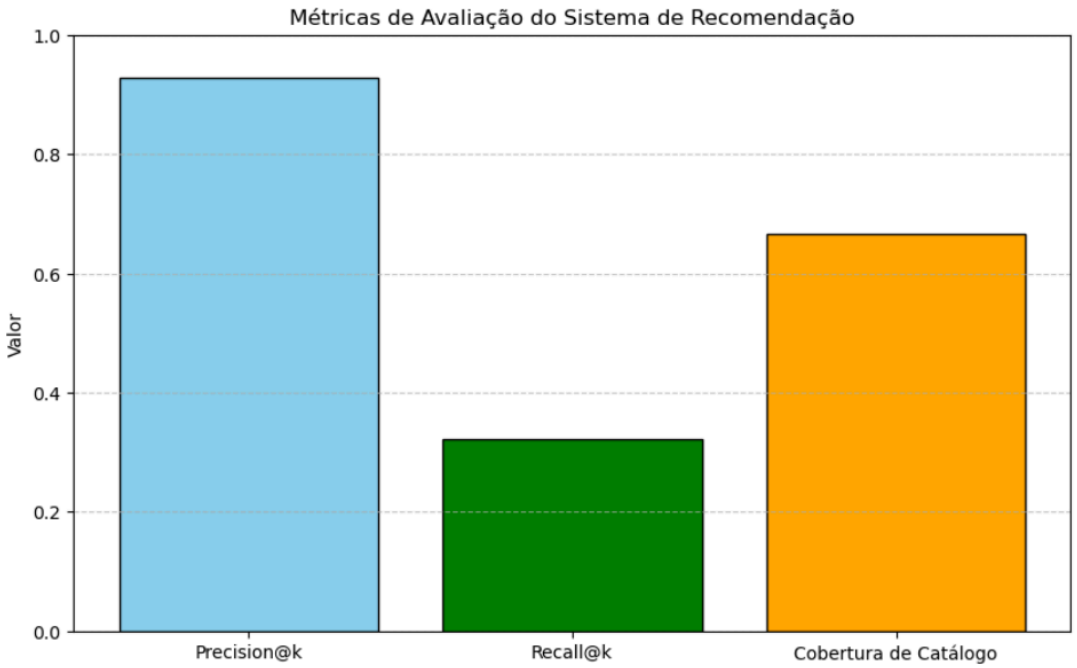
A métrica **Cobertura de catálogo** é utilizada para indicar a porcentagem de receitas diferentes recomendadas pelo sistema em relação ao total de receitas disponíveis

no catálogo. A cobertura do catálogo é o número de itens únicos recomendados dividido pelo tamanho do catálogo.

Com 66,67%, a cobertura do catálogo é relativamente boa, indicando que o sistema explora dois terços do catálogo de receitas disponíveis ao fazer recomendações. Isso é positivo porque sugere que o sistema não está apenas se concentrando em um pequeno conjunto de receitas populares ou frequentes, mas está oferecendo uma gama razoavelmente ampla de opções.



Um gráfico comparativo serve para que tenhamos uma visão geral das métricas e possamos compará-las de forma mais efetiva. Vemos neste gráfico que a precisão está realmente acima da média e a distribuição e varredura das receitas está boa, poderia melhorar, no entanto ainda temos a limitação gerada pela falta de rating.


Métricas de Avaliação do Sistema de Recomendação
Valor
1.0
0.8
0.6
0.4
0.2
0.0
Precision@k
Recall@k
Cobertura de Catálogo

8. Conclusão

Este trabalho teve como objetivo desenvolver um algoritmo de recomendação de receitas a partir dos itens comprados pelos usuários, visando reduzir o desperdício de alimentos e melhorar a nutrição. Neste sentido escolhemos fazer um algoritmo de recomendação de receitas a partir dos itens que o usuário comprou para reduzir o desperdício de comida e melhorar a nutrição das pessoas.

Para desenvolver este algoritmo utilizamos algumas bibliotecas e ferramentas de software livre para preparar os dados, através destas ferramentas o objetivo foi alcançado e o software foi desenvolvido com sucesso, tendo uma assertividade na relevância das recomendações superior a noventa por cento.

Contudo, enfrentamos desafios significativos, como a ausência de uma API para acesso aos dados das receitas e avaliações, o que limitou a utilização de algoritmos de recomendação mais avançados e impactou na avaliação de desempenho do nosso sistema. A falta de um rating nos dados iniciais nos obrigou a adotar métricas menos convencionais para validar a eficácia do sistema.

Identificamos vantagens na filtragem baseada em conteúdo, como a personalização, já que as recomendações são diretamente relacionadas aos dados específicos do usuário, a independência de outros usuários visto que, diferentemente da filtragem colaborativa, a filtragem baseada em conteúdo não requer a comparação com dados de terceiros podendo funcionar de maneira efetiva mesmo com um número limitado de usuários. Este tipo de sistema é bastante útil para cenários onde é possível definir claramente as características dos itens e onde essas características são um bom indicador das preferências do usuário.

No contexto de recomendar receitas com base em compras de ingredientes, este método permite um alinhamento direto entre o que os usuários compram e o que eles provavelmente gostarão de cozinhar. Porém é importante ressaltar que sem um retorno por parte do usuário, ou seja, uma avaliação ou uma indicação de que ele realmente reconhece uma relevância - gostou da receita, trabalhamos apenas com dados matemáticos e com teoria e a alma do sistema de recomendação é criar valor para o usuário.

Apesar de vasta literatura sobre algoritmos de recomendação, percebemos uma lacuna específica na aplicação desses para receitas. Existe um site https://myfridgefood.com que mostra uma lista de ingredientes, o usuário seleciona os itens que tem na geladeira e ele recomenda receitas com base nos dados fornecidos.

Assim como a Netflix utilizamos que utiliza o *top N video ranker e o algoritmo Video-video similarity,* nós utilizamos um top-k para selecionar as 3 receitas mais relevantes e uma função de similaridade para otimizar a recomendação. Isso nos levou a desenvolver uma abordagem inovadora que, embora independente de avaliações de usuários, ainda carece de aprimoramentos para melhorar o equilíbrio entre precisão e recall, e para expandir a cobertura do catálogo de receitas.

Concluímos analisando os resultados que, como melhorias em trabalhos futuros, o sistema precisa de ajustes para melhorar o recall, talvez aumentando o número de recomendações ou diminuindo os critérios de similaridade para inclusão de itens na lista de sugestões. Essa abordagem pode ajudar a equilibrar melhor a precisão e o recall, aumentando a abrangência das recomendações sem comprometer significativamente a relevância das mesmas.

A partir de alterações na base de dados, provavelmente trocando a fonte para um site de receitas que forneça dados via API poderemos implementar algoritmos de recomendação mais sofisticados, como por exemplo, utilizar redes neurais e outros algoritmos de agrupamento de receitas por proximidade ou por nota, tornando a experiência do usuário mais assertiva e solicitar a avaliação do usuário final após fazer a receita para aprender o gosto deste usuário e poder fazer recomendações com um nível de personalização mais apurado.

Ajustes contínuos e a inclusão de feedback direto serão cruciais para otimizar o desempenho do sistema ao longo do tempo.

Referências

Git hub:

https://github.com/VaniaJesus/Proj_Aplicado-3

ALEXANDER, N. Catered to your future self: Netflix's "predictive personalization" and the mathematization of taste. In: MCDONALD, K.; SMITH - ROWSEY, D. (ed.). The Netflix effect:technology and entertainment in the 21st Century. Nova York: Bloomsbury Academic, 2016. p. 81-98.

Algoritmo de recomendação: o que é e como ele contribui para as estratégias personalizadas. meio&mensagem. Disponível em: Algoritmo de recomendação: o que é e como eles funcionam? (meioemensagem.com.br). Acesso em: 20 de Março de 2024.

Cesta básica nacional. Universidade Federal de Mato Grosso do Sul. Disponível em: https://cpcs.ufms.br/custo-cesta-basica-setembro-2023/#:~:text=A%20cesta%20b%C3%A1sica%20de%20alimentos,%2C%20banha%2F%C3%B3leo%20e%20manteiga. Acesso em: 04 de Março de 2024.

Domine a Programação em Python com a Biblioteca Pandas: um Guia Completo. Awari. Disponível em: https://awari.com.br/domine-a-programacao-em-python-com-a-biblioteca-pandas-um-guia-completo/#:~:text=O%20Pandas%20%C3%A9%20uma%20biblioteca%20poderosa%20que%20oferece%20uma%20variedade,partir%20dos%20conjuntos%20de%20dados Acesso em: 08 de Março de 2024.

LANDOW, G. Hipertexto: la convergencia de la teoría crítica contemporánea y la tecnología. Barcelona: Paidós, 1992.

McKinney, W. (2012). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. O'Reilly Media.

O que são Redes Neurais. Ibm. Disponível em:O que são Redes Neurais? | IBM. Acessado em: 08 de Março de 2024

ODS. Nações Unidas Brasil. Disponível em: https://brasil.un.org/pt-br/sdgs. Acessado em: 08 de Março de 2024

Receitas, Site Tudo Gostoso. Disponível em: https://www.tudogostoso.com.br/. Acesso em: 04 de Março de 2024.

VanderPlas, J. (2016). Python Data Science Handbook: Essential Tools for Working with Data. O'Reilly Media.