

---

## SISTEMA DE RECOMENDAÇÃO DE PRATOS GASTRONÔMICOS DE ACORDO COM O PERFIL DO CLIENTE

MOREIRA, Willian Felipe de Almeida<sup>1</sup>

da SILVA, Matheus Luiz<sup>1</sup>

DIAS, Claudinei<sup>2</sup>

POSSAMAI, Diego Sauter<sup>2</sup>

SENHORINHA, Márcio José Kãms<sup>2</sup>

### RESUMO

A crescente demanda de dispositivos tecnológicos gera uma grande quantidade de dados que são manipulados por um grupo do conhecimento nomeado como Big Data, um dos segmentos mais influentes dessa área são os sistemas de recomendação e estão presentes em diferentes situações do ambiente virtual. A fim de criar uma experiência de usuário personalizada utilizando como base os dados coletados do usuário, o comércio eletrônico utiliza esse algoritmos buscando se destacar e se aproximar do usuário criando ambientes de recomendações personalizados. Compreendendo a crescente demanda e o impacto que os algoritmos de recomendações trazem na sociedade atual, suscitando debates políticos e de ética e transformando setores da economia, o presente artigo tem como objetivo discorrer sobre o desenvolvimentos desses algoritmos que utilizam aprendizado de máquina para propiciar uma recomendação com maior acurácia, com isso o leitor obterá pleno entendimento da teoria e prática de desenvolvimento do mesmo.

**Palavras-chave:** Sistema de Recomendações; Aprendizado de máquina; algoritmos

### ABSTRACT

The growing demand for technological devices generates a large amount of data that are manipulated by a group of knowledge named as Big Data, one of the most influential segments of this area are the recommendation systems are present in different situations of the virtual environment in order to create a personalized user experience using as a basis the data collected from the user, e-commerce uses these algorithms seeking to stand out and approach the user by creating an environment of personalized recommendations. Understanding the growing demand and the impact that recommendation algorithms bring to current society, raising political and ethical debates and transforming sectors of the economy, this article aims to discuss the development of these algorithms that use machine learning to provide a recommendation with greater accuracy, so the reader will get a full understanding of the theory and practice of its development.

**Keywords:** Recommendations System; machine learning; algorithms;

<sup>1</sup>Graduando do Curso de Engenharia da Computação do Centro Universitário UNISOCIESC{felipealmeida@hotmail.com, luizmatheus2342@gmail.com}

<sup>2</sup> Banca examinadora: {claudinei.dias, diego.sauter, marcio.senhorinha}@unisociesc.com.br

## 1 INTRODUÇÃO

O mundo da recomendação é vasto e tem muitas áreas de atuação. Além de ajudar a pessoa a efetuar a escolha por um produto/serviço, pode gerar muito lucro para as empresas que utilizam o sistema de recomendação.

Este projeto tem como objetivo mostrar as ferramentas e metodologias utilizadas para a construção de um sistema de recomendações apto a fazer uma recomendação personalizada de pratos para o usuário com base nos gostos pessoais do usuário, ou seja, com base nos pratos que o usuário costuma consumir e na avaliação empregada aos pratos. O estudo de caso deste projeto tem como base a gastronomia, e como o gosto pessoal é subjetivo, pode ser que às vezes não faça uma recomendação que irá agradar ao usuário. Porém, faz o possível para que tenha uma predição correta baseada nos métodos já conhecidos de filtragem colaborativa e recomendação através de algoritmos especializados em clusterização de um problema de classificação não supervisionada.

Este documento está estruturado da seguinte maneira: primeiramente serão visitadas as áreas de *machine learning* e análise de dados para que se possa entender como funciona esse mundo da extração das características de um dado objeto que se deseja classificar, após será mostrado os métodos para fazer a recomendação desses dados classificados. Por fim será apresentado o desenvolvimento na linguagem python do sistema de recomendações de pratos gastronômicos.

## 2 CONCEITOS FUNDAMENTAIS

Neste capítulo será abordado como funciona a busca pelas características importantes a serem inseridas para a classificação de um objeto, como tratar as informações obtidas e como usar as características selecionadas para que através do aprendizado de máquina efetue uma recomendação de forma eficiente.

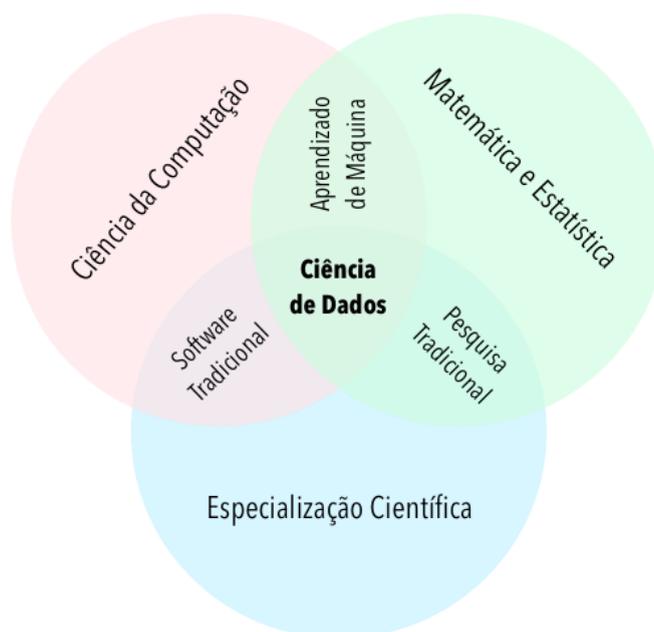
## 2.1 CIÊNCIA DE DADOS

“Ciência de dados é um conjunto de princípios fundamentais que norteiam a extração de conhecimento a partir dos dados” (PROVOST; FAWCET, 2016). Os dados sozinhos não representam informação alguma, para que os dados sirvam para algum propósito de informação, deve-se ter a definição para o que os dados serão utilizados e quais características devem ser levadas em conta para o estudo de caso em específico. Deve-se pensar em dados de forma analítica, identificando dados apropriados e escolhendo métodos adequados. Deve-se formular soluções de busca de dados e avaliar os resultados obtidos e pensar cuidadosamente sobre o contexto em que serão utilizados.

Em uma missão durante a segunda guerra mundial, o matemático Abraham Wald teve que analisar a melhor forma para que se construísse uma blindagem para os aviões do seu país, sendo que devido ao peso da blindagem não poderia construir um avião completamente blindado, pois ele não manteria voo. No primeiro momento Abraham notou que a perfuração não era comum nos motores. O mais leigo pensaria que não deveria ser um lugar para fazer a blindagem, mas o que ele percebeu é que os aviões que receberam tiros no motor, nem mesmo voltaram para contar história. Sendo assim não fazendo parte do estudo. Mas, sendo um dado de extrema importância para o planejamento da blindagem dos aviões (MANGEL; MARC, 1984).

Esse caso mostra que não apenas os dados importam, mas um conhecimento profundo do domínio do problema a ser analisado para que se possa tirar dos dados informações mais precisas e que seja possível fazer a previsão do futuro com base nos dados. Há certas características do objeto que se deseja classificar, que dependendo do escopo do objetivo que se deseja alcançar com as características desse objeto alvo não fazem diferença para a pesquisa ou podem gerar inconsistência nos resultados. Como o fator 'g' de inteligência em relação ao sexo da pessoa.

Como representado no diagrama de Veen da data Science (Figura 1). A ciência de dados é muito abrangente e complexa. O assunto está em alta por proporcionar diversas formas de alavancar negócios com estudos de dados .

**Figura 01:** Diagrama de Venn da data science

**Fonte:** Baudisch(2016)

Conforme exibido na Figura 1, a ciência de dados permeia as mais diferentes áreas do conhecimento, exigindo assim uma grande gama de conhecimentos diversificados, grandes principais áreas que permeiam esse campo de estudo podem ser classificadas da seguinte forma:

- a) Ciência da computação – Área em que os conceitos e fórmulas são transcritos em algoritmos para serem utilizados em uma linguagem de programação para que a aplicação possa ser desenvolvida com o objetivo de conquistar os resultados desejados.
- b) Matemática e Estatística – Área de estudos que contribuem com a combinação de dados.
- c) Especialização científica – Essa parte trata de como ocorrerá a busca e a organização dos dados.

Quando se estuda os dados para tomar uma decisão objetiva, deve-se não só entender o domínio do problema apresentado, mas também escolher as características adequadas que serão informativas para o processo de treinamento e descoberta do objeto alvo de estudo.

Para que seja possível trabalhar com as características selecionadas, recomenda-se usar a modelagem preditiva, que utiliza da estatística e de modelos matemáticos com a segmentação supervisionada que será a parte onde ocorre a seleção das melhores características para que se faça uma boa comparação e recomendação (PROVOST; FAWCET, 2016).

A descoberta dos atributos, ou seja, o vetor de características dos objetos que serão comparados para a recomendação é adquirido por meio de um modelo composto de uma fórmula matemática e ou uma especificidade lógica. A partir de uma linha de base, é então realizada a busca pelos registros que se deseja fazer as comparações. A linha de base que terá as características principais é a característica alvo. A característica alvo será a mais específica para que se possa fazer uma recomendação mais aproximada possível. No começo da busca das informações, já deve ter um filtro específico para que sejam separadas as informações não úteis ao estudo de caso (PROVOST; FAWCET, 2016).

## 2.2 MACHINE LEARNING

“Machine learning é o campo de estudo que dá aos computadores a capacidade de aprender sem serem explicitamente programados” (SAMUEL, 1959), essa definição clássica simplifica que o algoritmo toma decisões que com base em dados coletados do usuário ou previamente alimentados sem precisar programar especificamente cada uma dessas decisões. O aprendizado de máquina se integrou no dia a dia de boa parte da população mundial, estando presente em algoritmos de recomendação, filtros de Spam, Veículos autônomos, indústria etc (TERESA, 2021).

### 2.2.1 Aprendizado Supervisionado

Nesse modelo, o algoritmo usa os dados como treinamento para identificar padrões, onde já há vetores de características que identificam o alvo a ser encontrado. Basta que o dado analisado seja comparado com os dados que já possuem as

características certas e fazer o treinamento e análise para determinar qual a percentagem que o dado é similar às características já definidas do estudo.

As técnicas mais utilizadas no aprendizado supervisionado são: *Artificial neural networks*, *Decision tree*, *Vetorial support*, *K-nearest neighbors* e *Naive bayes*. O modelo de aprendizagem de máquina supervisionado, portanto, é a área no qual existem as mais diversas aplicações bem sucedidas e a maioria dos problemas já estão definidos (HONDA; FACURE; YAOHAO, 2017).

### **2.2.2 Aprendizado não Supervisionado**

É o modelo mais utilizado quando o objeto da análise não possui uma categorização definida. O modelo faz uma classificação própria de forma automatizada do dado com base na similaridade entre os dados como quantas vezes esses dados se formam em uma base do usuário em conjunto ou até mesmo na similaridade entre as características dos dados.

A depender do domínio do problema é extremamente difícil e custoso conseguir uma relação eficiente entre os dados. A começar que a base de dados deve já ter uma boa quantidade de dados para que faça uma classificação por semelhança adequada. Por vezes, é difícil conseguir dados suficientes para que se possa fazer uma comparação adequada de similaridade para então fazer a classificação (HONDA; FACURE; YAOHAO, 2017).

## **2.3 SISTEMA DE RECOMENDAÇÃO**

Sistemas de recomendações são sistemas capazes de realizar a recomendação de produtos/serviços ao usuário. Os sistemas de recomendações já são utilizados por muitos serviços atualmente, seja para recomendar um filme na NETFLIX, seja para recomendar uma playlist SPOTIFY, ou até mesmo produtos como no caso do AMAZON.

O sistema de recomendações vem dando bons frutos às empresas que o utilizam desde muitos anos. Já em 2013 o sistema de recomendações já era responsável por 33% do total de vendas feitas pelo e-commerce da Amazon (MAYE;SCHONBERGER, KENNETH, 2013). Outro caso bem sucedido do sistema de recomendações é o usado pela Netflix. De acordo com McKinsey, 75% dos usuários do Netflix vieram através da recomendação de filmes (CASTROUNIS, 2019).

Para que seja possível fazer a recomendação, além dos dados disponíveis para a devida comparação, deve ter um filtro empregado no sistema de recomendação para que possa indicar de que forma se dará a seleção dos dados a serem recomendados para o usuário. Esse filtro deve ser implementado ao objeto de análise com base no que se deseja atingir como resultado e qual será o melhor filtro para determinado domínio de problema.

O tráfego online gerado por um usuário, é constantemente monitorado pelos mais diversos sites para que possam vender um produto/serviço ou manter o usuário preso no site gerando visualização. Independente se o usuário do site comprar um produto ou apenas olhar para o produto, as informações do tempo que ele observou um produto já geraram um armazenamento de dados referente às escolhas do usuário.

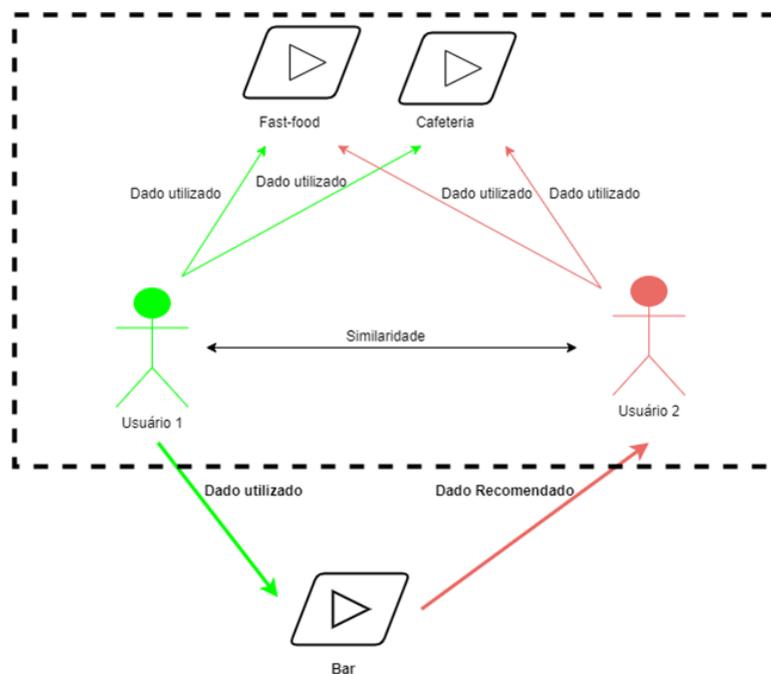
Apesar do histórico de compras de produtos efetuadas em ambiente virtual ser uma informação muito útil na hora de fazer a recomendação de produtos ao cliente, a visualização dos produtos pelo cliente e o tempo dessa visualização são de extrema importância para que o algoritmo faça o uso desses dados para que seja feita uma recomendação de produtos similares ao cliente (CUNHA; SIMÕES, 2018).

Outra forma de recomendar um produto para o usuário do site é com relação às páginas, grupos e comunidades que ele frequenta. Em um cenário em que um usuário de um site gosta muito de pets em geral. Caso ele frequente essas páginas, é muito provável que apareça na tela dele alguma propaganda de produtos para pets, quer seja por feed da página, quer seja por banner mostrado em alguma parte da página navegada.

### 2.3.1 Recomendação Baseada em Filtro Colaborativo

O tipo de recomendação por filtragem colaborativa é uma técnica que prioriza os grupos de usuários. Uma técnica que faz a comparação de todos os dados dos usuários da rede e faz a recomendação baseada em similaridade de dados entre os usuários conforme exemplificado na figura 02, estabelecendo como recomendação o dado que os faz diferentes entre si e que mostra-se eficaz para uma alta gama de aplicações (HERLOCKER et al. 2004).

**Figura 02:** Filtro colaborativo



**Fonte:** Autoria própria. (2021)

Para fazer a recomendação, o filtro colaborativo busca por usuários com dados similares e a partir desses dados faz a recomendação de itens ainda não consumidos pelo usuário que está recebendo a recomendação cruzada com os gostos de outro usuário com perfil parecido, com isso se evita que o usuário fique dentro de uma bolha de conteúdos, mas tem que ser medido para não diversificar demais o conteúdo e o usuário não se identificar com o que foi apresentado.

### 2.3.2 Recomendação Baseada em Conteúdo

Esse tipo de algoritmo recomenda novos conteúdos similares comparando com outros que o usuário já interagiu, cruzando os metadados contidos nos itens interagidos com outros similares. Com isso é possível medir a preferência do usuário e criar um sistema de recomendação baseado em seu gosto pessoal que é constantemente atualizado conforme a sua utilização. Para armazenar essas preferências é necessário criar um perfil de usuário armazenando todas essas interações, tanto as implícitas que são interações indiretas com o conteúdo (tempo de utilização, cliques etc.) quanto as explícitas que são interações diretas (curtidas, favoritos, etc). (D'ADDIO, 2015).

No geral, um objeto de análise possui em suas características diversos pesos para o sistema de recomendação poder fazer a comparação mais adequada e poder fazer a recomendação. Os objetos com as características mais parecidas, levando em conta o peso de cada característica são selecionados (LIMA;PIMENTA - 2002).

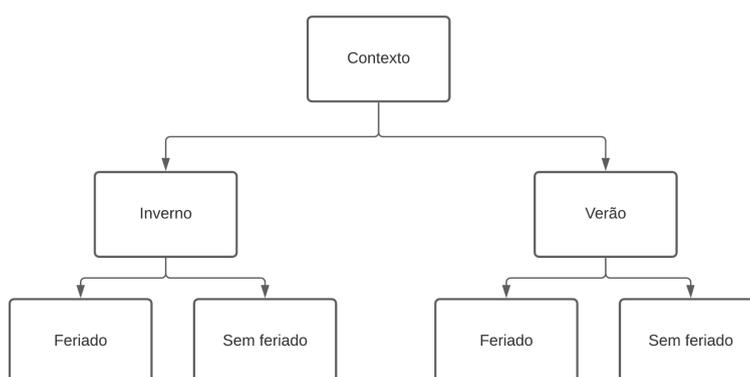
Os metadados são as características presentes que descrevem um item, como a sua descrição, tag, categoria, localização, preço, ingredientes, número de curtidas do usuário em determinados assuntos, etc. É parte fundamental do algoritmo de recomendação, pois uma métrica de dados a ser analisada pode comprometer totalmente o resultado.

Conforme D'Addio (2015), os metadados podem ser classificados em três tipos: estruturados onde os termos estão previamente categorizados como: Gênero, ano, preço, duração, etc. Os não estruturados onde os termos que não possuem uma padronização clara como: textos, resumos, descrições de itens, etc. O terceiro elemento é uma mistura dos dois pois mistura a categorização com a descrição como por exemplo: cabeçalhos, palavras-chaves e títulos.

### 2.3.4 Recomendação Sensível ao Contexto

A maioria dos algoritmos de recomendações atuais se concentram em fornecer informações somente se baseando na relação dos usuários com os conteúdos e acabam deixando informações contextuais de fora dessa equação. Esse tipo de dado leva em conta vários parâmetros referentes ao contexto em que o usuário está inserido, por exemplo pode recomendar água de coco caso esteja em uma tarde de verão na praia (ADOMAVICIUS;TUZHILIN, 2010). Esse método acrescenta uma nova variável de contexto, além dos usuários e de conteúdo, pode ser usada em conjunto com qualquer outro algoritmo de recomendação conforme representado na figura 03.

**Figura 03:** Exemplo de hierarquia com contexto



**Fonte:** Autoria própria (2021)

A pré-filtragem contextual é uma abordagem de incorporação de texto, onde utiliza os dados que se relacionam com o contexto para fazer um filtro nos dados relevantes e excluindo tudo que não esteja no escopo definido. Utilizando a maneira “bruta” dessa técnica pode ocorrer o excesso de especialização do conteúdo, onde a entrada de contexto utilizada limita muito a saída de resultado. Para resolver esse problema, Adomavicius e Tuzhilin (2005) propuseram uma abordagem de redução aproximada ou generalização de texto.

A Generalização de texto consiste em melhorar a pré-filtragem agrupando determinadas palavras em termos mais gerais, solucionando também outro problema que pode ocorrer, onde não existem dados suficientes baseados na palavra chave do contexto. Por exemplo: em uma situação onde o usuário está em um sábado a noite com

amigos e quer assistir filmes de karate, formando a seguinte entrada: R (noite, amigos, karate), nesse caso tem grandes chances de cair nos dois problemas descritos (especialização e dados insuficientes), com isso pode fazer uma generalização: R' (noite, acompanhado, artes marciais) (ADOMAVICIUS; TUZHILIN, 2010).

## 2.4 SACO DE PALAVRAS

É uma maneira de extrair os dados textuais, medindo a frequência do termo e agrupando todos os metadados contidos em cada documento individualmente, de modo que não leva toda a sua ordem e estrutura passando a considerar como palavras isoladas fazendo assim o processo de tokenização ou vetorização (FALK, 2019). A figura 04 apresenta a matriz com as frequência das palavras do dois exemplos a seguir:

João gosta de viajar..

Maria não gosta de viajar e não gosta de avião.

**Figura 04:** Exemplo de matriz com saco de palavras

Documento	João	vai	viajar	Maria	gosta	não	avião	de	e
João gosta de viajar	1	0	1	0	1	0	0	1	0
Maria não gosta de viajar	0	0	1	1	2	2	1	1	1

**Fonte:** Autoria própria (2021)

Na parte de vetorização de cada item é que ele mostra a sua utilidade, medindo a frequência que uma palavra aparece em cada documento, transformando elas em binário. Ao criar o vetor é possível medir o comportamento do usuário dentro da aplicação e posteriormente classificá-los utilizando um algoritmo de machine learning.

## 2.5 PRÉ-PROCESSAMENTO

Para que o algoritmo consiga classificar os dados têm que ser previamente tratados e melhorados, pois cada item interagido pelo usuário pode conter uma quantidade muito grande de informações que são dispersas entre si. O saco de palavras

não filtra esse problema e agrupa as palavras sem se importar com a relevância, com isso essa etapa tem como finalidade criar uma estrutura padronizada que possa depois ser utilizada como treinamento para o algoritmo de machine learning.

### 2.5.1 Stopwords

Ao extrair os textos de cada item e colocá-los em uma matriz utilizando o saco de palavras vem com isso vários termos irrelevantes que aparecem diversas vezes durante o texto como pronomes e conjunções por exemplo, devem ser removidos pois vão poluir e corromper o aprendizado do algoritmo (FALK, 2019). Para isso é utilizado o algoritmo stopwords removendo todos os termos automaticamente em uma lista pré-definida.

### 2.5.2 Lematização

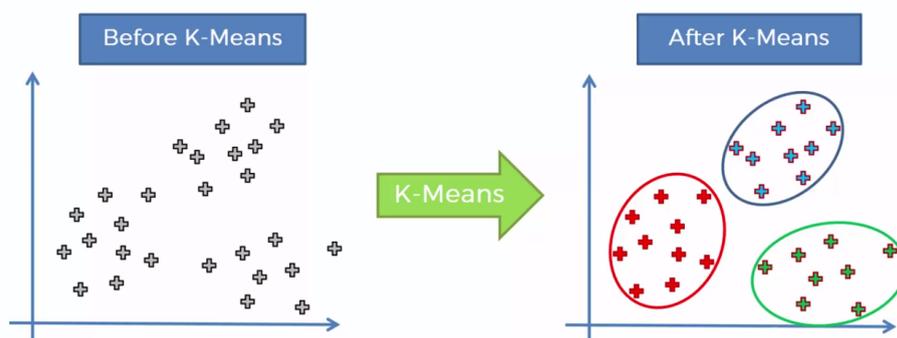
A *lematização* segue a ordem gramatical correta do idioma com a análise morfológica de palavras, normalmente com o objetivo de remover apenas desinências flexionais e retornar a sua forma básica ou ao seu lema. Essa técnica analisa palavra por palavra, preservando assim as características distintas que algumas palavras podem perder se elas fossem simplesmente agrupadas com semelhantes (MANNING, 2019).

## 2.6 ALGORITMO K-MEANS

K-means é um dos algoritmos mais utilizados em machine learning, ele utiliza uma técnica de agrupamento conhecida como clustering que consiste em separar objetos de acordo com suas características utilizando um método não hierárquico de agrupamento que agrupa os itens de acordo com a proximidade de semelhança das características, onde a sigla K é a quantidade de grupos estabelecida previamente (LINDEN, 2009).

De acordo com a definição figura 05 um grupo também pode ser definido como uma aglomeração de pontos no espaço tal que a distância entre quaisquer dois pontos no grupo é menor do que a distância entre qualquer ponto no grupo e qualquer ponto fora destes (EVERITT, 2001).

Figura 05: K-Means.



Fonte: medium (2020)

### 2.6.1 Escolha Adequada do Número de Clusters

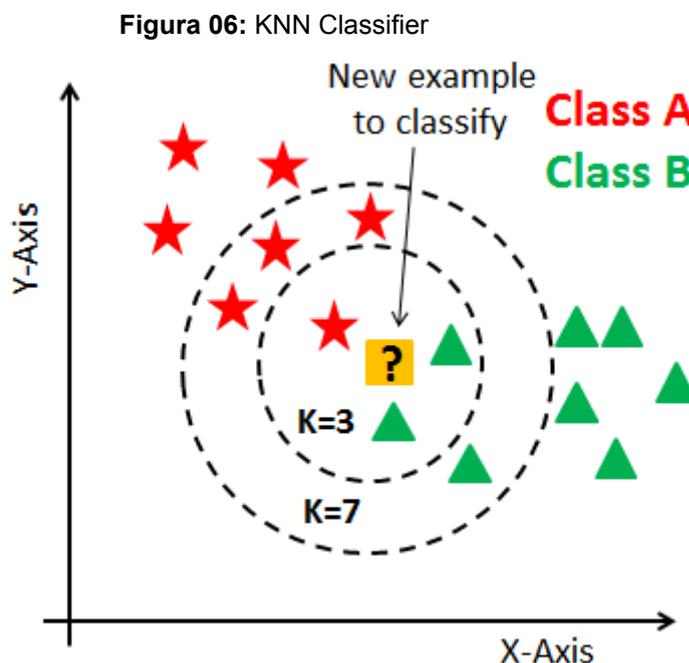
O método de Elbow tem por objetivo efetuar a clusterização dentro do intervalo de valores dos dados disponíveis já definidos pelo supervisor, no método é efetuada a soma dos quadrados dos erros (SSE) sobre as distâncias dos itens de cada classe em relação aos centróides. O ponto de inflexão, o qual é o número aproximado que deve ser adotado pela variável  $k$  de clusterização para uma distribuição eficiente dos itens, se dá quando o decréscimo da SSE torna-se menor ao passo que aumenta a unidade do número de classes. (MORAIS, 2016)

Segundo ROUSSEEUW (1987) “Silhouette se refere a um método de interpretação e validação de consistência dentro de grupos de dados. A técnica fornece uma representação gráfica sucinta de quão bem cada objeto foi classificado”.

## 2.7 ALGORITMO KNN

Segundo Bodadilla et al (2011) o algoritmo K - Nearest Neighbors (KNN) faz a predição da vizinhança  $k$  do usuário solicitante da recomendação com base em suas classificações, positivas ou negativas, para então fazer a previsão das avaliações do usuário.

KNN trata-se de um método para obter a classificação de objetos que se encontrem o mais próximo possível do objeto em comparação, pode ser lido também como a quantidade K de vizinhos mais próximos pela similaridade entre eles (DANADES et al, 2016).



Fonte: Inferir (2019)

O modelo KNN é bem simples de ser compreendido, apenas é necessário que tenha uma noção de distância e que quanto maior é a proximidade dos pontos, maior é a similaridade entre eles (GRUS, 2016).

## 2.8 PROBLEMA DO USUÁRIO NOVO (COLD START)

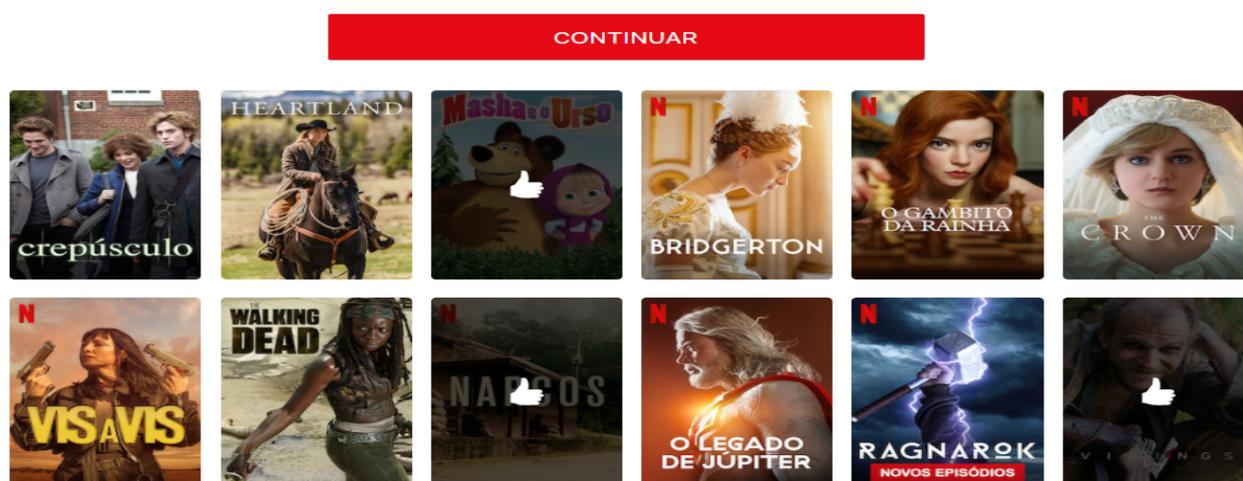
O usuário ao entrar na aplicação não possui ainda um perfil de usuário com informações suficientes para uma boa recomendação personalizada, com isso pode se sentir deslocado ao se deparar com conteúdos muitas vezes aleatórios e não se identificar com o que está sendo apresentado, com isso pode ocorrer a desistência do usuário em permanecer no ambiente. (BOBADILLA et al, 2012).

Conforme exemplificado na Figura 07, uma solução para esse problema é fazer o próprio usuário treinar o algoritmo, selecionando previamente uma série de conteúdos que mais se adequa ao seu interesse, para que o algoritmo treine e se adapte ao seu gosto pessoal, antes dele começar a fazer uso da aplicação. Essa técnica é chamada de coleta explícita (ADOMAVICIUS; TUZHILIN, 2010).

**Figura 07:** Recomendação de conteúdo.

**③ Luiz, escolha 3 favoritos.**

Vamos usar seus favoritos para achar mais filmes e séries para você! Clique nos títulos que você gosta!



Fonte: Netflix (2021)

### 3 RECOMENDAÇÃO DE PRATOS GASTRONÔMICOS

Este capítulo tem como objetivo apresentar o sistema desenvolvido para recomendações de pratos baseado nos gostos pessoais. O sistema desenvolvido tem como foco a área de gastronomia, o objetivo é fazer recomendações para que facilite na hora da escolha de um prato pelo consumidor.

Apesar da área apresentada no objeto de estudo, a gastronomia é um segmento difícil de oferecer recomendações para usuários, pois, além da quantidade variada de características presentes em cada prato, tem o fator “gosto pessoal”, o gosto pessoa varia conforme de pessoa á pessoa, o que torna a recomendação por filtragem colaborativa não tão eficiente, pois as recomendações podem variar muito e o usuário pode não se identificar com o que foi apresentado á ele. Outros fatores como o veganismo, que por mais que a pessoa possa vir a gostar daquela receita, esta pessoa acabe não optando por ela por envolver questões superiores ao seu paladar, outra questão é que

determinadas pessoas são alérgicas a certos alimentos, o que também pode acabar resultando em uma não receptividade quanto às recomendações. Mas essa abordagem é útil para mostrar novas opções que o usuário ainda não tenha experimentado e possa a vir a gostar.

A recomendação pelo conteúdo e a recomendação de forma colaborativa no geral funcionam muito bem para as mais diversas áreas que estão empregadas. Com isso esse trabalho busca fazer a melhor recomendação utilizando as técnicas já conhecidas de recomendação no meio acadêmico.

Os diferentes sistemas de recomendação podem oferecer resultados mais interessantes quando usados em conjunto, o filtro colaborativo e o filtro por conteúdo que quando combinados se complementam. O colaborativo faz com que as preferências do consumidor que está interagindo com o sistema sejam comparadas com as preferências dos demais consumidores, dessa forma será recomendado ao consumidor pratos cujo os gostos pessoais mais se aproximam dos gostos do consumidor que está interagindo com o sistema, sendo a recomendação baseada nos pratos em que o consumidor pareado mais consumiu e avaliou como positiva.

Serão apresentadas as partes principais do desenvolvimento, começando pela inserção do *dataset* para o problema, passando pela análise e estruturação dos dados, o funcionamento da classe do consumidor e como se dará as adições de pratos ao seu histórico de pratos consumidos. Tendo como foco principal a utilização da tabela ajustada para que pudesse utilizar o algoritmo de aprendizado de máquina e por fim efetuar a avaliação sobre a clusterização das receitas.

O sistema de recomendação é composto por algumas técnicas importantes. O que torna um sistema de recomendação preciso é a capacidade de surpreender o usuário e não apenas recomendar o previsível, ou o que o usuário espera que seja recomendado a ele. Nesse contexto, o sistema de recomendação desenvolvido neste empreendimento foi projetado com as técnicas de filtragem:

A filtragem colaborativa tem como princípio fazer a recomendação com base na combinação por comparação de receitas dispostas no histórico de receitas do consumidor chave que é o que está solicitando a recomendação e a lista de receitas consumidas

pelos demais usuários. A partir da comparação é recomendado ao usuário o prato que mais combina com os gostos do usuário encontrado, o que proporciona a recomendação é aplicação da diferença entre os pratos que o usuário chave já consumiu e os pratos que o usuário que ocorreu a combinação já consumiu. Ou seja, o item que será recomendado é o item ao qual o usuário chave não possui, porém que o usuário recomendado por proximidade de preferências possui e avaliou positivamente.

Dentro da filtragem colaborativa há os filtros por popularidade que fazem a recomendação baseado em quantas vezes o usuário pareado consumiu a refeição. E não menos importante, o filtro por avaliação, que faz a recomendação ao usuário baseado na nota de um a dois, sendo um equivalente a “não gostou” e dois equivalente a “gostou”.

Além da filtragem colaborativa, foi utilizada a filtragem baseada em conteúdo, que recomenda para o usuário pratos que contenham características similares ao prato que o consumidor tem preferência. Os dois filtros de recomendações por colaboração e por conteúdo combinados propiciam uma recomendação com mais assertividade, pois uma técnica é capaz de anular o defeito da outra.

### 3.1 TECNOLOGIAS UTILIZADAS

Para este projeto foi utilizada a linguagem de programação *python* que proporciona excelentes bibliotecas para construir projetos com aprendizado de máquina de forma eficiente.

O ambiente de desenvolvimento utilizado foi o *Jupyter Notebook* através do distribuidor de linguagem *Anaconda*, o distribuidor *Anaconda* já vem com diversas bibliotecas instaladas, formando um ambiente completo para o desenvolvimento de aplicações de *data science*, o *Jupyter Notebook* vem como ferramenta padrão para o desenvolvimento o qual fornece um ambiente que para além de proporcionar um desenvolvimento limpo, permite a documentação do que está sendo desenvolvido de forma simples.

Foram utilizadas neste projeto as bibliotecas *pandas*, *numpy*, *matplotlib* e *sklearn*. Cada uma das bibliotecas utilizadas neste projeto tem funções próprias de extrema importância. Sendo a biblioteca *pandas* a parte essencial na hora de fazer a análise de

dados e a manipulação desses dados. A biblioteca *numpy* por sua vez se mostrou excelente para manipular tabelas, reestruturá-las e fazer operações matemáticas matriciais com os dados. A biblioteca principal do projeto a *sklearn* foi a que ofereceu os algoritmos responsáveis pelo aprendizado de máquina, enquanto que a biblioteca *matplotlib* por sua vez, mostrou-se uma ferramenta muito interessante para que mostrasse os resultados matemáticos extraídos do aprendizado de forma gráfica.

### 3.2 RECOMENDAÇÃO BASEADA EM CONTEÚDO K-MEANS

Os dados presentes no dataset utilizado neste projeto contam com um conjunto de receitas retiradas do site “Tudo gostoso” e o dataset utilizado foi desenvolvido de autoria própria. A recomendação por conteúdo foi baseada na coluna “características” do dataset (Figura 08), a qual conta com diversas características de cada prato.

**Figura 08:** Dataset.

```

1 import numpy as np
2 import pandas as p
3 from scipy import sparse
4 from sklearn.metrics.pairwise import cosine_similarity

```

```

1 dataset = p.read_json('CulinariaBrasileira_ATT.json')

```

ID	Restaurante	nome	ingredientes	características	K-classes
0	Bom Apetite	LASANHA À BOLONHESA	[margarina, farinha de trigo, leite, creme de ...	{'Textura': 3, 'stDoce': 0, 'stQuente': 1, 'Ca...	3
1	Sra Fome	PANQUECA DE CARNE MOÍDA	[farinha de trigo, leite, ovo, óleo, sal, carn...	{'Textura': 3, 'stDoce': 0, 'stQuente': 1, 'Ca...	3
2	Sra Fome	MACARRÃO COM REQUEIJÃO	[macarrão, azeite de oliva, alho, tomate, requ...	{'Textura': 3, 'stDoce': 0, 'stQuente': 1, 'Ca...	3
3	Divino Sabor	Pudim de Leite Moça	[leite condensado, leite, ovo, açúcar]	{'Textura': 1, 'stDoce': 1, 'stQuente': 0, 'Ca...	7
4	Delícias Caseiras	Bolo de Banana com Calda	[banana, farinha de trigo, açúcar para bolo, a...	{'Textura': 3, 'stDoce': 1, 'stQuente': 0, 'Ca...	0
...	...	...	...	...	...
61	Pé de fava	MOUSSE DE LIMÃO	[leite condensado, creme de leite, limão]	{'Textura': 1, 'stDoce': 1, 'stQuente': 0, 'Ca...	7
62	Pé de fava	CALDEIRADA DE FRUTOS DO MAR	[cação, lagostin, camarão, mexilhão, lagostin, ...	{'Textura': 1, 'stDoce': 0, 'stQuente': 1, 'Ca...	1
63	Pé de fava	TALHARIM COM FRUTOS DO MAR E MOLHO BRANCO	[talharim, cebola, pimentao, alho, camarão, cr...	{'Textura': 3, 'stDoce': 0, 'stQuente': 1, 'Ca...	10
64	Pé de fava	PAELLA DE FRUTOS DO MAR	[tomate, cebola, alho, sal, salsa, curry, arro...	{'Textura': 3, 'stDoce': 0, 'stQuente': 1, 'Ca...	1
65	Pé de fava	RISOTO DE FRUTOS DO MAR	[arroz, polvo, camarão, mexilhão, lula, vinho ...	{'Textura': 3, 'stDoce': 0, 'stQuente': 1, 'Ca...	1

**Fonte:** Autoria própria (2021)

Inicialmente com o auxílio da biblioteca *pandas*, foi necessário ser feita a conversão dos dados de formato *JSON* para o formato de *DataFrame*. O que resultou em seis colunas, sendo a primeira o código do prato, sendo o código importante para que se

pudesse localizar de forma mais rápida na tabela e para facilitar na recomendação por conteúdo. Na segunda coluna apresenta-se o nome do restaurante, que é de extrema importância para que o consumidor efetue a escolha correta do restaurante que tem o seu prato recomendado. Já a terceira coluna refere-se ao nome do prato. A quarta coluna do dataset é a coluna que apresenta os ingredientes que compõem o prato. A quinta coluna é a coluna chave da recomendação onde se encontram as características da receita, as quais são peça fundamental para a recomendação e clusterização para este projeto. A última coluna serve para guardar a informação de clusterização caracterizada de forma supervisionada para cada receita.

Foi necessário fazer a filtragem do dataset para que apresentasse as colunas do código do prato e das características do prato (figura 09), fazendo com que o dataset apenas mostrasse o código da receita e as características presentes em cada receita.

**Figura 09:** Separação por características.

```

1 datasetM = dataset['caracteristicas'].apply(p.Series)
2
3 qtd_pratos = 89
4
5 ColaborativoIngredientesReceitas = datasetM.iloc[:qtd_pratos,]
6
7 ColaborativoIngredientesReceitas.insert(0, 'ID', ColaborativoIngredientesReceitas.index)
8
9 ColaborativoIngredientesReceitas

```

	ID	Textura	stDoce	stQuente	Carne_Tipo	Laticinio_Tipo	stQueijo	Gluten_Tipo	stFermento	Massa_Tipo	stUnidade	stCacau	Classe
0	0	3	0	1	1	1	1	1	0	1	0	0	12
1	1	3	0	1	1	1	1	1	0	1	1	0	9
2	2	3	0	1	0	1	1	1	0	2	0	0	9
3	3	1	1	0	0	1	0	0	0	0	0	0	5
4	4	3	1	0	0	1	0	1	1	0	0	0	3
...	...	...	...	...	...	...	...	...	...	...	...	...	...
61	61	1	1	0	0	1	0	0	0	0	0	0	1
62	62	1	0	1	4	0	0	0	0	0	0	0	0
63	63	3	0	1	4	1	1	0	0	2	0	0	9
64	64	3	0	1	4	0	0	0	0	0	0	0	0
65	65	3	0	1	4	0	0	0	0	0	0	0	0

**Fonte:** Autoria própria (2021)

A similaridade dos pratos se deu pelas características (Figura 10) que cada receita possui. A textura e o sabor são as características mais relevantes deste *dataset*. Sendo que a característica relacionada à textura é dividida em quatro opções, sendo a opção zero que é inserida quando a comida for líquida, um quando a comida for cremosa, dois

quando ela se apresentar no estado pastoso, e três quando a receita se apresentar no estado sólido.

**Figura 10:** Características das Receitas.

	0	1	2	3	4
Qual a Textura?	Líquida	Cremosa	Pastosa	Sólida	
Qual o Sabor?	Salgado	Doce			
É Servido por Unidade?	Não	Sim			
Servida Quente?	Não	Sim			
Qual o Tipo de Carne?	Sem Carne	Vermelha	Branca	Peixe	Fruto do Mar
Tem Laticínio na Receita?	Não	Sim			
Qual o Tipo de Massa	Sem Massa	Pastel/lasanha/etc	Espaguetti/Penne/etc		
Vai Queijo na Receita?	Não	Sim			
Vai Fermento na Receita?	Não	Sim			
Vai cacau na receita?	Não	Sim			
Qual o Tipo de Gluten?	Sem Gluten	Farinha de Trigo	Pão	Bolacha	
A Qual Classe Pertence o Alimento?					

FrutoDoMar	0
Mousse	1
Pavê	2
Bolo/cupcake	3
TortaDoce	4
Pudim	5
TortaSalgada/bolo	6
SopaECaldo	7
Creme	8
Massa	9
Lanche	10
pizza	11
lasanha	12

Fonte: Autoria própria (2021)

### 3.3 CONSUMIDOR

A classe consumidor criada (Figura 11), possibilitou a associação do código do consumidor aos pratos que ele já consumiu e os pratos que ao efetuar o cadastro foram previamente selecionados como sendo do gosto pessoal do consumidor. Possibilitando assim a fuga do problema conhecido como *cold-start*, adicionando as receitas prévias à lista de pratos já consumidos, formando assim um histórico das preferências do consumidor mesmo que este ainda não tenha consumido nenhuma receita.

**Figura 11:** Classe do Consumidor.

```

import math

class Consumidor:
    avaliacao = 0
    consumo = 0

    def __init__(self, nome, Id):
        self.nome = nome
        self.pratos = []
        self.UserID = Id

    def addPrevios(self, prato):
        lista = prato.tolist()
        lista.append(0)
        lista.append(2)
        lista.append(self.UserID)
        self.pratos.append(lista)

def add(self, prato, avaliacao):
    if math.isnan(avaliacao):
        print("A avaliação deve ser um número inteiro!")
    else:
        if avaliacao < 0 and avaliacao > 1:
            print("A avaliação deve estar entre 0 e 1 sendo 0 - Não gostou e 1 - Gostou")
        else:
            lista = prato.tolist()
            achou = False
            if (len(self.pratos) > 0):
                for i in range(len(self.pratos)):
                    if self.pratos[i][0] == lista[0]:
                        self.pratos[i][-3] = self.pratos[i][-3] + 1
                        self.pratos[i][-2] = avaliacao
                        achou = True
            if achou == False:
                lista.append(1)
                lista.append(avaliacao)
                lista.append(self.UserID)
                self.pratos.append(lista)
            else:
                lista.append(1)
                lista.append(avaliacao)
                lista.append(self.UserID)
                self.pratos.append(lista)

```

Fonte: Autoria própria (2021)

Na classe do consumidor (Figura 11) a função `add`, adiciona pratos à lista do consumidor já com a avaliação e quantidade de vezes que foi consumido. Se o prato for adicionado previamente para o sistema de recomendação, o prato é adicionado a lista dos

pratos consumidos do consumidor apenas para auxílio da filtragem colaborativa. Sendo assim, o prato recebe o valor de consumo igual a zero e avaliação positiva igual a dois.

### 3.4 FILTRAGEM POR CONTEÚDO K-MEANS

A escolha do algoritmo k-means para este projeto se deu pelo fato de ser um algoritmo não supervisionado que permite que se faça uma classificação para uma situação onde não se sabe a qual grupo de receitas um determinado prato está classificado. Além de ser um algoritmo popular no meio acadêmico, o algoritmo possui um vasto arsenal de conteúdo publicado na internet pelas mais diversas línguas.

#### 3.4.1 Validação do Número de Clusters

Para uma escolha eficaz do número de clusters foi utilizado o método Silhouette (Figura 12), que faz cálculos para determinar a proximidade entre as características dos itens e com base nesse princípio julgar qual o número de classes mais eficiente para efetuar a separação dos pratos em clusters.

**Figura 12:** Silhouette score.

#### K-means

```

1 from sklearn.metrics import silhouette_samples, silhouette_score
2 import matplotlib.pyplot as plt
3 from sklearn.cluster import KMeans
4 import warnings
5 warnings.filterwarnings("ignore")
6
7 %matplotlib inline

```

#### Escolha automática do K mais adequado através do método Silhouette

```

1 ##Transformação das distancia entre as características para distancia coseno
2 length = np.sqrt((ContentNP**2).sum(axis=1))[:,None]
3
4 ContentNP = ContentNP / length
5
6 Kmeans_por_k = [KMeans(n_clusters=k, random_state=200).fit(ContentNP) for k in range(1, 40)]
7
8 silhouette_scores = [silhouette_score(ContentNP, model.labels_, metric="cosine") for model in Kmeans_por_k[1:]]
9
10 silhouette_scores
11
12 number_of_clusters = np.argmax(silhouette_scores)

```

**Fonte:** Autoria própria (2021)

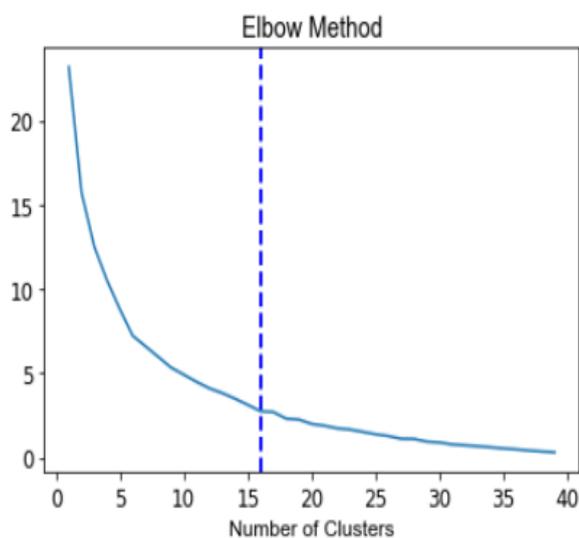
Imprescindível a utilização deste método antes do método de Elbow (Figura 13), pois possibilita a escolha automática do número adequado de clusters.

Figura 13: K-means e Elbow.

```

1 wcss = []
2
3
4 for i in range(1,40):
5     kmeans = KMeans(n_clusters=i, init = 'k-means++', random_state=0).fit(ContentNP)
6     wcss.append(kmeans.inertia_)
7
8 plt.title('Elbow Method', family = 'Arial', fontsize = 14, color = 'black')
9 plt.xlabel('Number of Clusters', family = 'Arial', fontsize = 12, color = 'black')
10 plt.xticks(fontsize = 12, color= 'black')
11 plt.yticks(fontsize = 12, color= 'black')
12 plt.axvline(x = number_of_clusters, linewidth = 2, color = 'blue', linestyle = '--')
13 plt.plot(range(1,40), wcss)
14
15 k = KMeans(n_clusters= number_of_clusters, random_state=0).fit(ContentNP)
16
17 k.labels_
18
19
20 dataset['K-classes'] = k.labels_

```



Fonte: Autoria própria (2021)

Devido ao número de classes ser incerto utilizou-se o método de Elbow que consiste em analisar os dados e apresentar o esboço do gráfico para a verificação de qual é o melhor número de classes para que o algoritmo faça a distribuição mais aproximada e a classificação em clusters para cada receita.

Figura 14: Recomendação por conteúdo.

```

1 def RecomendacaoBaseadaEmConteudo(Consumidor, df):
2
3     classe = 0
4     selecionado = 0
5     maisConsumido = 0
6     achou = False
7
8     for receita in Consumidor.pratos:
9         if receita[-2] == 2:
10            if receita[-3] > maisConsumido:
11                maisConsumido = receita[-1]
12                selecionado = receita[0]
13                achou = True
14
15     if achou == True:
16         classe = df._get_value(selecionado, 'K-classes')
17         dataF = df.loc[dataset['K-classes'] == classe]
18
19         print("\nMais Consumido/Melhor Avaliado pelo consumidor:")
20         display(df.loc[dataset['ID'] == selecionado])
21         print("\nRECOMENDAÇÃO:")
22         display(dataF.sample())
23
24     else:
25         print("Não há prato a ser recomendado, o consumidor não selecionou suas preferências.")
26
27
28
29
30
31 RecomendacaoBaseadaEmConteudo(Consumidor1, dataset)
32

```

Mais Consumido/Melhor Avaliado pelo consumidor:

ID	Restaurante	nome	ingredients	caracteristicas	K-classes	
4	4	Delícias Caseiras	Bolo de Banana com Calda	[banana, farinha de trigo, açúcar para bolo, a...	{'Textura': 3, 'stDoce': 1, 'stQuente': 0, 'Ca...	13

RECOMENDAÇÃO:

ID	Restaurante	nome	ingredients	caracteristicas	K-classes	
23	23	Delícias Caseiras	PÃO DOCE RECHEADO DE GOIABADA	[farinha de trigo, açúcar, fermento, leite, ov...	{'Textura': 3, 'stDoce': 1, 'stQuente': 0, 'Ca...	13

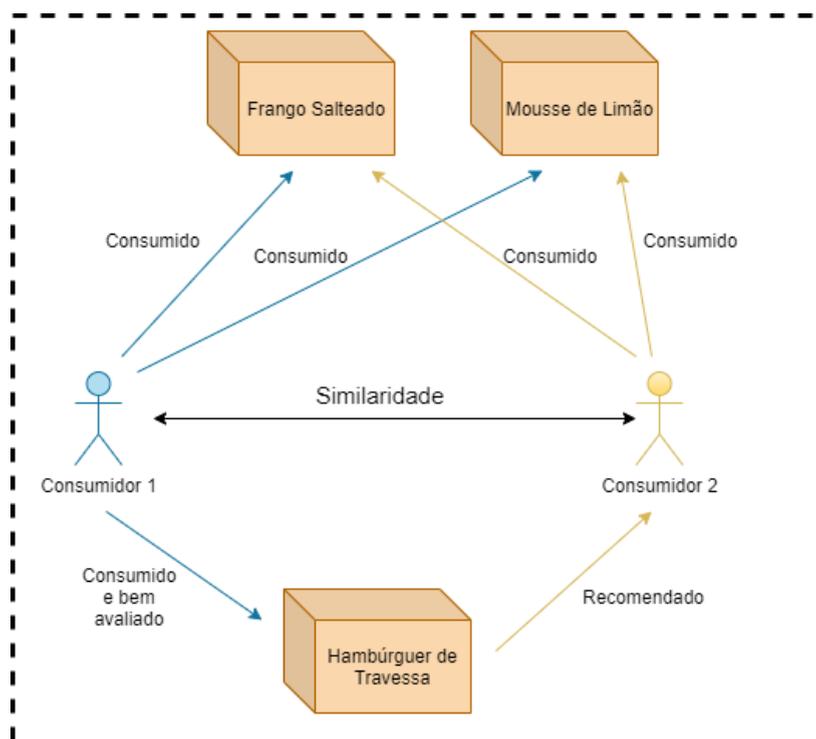
Fonte: Autoria própria (2021)

A função “*RecomendacaoBaseadaEmConteudo*” (Figura 14) faz a recomendação pela similaridade do prato mais bem avaliado pelo usuário e também mais consumido. A partir do prato melhor avaliado é feita a busca no dataset pelos pratos com o mesmo número da classe sugestionada pelo K-means, ou seja, utilizando a que possui maior proximidade em relação ao vetor de características do prato melhor avaliado pelo consumidor.

### 3.5 FILTRAGEM COLABORATIVA

A função da busca pelo usuário mais próximo no quesito similaridade está ligada diretamente ao método de filtragem colaborativa (Figura 15).

**Figura 15:** Recomendação por filtragem colaborativa.



**Fonte:** Autoria própria (2021)

A utilização da filtragem colaborativa propiciou a recomendação de forma dinâmica, não apenas de uma recomendação com base nas características dos pratos consumidos, mas uma recomendação que faz a similaridade entre os gostos pessoais de cada consumidor e a partir da proximidade de gostos efetua a recomendação (Figura 15).

Para a recomendação por meio do filtro colaborativo foi necessário fazer o tratamento dos dados (Figura 16) previamente à utilização do algoritmo.

Figura 16: Preparação dos dados.

```

1 lstPratos = []
2
3 for consumidor in lstConsumidores:
4     for prato in consumidor.pratos:
5         lstPratos.append(prato)
6
7 ##Lista com todos os pratos consumidos pelos Consumidores
8 a = lstPratos
9 pratos = []
10 tabela = []
11
12 for x in range(0, len(a)):
13     achou = False
14     for i in pratos:
15         if a[x][0] == i:
16             achou = True
17             break;
18     if achou == False:
19         pratos.append(a[x][0])
20
21 pratos.sort()
22 tabela.append(pratos)
23
24 for Consumidor in lstConsumidores:
25     lst = []
26     lst.append(Consumidor.UserID)
27     consumiu = False
28     avaliacao = 0
29
30     for pt in range(0, len(tabela[0])):
31         avaliacao = 0
32         for prato in Consumidor.pratos:
33             if prato[0] == tabela[0][pt]:
34                 consumiu = True
35                 avaliacao = prato[-2]
36                 break
37         if consumiu == True:
38             lst.append(avaliacao)
39         else:
40             lst.append(avaliacao)
41
42     tabela.append(lst)
43
44 tabela[0].insert(0, 'ID')
45
46 a = p.DataFrame(tabela)
47
48 df = a
49
50 #inserção da primeira linha como nome de colunas
51 df.columns = a.iloc[0]
52
53 #Transformando a coluna dos indicadores em index(row)
54 df = df.set_index('ID')
55
56 #Excluindo a primeira linha onde ficava os nomes de cada característica.
57 df = df.drop('ID')
58
59 df = df.rename_axis(index=None)
60 X = df.astype(float)
61
62 #Trocando o valor do código dos pratos pelo nome dos pratos.
63 for col_name in X.columns:
64     X.rename(columns={col_name: dataset._get_value(col_name, 'nome')}, inplace=True)

```

Fonte: Autoria própria (2021)

Para a recomendação colaborativa foi utilizado o algoritmo supervisionado KNN (figura 17), o qual proporciona uma recomendação pela avaliação que cada consumidor

faz para o prato consumido. Os dados apresentados (figura 17) mostram o distanciamento entre cada prato.

Figura 17: KNN.

### KNN - K-Nearest Neighbors

```

1 def standardize(row):
2     new_row = (row-row.mean())/(row.max()-row.min())
3     return new_row
4
5 df_std = X.apply(standardize)
6
7 item_similarity = cosine_similarity(df_std.T)
8
9 item_similarity_df = p.DataFrame(item_similarity, index = X.columns, columns = X.columns)
10 item_similarity_df

```

	Pudim de Leite Moça	Bolo de Banana com Calda	TORTA DE LIQUIDIFICADOR DE FRANGO	BOLO DE MILHO DE LATA	TORTA DE TAPIOCA COM FRANGO	BOLO DE MILHO CREMOSO	PAVÊ DE MORANGO COM CHOCOLATE	BOLO GELADO	PANQUECA AMERICANA	TORTA ALEMÃ	BOLO SALGADO DE SALSICHA E MILHO	FRANG XADREZ
Pudim de Leite Moça	1.000000	0.188887	0.612372	0.612372	1.000000	0.188887	0.612372	0.612372	-0.888887	-0.888887	-0.888887	-0.888887
Bolo de Banana com Calda	0.188887	1.000000	-0.408248	0.612372	0.188887	1.000000	0.612372	-0.408248	-0.888887	-0.888887	-0.888887	-0.888887
TORTA DE LIQUIDIFICADOR DE FRANGO	0.612372	-0.408248	1.000000	-0.250000	0.612372	-0.408248	-0.250000	1.000000	-0.408248	-0.408248	-0.408248	-0.408248
BOLO DE MILHO DE LATA	0.612372	0.612372	-0.250000	1.000000	0.612372	0.612372	1.000000	-0.250000	-0.408248	-0.408248	-0.408248	-0.408248
TORTA DE TAPIOCA COM FRANGO	1.000000	0.188887	0.612372	0.612372	1.000000	0.188887	0.612372	0.612372	-0.888887	-0.888887	-0.888887	-0.888887
BOLO DE MILHO CREMOSO	0.188887	1.000000	-0.408248	0.612372	0.188887	1.000000	0.612372	-0.408248	-0.888887	-0.888887	-0.888887	-0.888887
PAVÊ DE MORANGO COM CHOCOLATE	0.612372	0.612372	-0.250000	1.000000	0.612372	0.612372	1.000000	-0.250000	-0.408248	-0.408248	-0.408248	-0.408248
BOLO GELADO	0.612372	-0.408248	1.000000	-0.250000	0.612372	-0.408248	-0.250000	1.000000	-0.408248	-0.408248	-0.408248	-0.408248
PANQUECA AMERICANA	-0.888887	-0.888887	-0.408248	-0.408248	-0.888887	-0.888887	-0.408248	-0.408248	1.000000	1.000000	1.000000	1.000000
TORTA ALEMÃ	-0.888887	-0.888887	-0.408248	-0.408248	-0.888887	-0.888887	-0.408248	-0.408248	1.000000	1.000000	1.000000	1.000000
BOLO SALGADO DE SALSICHA E MILHO	-0.888887	-0.888887	-0.408248	-0.408248	-0.888887	-0.888887	-0.408248	-0.408248	1.000000	1.000000	1.000000	1.000000
FRANG XADREZ	-0.888887	-0.888887	-0.408248	-0.408248	-0.888887	-0.888887	-0.408248	-0.408248	1.000000	1.000000	1.000000	1.000000
RISOTO DE FRUTOS DO MAR	-0.408248	-0.408248	-0.250000	-0.250000	-0.408248	-0.408248	-0.250000	-0.250000	0.612372	0.612372	0.612372	0.612372

Fonte: Autoria própria (2021)

O distanciamento aqui empregado refere-se à similaridade de um prato em relação ao outro. Sendo que quanto menor a distância entre uma receita e outra, maior é a similaridade entre elas.

No momento de fazer a recomendação do prato mais adequado ao consumidor (figura 18). Aparecerá sempre acima os pratos que o consumidor já possui. E após eles

aparecerá o prato mais recomendado de acordo com o pareamento feito pelas preferências de cada consumidor.

**Figura 18:** Recomendação por colaboração.

```

1
2 def get_similar(nomeprato, avaliacao):
3     similar_score = item_similarity_df[nomeprato]*(avaliacao)
4     similar_score = similar_score.sort_values(ascending=False)
5
6     return similar_score
7
8 pratos_similares = p.DataFrame()
9
10 for prato in Consumidor4.pratos:
11     pratos_similares = pratos_similares.append(get_similar(dataset._get_value(prato[0], 'nome'), prato[-2]), ignore_index=True)
12
13 pratos_similares.head()
14 pratos_similares.sum().sort_values(ascending=False)
15

```

BOLO SALGADO DE SALSICHA E MILHO	8.000000
FRANGO XADREZ	8.000000
PANQUECA AMERICANA	8.000000
TORTA ALEMÃ	8.000000
RISOTO DE FRUTOS DO MAR	4.898979
BOLO DE MILHO DE LATA	-3.265986
PAVÊ DE MORANGO COM CHOCOLATE	-3.265986
BOLO GELADO	-3.265986
TORTA DE LIQUIDIFICADOR DE FRANGO	-3.265986
BOLO DE MILHO CREMOSO	-5.333333
Bolo de Banana com Calda	-5.333333
Pudim de Leite Moça	-5.333333
TORTA DE TAPIOCA COM FRANGO	-5.333333

dtype: float64

**Fonte:** Autoria própria (2021)

## 4 CONCLUSÃO

Este projeto teve como finalidade a construção de um sistema de recomendações para propor pratos para os consumidores baseado no conteúdo dos pratos favoritos do consumidor e também na recomendação baseada na colaboração dos demais consumidores presentes no sistema.

Os livros e artigos presentes nas referências obtidas por este projeto contribuíram para elucidar temas importantes no decorrer do desenvolvimento. Ajudando a desenvolver o algoritmo e projeto para a seleção das melhores características dos pratos, os classificando para fazer uma boa recomendação e clusterização de dados com base nos dados previamente tratados utilizando abordagens referenciadas ao decorrer do artigo.

Notou-se que a quantidade de dados não é a principal métrica e pode até mesmo prejudicar a aprendizagem do algoritmo, poluindo com informações desnecessárias e diminuindo sua acurácia, com isso devemos sempre reavaliar as escolhas das características juntamente com um conhecimento profundo sobre o objeto a ser

classificado, se levando em consideração a qualidade dos dados obtidos e as métricas que vão ser analisadas.

## REFERÊNCIAS

ADOMAVICIUS, Gediminas; TUZHILIN A. **Recommender Systems Handbook**. 01. ed. Minnesota, US: Springer p. 217-253, 2010.

AFENI, B. O.; ARULEBA, T. I.; OLOYEDE, I. A. “**Hypertension Prediction System Using Naive Bayes Classifier**”; 2017. Disponível em :  
<<https://journaljamcs.com/index.php/JAMCS/article/download/23950/44739#:~:text=This%20research%20has%20developed%20hypertension,of%20patients%20getting%20a%20hypertension>> Acessado em 10 jun. 2021.

BOBADILLA, J. et al. **A collaborative filtering approach to mitigate the new user cold start problem. Knowledge-Based System**, Spain, v.26, p. 224-238, 02/2012.  
Disponível em: <<https://www.dblp.org/db/journals/kbs/kbs26.html>> Acesso em 02 jun. 2021.

BOBADILLA, J., Hernando, A., Ortega, F. e Bernal, J. **A framework for collaborative filtering recommender systems. Expert Systems with Applications**, v. 38, n. 12, p.14609–14623, 2011.

BAUDISCH, A. R. **Ciência de dados é explorar big data para fazer perguntas para prever o futuro. Medium**, 2016. Acesso em: 10 nov. 2018. Disponível em:  
<<https://medium.com/@AlfredBaudisch/o-que-é-ciência-de-dados-data-science-7af5bdac101a>>.Acessado em 15 mai. 2021.

CASTROUNIS, A. **AI for People and Business: A Framework for Better Human Experiences and Business Success** - 2019.

CUNHA, M. N; SIMÕES, A. L. “**Comunicação na internet e a violação do direito à privacidade: uma análise avaliativa das políticas e termos de uso na internet**” - 2018  
Disponível em:  
<[https://seer.uscs.edu.br/index.php/revista\\_comunicacao\\_inovacao/article/download/5093/2472/16961](https://seer.uscs.edu.br/index.php/revista_comunicacao_inovacao/article/download/5093/2472/16961)> Acessado em 14 jun. 2021.

D’ADDIO, Rafael Martins. **Filtragem baseada em conteúdo auxiliada por métodos de indexação colaborativa**. Tese(mestrado) - Ciências da computação e Matemática Computacional, Universidade de São paulo Campus de São Carlos. São Paulo, p. 30-31. 2015. Disponível em:  
[https://www.teses.usp.br/teses/disponiveis/55/55134/tde-26112015-144141/publico/RafaelDadio\\_revisada.pdf](https://www.teses.usp.br/teses/disponiveis/55/55134/tde-26112015-144141/publico/RafaelDadio_revisada.pdf) . Acesso em 10 jun. 2021.

DANADES, A. et al. **“Comparison of Accuracy Level K-Nearest Neighbor Algorithm and Support Vector Machine Algorithm in Classification Water Quality Status”** pp. 137–141, 2016.

Everitt, B. S., Landau, S., and Leese, M., **Cluster 55 Analysis, Arnold**, 4th Edition, 2001.

FALK, K. **Practical Recommender Systems**. 01. ed. Shelter Island, Nova York, EUA: Manning p. 249-273, 2019.

FAWCETT, T.; PROVST, F. **Data Science para Negócios** - 2016

GRUS, J. **Data Science do Zero Primeiras Regras com o Python** - 2016

LIMA, P. S. R. **Personalização de Interfaces Web para Sites Institucionais com Base em Perfis de Usuários**. 2002. Dissertação (Mestrado em Ciência da Computação) - Instituto de Informática, UFRGS, Porto Alegre - RS. Disponível em: <<https://lume.ufrgs.br/handle/10183/3802>> Acessado em 3 mai. 2021.

Linden, Ricardo **Técnicas de Agrupamento - Revista de Sistemas de informação FSMA n. 4**, pp. 18-36, 2009.

MANGEL, M.; SAMANIEGO, F. J. **Abraham Wald's work on aircraft survivability**. - 1984. Disponível em : <[https://www.researchgate.net/publication/254286514\\_Abraham\\_Wald's\\_Work\\_on\\_Aircraft\\_Survivability](https://www.researchgate.net/publication/254286514_Abraham_Wald's_Work_on_Aircraft_Survivability)> Acessado em 05 mai. 2021.

MAYER-SCHONBERGER, V.; CUKIER, K. **Big Data: A Revolution That Will Transform How We Live, Work, and Think** - 2013.

MANNING, Christopher D.; RAGHAVAN, Prabhakar; SHUTZE, Hinrich; **Introduction to information retrieval**. 01. ed. Cambridge: Cambridge University Press, 2008. p. 32.

MORAIS, C. M. M. D. **Proposição de indicadores para investigação de licitações por meio de técnicas de reconhecimento de padrões estatísticos e mineração de dados** 2016. xx, 126 f., il. Universidade de Brasília, Brasília, 2016.

PANNIELLO, Umberto; GORGOGNONE, Michele, **Incorporating context into recommender systems: an empirical comparison of context-based approaches**. **Electronic Commerce Research**. v.12, n.1, p-30, mar/. 2012.

ROUSSEEUW, Peter J. **Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis**. *Computational and Applied Mathematics*. 20: 53–65, 1987.

STEPHENSON-LAWS, J.; FOUNDER, J.D. - **Life is Full of Decisions, But Don't Make Them When You're Hungry!** - 2019. Disponível em: <<https://www.phlabs.com/life-is-full-of-decisions-but-dont-make-them-when-youre-hungry>> Acessado em 10 jun. 2021.

SAMUEL, Arthur. **Some Studies in Machine Learning Using the Game of Checkers.** IBM Journal of Research and Development. California, v.3, n.1, p.210, 1959.

TRAN, T.; COHEN, R. **Hybrid Recommender Systems for Electronic Commerce - 2000**  
Disponível em:

<<https://www.aaai.org/Papers/Workshops/2000/WS-00-04/WS00-04-012.pdf>> Acessado em 20 mai. 2021.