UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Sistema de Recomendação de Pratos por Avaliação

Fabiano Vieira de Souza

Trabalho de Conclusão de Curso
MBA em Inteligência Artificial e Big Data

ICMC USP SÃO CARLOS

UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Sistema de Recomendação de Pratos por Avaliação

Fabiano Vieira de Souza

USP - São Carlos
2023

Fabiano Vieira de Souza

Sistema Para Recomendação de Pratos por Avaliação

Trabalho de conclusão de curso apresentado ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientador: Marcelo G. Manzato

USP - São Carlos

2023

RESUMO

VIEIRA DE SOUZA, FABIANO. P. **Sistema Para Recomendação de Pratos por Avaliação**. 2023. 52 f. Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2020.

Este trabalho aborda a integração entre inteligência artificial (IA) e big data para aprimorar sistemas de recomendação de pratos com base em avaliações. Utilizando técnicas avançadas de pré-processamento de dados. A pesquisa visou otimizar a qualidade das informações coletadas, aprimorando a eficácia do sistema proposto. A implementação do algoritmo K-Nearest Neighbors (KNN) foi explorada para analisar padrões nas avaliações dos usuários, gerando recomendações mais precisas e personalizadas. Além disso, o método Silhouette foi empregado para a validação interna do modelo, contribuindo para a identificação do número ideal de clusters e, assim, otimizando a segmentação dos dados.

A convergência destas técnicas de IA com a capacidade de processamento robusto do big data resulta em um sistema de recomendação de pratos mais sofisticado. Ao enfatizar o pré-processamento de dados, o uso do KNN e a validação interna através do método Silhouette, este trabalho não destaca apenas a eficácia da IA na análise de grandes conjuntos de dados, mas também ressalta a importância do pré-processamento e da validação para sistemas de recomendação mais precisos e confiáveis. Essa abordagem promissora apresenta contribuições significativas para a interseção entre inteligência artificial, big data e aprimoramento de sistemas de recomendação alimentar.

Palavras-chave: Sistema de Recomendações, método Silhouette, big data;

ABSTRACT

VIEIRA DE SOUZA, FABIANO. P. **System for Recommending Dishes by Evaluation**. 2020. 52 f. Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2020.

This work addresses the integration between artificial intelligence (AI) and big data to improve food recommendation systems based on reviews. Using advanced data pre-processing techniques, the research aimed to optimize the quality of the information collected, improving the effectiveness of the proposed system. The implementation of the K-Nearest Neighbors (KNN) algorithm was explored to analyze patterns in user reviews, generating more accurate and personalized recommendations. Furthermore, the Silhouette method was used for internal validation of the model, contributing to the identification of the ideal number of clusters and, thus, optimizing data segmentation.

The convergence of these AI techniques with the robust processing capacity of big data results in a more sophisticated food recommendation system. By emphasizing data pre-processing, the use of KNN and internal validation through the Silhouette method, this work not only highlights the effectiveness of AI in analyzing large data sets, but also highlights the importance of pre-processing and validation for more accurate and reliable recommendation systems. This promising approach makes significant contributions to the intersection between artificial intelligence, big data and the improvement of food recommendation systems.

Keywords: Recommendations System, method Silhouette, big data;

SUMÁRIO

1 INTRODUÇÃO

No âmbito empresarial, a contextualização em *Big Data* tornou-se essencial para compreender e explorar o potencial estratégico dos dados. Empresas de diferentes setores buscam extrair *insights* valiosos a partir da análise de grandes conjuntos de dados, visando aprimorar a tomada de decisões, compreender padrões de comportamento do consumidor e identificar oportunidades de inovação.

No entanto, Big Data não se resume apenas às possibilidades de vantagens estratégicas. A coleta e análise maciça de dados também levantam questões éticas e de privacidade que não podem ser ignoradas. Questões sobre transparência, consentimento e segurança dos dados emergem como preocupações cruciais, demandando uma abordagem ética e responsável na utilização dessas informações.

Quanto a evolução dos sistemas de recomendação abrange desde abordagens tradicionais, como filtragem colaborativa e filtragem baseada em conteúdo, até métodos mais avançados, como aprendizado profundo (*deep learning*) e técnicas híbridas. Essa diversidade reflete a constante busca por estratégias mais eficazes na identificação de padrões complexos e na entrega de recomendações cada vez mais precisas.

Entretanto, a contextualização em sistemas de recomendação não está isenta de desafios, questões éticas, como a privacidade do usuário e a justiça algorítmica, surgem como preocupações críticas que exigem abordagens equilibradas. Além disso, a necessidade de transparência nos algoritmos torna-se vital para construir a confiança do usuário e garantir que as recomendações sejam percebidas como relevantes e confiáveis.

Diante desse cenário, este projeto tem como finalidade apresentar um estudo na área sistemas de recomendação com a analise da performance de duas técnicas muito utilizadas em plataformas de recomendação que utilizam a filtragem colaborativa, apto a fazer recomendação de pratos com base em avaliações. Nesse caso, o sistema é alimentado com informações de histórico de compras e avaliações de usuários. A partir dessas informações, o sistema tem a capacidade de reconhecer padrões e preferências compartilhadas entre os usuários, possibilitando assim a formulação de sugestões personalizadas.

Por exemplo, imagine um site de delivery de comida. O sistema pode coletar informações sobre os pratos que os usuários já pediram anteriormente, e utilizar essas informações para fazer sugestões de pratos que são populares entre usuários que possuem um histórico de compras similar.

Além disso, o sistema também pode levar em conta outras informações, como as avaliações dos usuários e a popularidade geral de cada prato. Com o tempo, o sistema pode aprimorar suas sugestões de acordo com o feedback dos usuários, tornando as recomendações ainda mais precisas e personalizadas. Essa abordagem pode ajudar a melhorar a experiência do usuário, aumentar a satisfação e fidelidade dos clientes, e gerar mais receita financeira para empresa.

A organização deste documento segue uma abordagem sequencial, começando pela exploração das áreas de *machine learning* e análise de dados. Essa primeira etapa visa proporcionar uma compreensão abrangente do processo de extração de características de um objeto a ser classificado. Em seguida, serão abordados os métodos utilizados para recomendar dados previamente classificados, destacando as técnicas empregadas nesse contexto.

Por último, o foco será direcionado para o desenvolvimento do sistema de recomendações de pratos gastronômicos, realizado na linguagem de programação Python. Este segmento fornecerá insights sobre a implementação prática das teorias e metodologias exploradas anteriormente, consolidando os conhecimentos adquiridos ao longo do documento.

2 CONCEITOS FUNDAMENTAIS

Neste capitulo vamos explorar os conceitos fundamentais do aprendizado de máquina aplicado a recomendações, para compreender como algoritmos avançados interpretam e utilizam informações, resultando em experiências de usuário mais personalizadas e eficazes. Neste contexto, a introdução aos conceitos essenciais do aprendizado de máquina para recomendações é um passo significativo para desvendar os mecanismos subjacentes a essa tecnologia inovadora.

2.1 Ciência de dados

A ciência de dados emerge como a peça fundamental para o futuro da inteligência artificial, capacitando a transformação de conceitos antes vistos apenas em filmes de ficção científica de Hollywood em realidade. O advento da era da big data trouxe consigo uma crescente demanda por armazenamento de dados, tornando-se a principal preocupação para os setores empresariais.

Inicialmente, o foco estava na criação de estruturas e soluções eficazes para o armazenamento de dados. No entanto, com o sucesso na resolução desse problema, a atenção

passou a se concentrar no processamento eficiente desses dados. Assim, torna-se crucial compreender o que é ciência de dados e como ela opera.

A ciência de dados dedica-se à extração de informações dos dados, proporcionando embasamento para tomada de decisões nos negócios, planejamento estratégico e diversas outras aplicações. Este campo envolve a aplicação de ferramentas analíticas avançadas e princípios científicos.

Recorrendo a técnicas estatísticas e de ciência da computação, ela examina e dá sentido a conjuntos de dados grandes e complexos, permitindo a tomada de decisões informadas.

Compreender a ciência de dados torna-se crucial para as empresas, uma vez que pode contribuir significativamente para a melhoria das estratégias de marketing e vendas, identificação de novas oportunidades de negócios e aumento da eficiência operacional. Esse conhecimento pode traduzir-se em vantagens competitivas sobre outras empresas, consolidando a ciência de dados como uma disciplina que integra diversos campos acadêmicos como:

- Engenharia De Dados
- Preparação De Dados
- Mineração De Dados
- Análise Preditiva
- Aprendizado De Máquina
- Visualização De Dados
- Programação De Software
- Matemática
- Estatística

2.1.1 Importância da ciência de dados

Atualmente, as organizações estão com uma vasta quantidade de dados. Ao integrar diversas técnicas, tecnologias e ferramentas, a ciência de dados desempenha um papel crucial na obtenção de conclusões esclarecedoras.

Empresas lidam com volumes significativos de dados em setores como comércio eletrônico, finanças, medicina, recursos humanos, entre outros. Esses dados são processados por meio de tecnologias e métodos específicos da ciência de dados.

A ciência de dados capacita as empresas a tomar decisões informadas com base em dados.

Essa disciplina é essencial para o crescimento e a tomada de decisões nas organizações, apresentando benefícios como:

1. Análise de dados do cliente para identificar padrões e tendências, aprimorando a experiência do cliente.
2. Avaliação de dados operacionais para aumentar a eficiência e reduzir custos.
3. Análise de dados para fornecer insights que aprimorem a tomada de decisões organizacionais.
4. Identificação de novas oportunidades e desenvolvimento de produtos e serviços inovadores.
5. Detecção e prevenção de ataques cibernéticos por meio do estudo de dados e identificação de padrões

2.2 Aprendizagem de máquina

A aprendizado de máquina, ou "*machine learning*" em inglês, é um subcampo da inteligência artificial que se concentra no desenvolvimento de algoritmos e modelos que permitem que os computadores aprendam e tomem decisões com base em dados. Isso permite que ao invés de serem explicitamente programados para realizar tarefas específicas, os sistemas de aprendizado de máquina sejam projetados para aprender e melhorar com a experiência, de forma autônoma.

2.2.1 Aprendizado supervisionado

O aprendizado supervisionado representa um dos paradigmas fundamentais em aprendizado de máquina, no qual um algoritmo é instruído por meio de um conjunto de dados contendo exemplos previamente rotulados. Em termos mais simples, cada instância no conjunto de treinamento consiste em um conjunto de características (variáveis) acompanhado por um rótulo (etiqueta) que indica a saída desejada..

A meta do aprendizado supervisionado é internalizar uma função capaz de mapear as características de entrada para as saídas rotuladas, permitindo que o modelo realize previsões precisas em dados não rotulados. Esse enfoque visa capacitar o algoritmo a generalizar seu entendimento a partir dos exemplos conhecidos, de forma a aplicar esse conhecimento a novos conjuntos de dados ainda não observados.

Aqui estão os principais componentes do aprendizado supervisionado:

1 - Conjunto de Dados de Treinamento: Consiste em uma coleção de exemplos, cada um com características de entrada e seus rótulos correspondentes. Esses exemplos são usados para treinar o modelo.

2 - Modelo de Aprendizado: É o algoritmo ou a estrutura que o aprendizado supervisionado utiliza para aprender a relação entre as características de entrada e os rótulos de saída. O objetivo é encontrar um modelo que generalize da melhor forma para os novos dados.

3 - Função Objetivo: Esta função mede o quão bem o modelo está realizando em relação aos rótulos reais no conjunto de treinamento. O objetivo é ajustar o modelo de forma a minimizar o erro entre as previsões e os rótulos reais.

4 - Conjunto de Dados de Teste: Após treinar o modelo, você o testa em um conjunto de dados de teste separado, que não foi visto durante o treinamento. Isso permite avaliar a eficiência/eficaz do modelo generaliza para novos dados.

O aprendizado supervisionado é amplamente utilizado em uma variedade de aplicações, incluindo classificação (onde o objetivo é prever uma categoria ou classe), regressão (onde o objetivo é prever um valor numérico), detecção de anomalias, tradução automática, reconhecimento de voz, visão computacional, entre outros.

Exemplos de algoritmos de aprendizado supervisionado incluem regressão linear, regressão logística, árvores de decisão, redes neurais, máquinas de vetores de suporte (SVM), entre outros. O sucesso do aprendizado supervisionado depende da qualidade do conjunto de dados de treinamento e da escolha apropriada do modelo e dos hiper parâmetros.

2.2.2 Aprendizado não supervisionado

O aprendizado não supervisionado constitui um modelo essencial no campo de aprendizado de máquina, caracterizado pelo treinamento de um algoritmo em um conjunto de dados desprovido de supervisão explícita ou rótulos de classe.

Em contraste com o aprendizado supervisionado, no qual um modelo é instruído com exemplos rotulados para realizar previsões ou classificações, o aprendizado não supervisionado destaca-se pela capacidade de extrair informações, padrões e estruturas subjacentes dos dados sem orientação prévia. Nesse paradigma, o algoritmo é desafiado a descobrir de forma autônoma as relações e características intrínsecas presentes nos dados, proporcionando insights valiosos sem a necessidade de guia explícito.

Existem duas tarefas comuns no aprendizado não supervisionado:

1-Agrupamento (Clustering): A tarefa principal do agrupamento é dividir os dados em grupos ou clusters de acordo com a similaridade entre os elementos. Em outras palavras, o objetivo é encontrar estruturas intrínsecas nos dados, onde os itens dentro do mesmo grupo são semelhantes entre si, do que com itens de outros grupos. O algoritmo de k-médias é um exemplo comum de algoritmo de agrupamento.

2-Redução de Dimensionalidade: Nesse caso, o objetivo é reduzir a complexidade dos dados, mantendo as informações mais importantes. Muitas vezes, isso envolve a projeção dos dados de um espaço dimensional mais alto para um espaço dimensional mais baixo. A Análise de Componentes Principais (PCA) é um exemplo de técnica de redução de dimensionalidade.

O aprendizado não supervisionado é frequentemente usado em situações em que os dados não possuem rótulos ou quando o objetivo é explorar a estrutura subjacente dos dados, descobrir padrões desconhecidos ou reduzir a dimensionalidade para visualização ou simplificação.

Além disso, o aprendizado não supervisionado é útil no pré-processamento de dados para tarefas de aprendizado supervisionado, onde a redução de dimensionalidade ou a identificação de grupos de dados podem melhorar o desempenho do modelo.

É importante mencionar que, devido à falta de rótulos, o aprendizado não supervisionado é muitas vezes mais desafiador e subjetivo, uma vez que a interpretação dos resultados depende do domínio do problema e dos objetivos do analista de dados.

2.3 Técnica de recomendação

Métodos de sugestão, comumente referidos como sistemas de recomendação, compreendem uma variedade de técnicas e algoritmos empregados para oferecer sugestões de itens (produtos, serviços, conteúdo etc.) aos usuários, levando em consideração suas preferências, histórico de interações ou características individuais. Essas técnicas desempenham um papel significativo em diversas aplicações, como comércio eletrônico, plataformas de streaming, mídias sociais, sistemas de informação, entre outros. A gama de técnicas de recomendação é vasta, abrangendo diferentes abordagens para fornecer sugestões personalizadas e relevantes aos usuários, como:

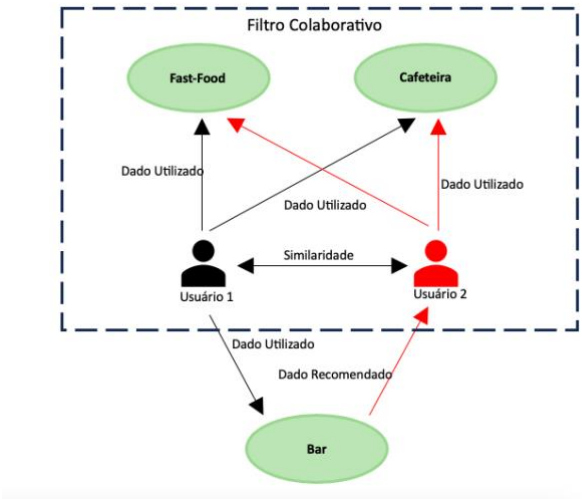
- Filtragem Colaborativa;
- Recomendação Baseada em Conteúdo;
- Recomendação Híbrida;
- Recomendação Contextual.

Essas técnicas de recomendação são essenciais em serviços online e plataformas que lidam com grandes volumes de informações, permitindo aos usuários descobrir novos itens de interesse e melhorando a experiência deles. Eles são alimentados por algoritmos de aprendizado de máquina, análise de dados e processamento de informações para fornecer recomendações personalizadas.

2.3.1 Filtragem Colaborativa

A abordagem de recomendação por filtragem colaborativa se concentra em grupos de usuários. Nessa técnica, todos os dados dos usuários na rede são comparados, utilizando a semelhança entre os dados como critério, conforme ilustrado na figura 1. A recomendação é feita com base nos dados que distinguem os usuários entre si, sendo eficaz em diversas aplicações. Essencialmente, essa técnica identifica padrões de preferências entre grupos de usuários e sugere itens com base nessas diferenças percebidas, oferecendo recomendações personalizadas.

Figura 1: Filtro colaborativo



Fonte: Autoria própria. (2023)

No processo de recomendação por filtro colaborativo, a busca é direcionada a usuários que apresentam padrões de comportamento semelhantes. A recomendação é então elaborada a partir desses dados, sugerindo itens ainda não explorados pelo usuário em questão. Essa sugestão é refinada ao cruzar as preferências desse usuário com as de outros que compartilham um perfil semelhante. Essa abordagem visa evitar que o usuário fique restrito a um conjunto limitado de conteúdos, promovendo variedade. Contudo, é crucial encontrar um equilíbrio para

não diversificar excessivamente, o que poderia resultar em recomendações que não se alinham aos interesses do usuário. Portanto, é necessário realizar uma medida cuidadosa para garantir que o conteúdo sugerido seja diversificado o suficiente, mas ainda assim relevante para o usuário.

2.3.2 Recomendação baseada em conteúdo

As sugestões são formuladas considerando a correlação entre o conteúdo dos itens e as preferências do usuário, em vez de depender das opiniões explícitas dos usuários sobre os itens. Os itens podem ser caracterizados por diversos atributos, como gênero, diretor, ano, faixa etária, título, descrição, entre outros, especialmente no contexto de filmes..

Por exemplo, se um usuário assiste predominantemente filmes de terror, é vantajoso que um Sistema de Recomendação sugira lançamentos do mesmo gênero, pois o terror é claramente a preferência desse usuário. Outra situação ilustrativa seria um usuário que compra vários livros do mesmo autor, indicando uma inclinação positiva por esse autor e, por extensão, por outros livros escritos por ele.

O sistema aprende a fazer recomendações com base no perfil do usuário, podendo esse perfil ser construído a partir dos itens que o usuário apreciou no passado ou de atributos predefinidos em seu perfil, como preferências por gênero ou faixa etária. A similaridade entre usuários é determinada a partir do feedback, tanto implícito quanto explícito, fornecido pelos usuários.

2.3.3 Recomendação Híbrida

Recomendação híbrida é uma abordagem em sistemas de recomendação que combina múltiplas técnicas ou fontes de informação para fornecer sugestões personalizadas aos usuários. Essa abordagem é projetada para superar as limitações das abordagens puramente baseadas em conteúdo ou filtragem colaborativa, oferecendo recomendações mais precisas e abrangentes.

Existem dois principais tipos de recomendação híbrida:

Recomendação Híbrida por Fusão: Nesse tipo de recomendação híbrida, as recomendações de diferentes abordagens (por exemplo, baseadas em conteúdo e filtragem colaborativa) são geradas independentemente e, em seguida, combinadas para criar uma lista final de recomendações. A combinação pode ser ponderada, onde as recomendações de

diferentes técnicas têm pesos diferentes, ou simplesmente uma concatenação das recomendações.

Recomendação Híbrida por Comutação: Nesse tipo de recomendação híbrida, o sistema pode alternar entre diferentes abordagens com base no contexto ou nas características do usuário. Por exemplo, ele pode usar filtragem colaborativa quando há informações suficientes sobre o comportamento do usuário e, em seguida, alternar para recomendação baseada em conteúdo quando as informações do usuário são limitadas.

A recomendação híbrida oferece várias vantagens:

- Reduz a dependência de uma única técnica, tornando as recomendações mais robustas e adaptáveis.
- Pode fornecer sugestões mais precisas, aproveitando o melhor de diferentes abordagens.
- Lida com o problema de arranque a frio (quando um sistema não tem informações iniciais sobre um novo usuário ou item) de forma mais eficaz.
- Pode compensar as deficiências de uma técnica com os pontos fortes de outra.

A implementação de recomendações híbridas pode ser desafiadora, pois envolve a integração de diferentes técnicas de recomendação e a consideração de como combinar ou alternar entre elas. No entanto, essas abordagens são amplamente utilizadas em serviços de recomendação em diversos domínios, como comércio eletrônico, streaming de conteúdo, redes sociais e muito mais, para fornecer recomendações mais personalizadas e relevantes aos usuários.

2.3.4 Recomendação sensível ao contexto

A recomendação sensível ao contexto refere-se a uma abordagem em sistemas de recomendação que leva em consideração o contexto ou situação específica em que um usuário está inserido ao fornecer sugestões personalizadas. Em vez de depender apenas do histórico de interações passadas do usuário, esse método considera fatores ambientais, temporais ou situacionais para ajustar as recomendações.

O contexto pode incluir informações como localização geográfica, dispositivo utilizado, horário do dia, atividades recentes do usuário, entre outros elementos que possam influenciar as preferências do usuário naquele momento específico. Dessa forma, a recomendação sensível ao contexto busca oferecer sugestões mais relevantes e adaptadas às circunstâncias imediatas do usuário, proporcionando uma experiência mais personalizada e eficaz.

Alguns dos fatores de contexto considerados podem incluir:

1. Localização Geográfica: A recomendação pode variar com base na localização do usuário. Por exemplo, em um aplicativo de restaurantes, as sugestões podem ser adaptadas de acordo com os restaurantes nas proximidades.
2. Dispositivo Utilizado: As sugestões podem ser ajustadas com base no tipo de dispositivo que o usuário está usando. Por exemplo, as recomendações em um aplicativo móvel podem diferir daquelas em uma plataforma web.
3. Horário do Dia: Dependendo do horário, certos tipos de conteúdo ou atividades podem ser mais apropriados. Por exemplo, sugestões de músicas pela manhã podem ser diferentes das sugestões à noite.

A ideia central é adaptar as sugestões não apenas com base no histórico individual do usuário, mas também nas condições específicas em que o usuário se encontra no momento da recomendação. Isso visa proporcionar uma experiência mais personalizada, levando em consideração o contexto em constante mudança da vida do usuário.

2.4 Pré-processamento

O pré-processamento refere-se a uma etapa crucial que antecede a aplicação de algoritmos de aprendizado. Essa fase envolve a preparação e limpeza dos dados brutos, com o objetivo de torná-los mais adequados e compreensíveis para os modelos de *machine learning*.

O processo inclui atividades como tratamento de valores ausentes, normalização de dados, codificação de variáveis categóricas, remoção de outliers, entre outras técnicas. O pré-processamento desempenha um papel fundamental na qualidade e desempenho dos modelos, pois influencia diretamente a capacidade do algoritmo de identificar padrões significativos nos dados. Uma abordagem cuidadosa no pré-processamento contribui para resultados mais precisos e robustos durante as fases subsequentes de treinamento e avaliação do modelo..

2.5 K-Means

O método de clusterização denominado algoritmo K-Means realiza a divisão de um conjunto de N itens em K subconjuntos distintos. Cada um desses subconjuntos é composto por objetos semelhantes, determinados através da análise da medida de distância entre eles. Essa técnica é aplicável a uma variedade de contextos e pode ser utilizada para resolver diferentes tipos de problemas:

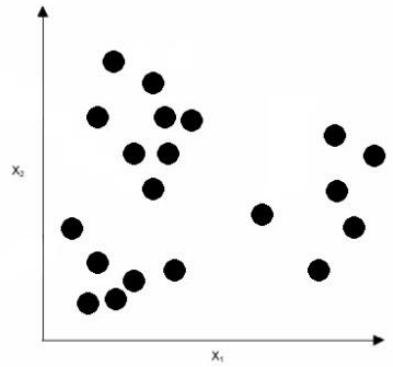
- No âmbito do marketing, a aplicação do algoritmo K-Means pode ser observada na identificação de distintos grupos de clientes.
- Na geografia, o algoritmo é empregado para identificar áreas que compartilham propósitos similares, utilizando observações da terra.
- No setor de seguros, a técnica é utilizada para identificar grupos de clientes que frequentemente fazem comunicação de sinistros.
- Em planejamento, o K-Means é aplicado na identificação de grupos de casas com base em critérios como tipo, valor e localização geográfica..
- Na área da saúde, o algoritmo é utilizado para agrupar pacientes que apresentam sintomas semelhantes.
- Na indústria cinematográfica, a técnica é aplicada para identificar grupos de usuários que compartilham interesses específicos, como preferências por gêneros ou filmes particulares.

Cada agrupamento formado é caracterizado pelos membros que o compõem e pelo seu ponto central, conhecido como centroide. Para operar, o algoritmo necessita de três dados como entrada:

- O conjunto de instancias (amostras)
- Métrica de distância para calcular a distância entre os vizinhos (amostras)
- O valor de K que é o número de centroides ou clusters a serem formados

De maneira geral, para identificar os agrupamentos, o algoritmo realiza uma série de passos, conforme delineado por (OLIVEIRA, 2002) e como apresentado na figura 3:

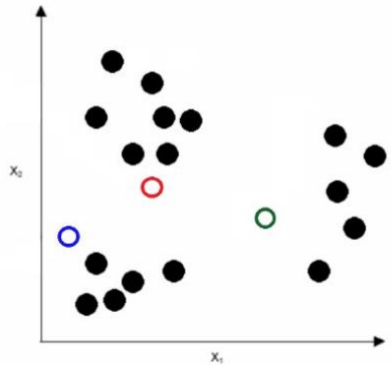
Figura 3: Conjunto de N itens, algoritmo ainda não executado



Fonte: Autoria própria.

Passo 1: Escolher de forma arbitrária K elementos para atuarem como as sementes (ou centroides) dos clusters iniciais, como ilustrado na figura 4:

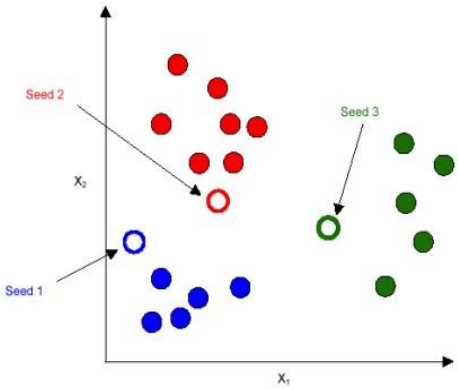
Figura 4: K itens selecionados para servirem como sementes



Fonte: Autoria própria.

Passo 2: Atribuir cada item ao centro do cluster mais próximo (com maior similaridade), formando grupos, conforme demonstrado na figura 5.

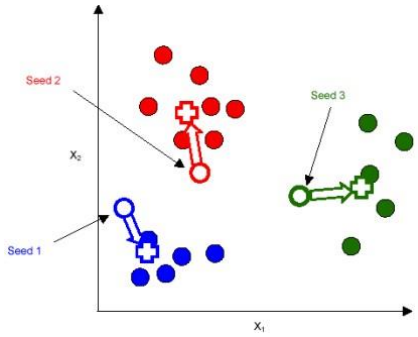
Figura 5: formação inicial de grupos



Fonte: Autoria própria.

Passo 3: Calcular os novos centroides a partir de todos os elementos existentes em cada cluster como vemos na figura 6:

Figura 6: Recálculo dos centroides



Fonte: Autoria própria.

Passo 4: Associar cada registro aos novos Centroides.

Passo 5: Repita a partir do passo 2 até que não ocorra mais mudanças nos itens que compõe o cluster

De acordo com RICCI (2010), a maior parte da convergência acontece nas primeiras iterações do algoritmo, assim, a condição de parada pode ser ajustada para "até que relativamente poucos pontos mudem de cluster", aprimorando a eficiência do algoritmo. É preferível que as distâncias entre os vetores (objetos) dentro de um grupo (cluster) sejam reduzidas, enquanto as distâncias entre diferentes clusters sejam ampliadas.

2.5.1 Seleção apropriada da quantidade de clusters.

A definição do número de agrupamentos é uma prática comum em análises de clusters, e neste projeto, adotei o método Silhouette. O Silhouette é uma métrica interna de validação que avalia a coesão e a separação dos agrupamentos em relação aos dados. O objetivo é identificar o número ideal de agrupamentos que otimize a similaridade dentro de cada agrupamento e minimize a similaridade entre agrupamentos diferentes.

O método Silhouette calcula um índice para cada ponto de dados com base em seu próprio agrupamento e nos agrupamentos vizinhos mais próximos. Os valores do Silhouette variam de -1 a 1, sendo que valores mais altos indicam uma separação mais eficaz entre os agrupamentos. Geralmente, o número ideal de agrupamentos é aquele que maximiza a média dos valores do Silhouette para todos os pontos de dados.

Para determinar o número ideal de clusters usando o método Silhouette, foi realizado o seguinte procedimento:

1. Aplicado o algoritmo de clustering (k-means) para diferentes valores de k (número de clusters).

2. Para cada resultado de clustering, foi calculado o valor médio do Silhouette para todos os pontos de dados.
3. Identificado o valor de k que maximiza a média do Silhouette.

Essa abordagem auxilia na identificação de uma configuração de agrupamento que se adequa de maneira ótima aos dados, levando em conta tanto a coesão interna quanto a separação entre os clusters. Embora o método Silhouette seja bastante empregado, é aconselhável avaliá-lo em conjunto com outras técnicas de validação de agrupamento para uma análise mais abrangente e robusta.

2.6 K-Nearest Neighbor

O K-Nearest Neighbors (K-NN), ou K-Vizinhos Mais Próximos em português, é um algoritmo de aprendizado de máquina supervisionado utilizado tanto para classificação quanto para regressão. No K-NN, um objeto é classificado pela maioria dos votos de seus vizinhos mais próximos. A "K" em K-NN representa o número de vizinhos mais próximos que serão considerados ao tomar uma decisão de classificação ou previsão.

O algoritmo funciona calculando a distância entre o ponto de dados que está sendo classificado ou previsto e todos os outros pontos de dados no conjunto de treinamento. Em seguida, identifica os "K" pontos mais próximos a esse ponto e determina a classe (no caso de classificação) ou realiza uma média (no caso de regressão) para obter a previsão final..

Este método tem sido usado em diversas aplicações no campo de data-mining, métodos estatísticos, reconhecimento de padrões e processamento de imagens.

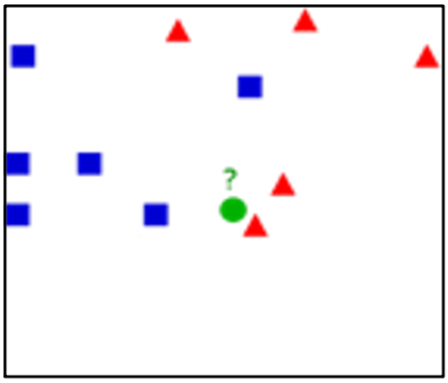
O algoritmo requer três dados como entrada como são apresentados na figura 7:

• O conjunto de instancias (amostras)

• Métrica de distância para calcular a distância entre os vizinhos (amostras)

• O valor de K, o número de vizinhos mais próximos a serem recuperados Em geral, para classificar um item, será executado dois passos

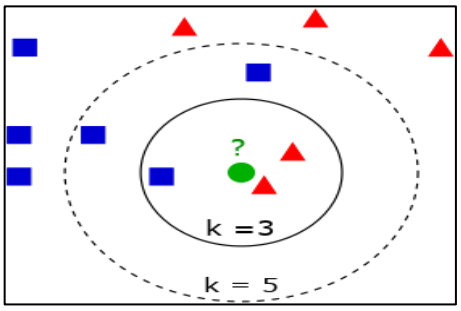
Figura 7: Conjuntos de N itens



Fonte: Autoria própria. (2021)

Passo 1: Computar a distância entre um item alvo até outros itens do treinamento Passo 2: Identificar os k vizinhos mais próximos do item alvo como apresentado na figura 8:

Figura 8: Exemplo de classificação KNN



Fonte: Autoria própria.

Na Figura (10), a amostra de teste é o círculo verde, que deve ser classificada. Se k=3 compreendido pelo círculo de linha continua, o objeto verde é mais semelhante ao triangulo do que ao quadrado, porque há dois triângulos e apenas um quadrado dentro do círculo interno.

O conceito do modelo KNN é bastante intuitivo, exigindo apenas uma compreensão básica de distância. Quanto mais próximos os pontos estão, maior é a similaridade entre eles..

3 TENDÊNCIAS INOVADORAS EM SISTEMAS DE RECOMENDAÇÃO

Ao analisar propostas preeminentes no campo de sistemas de recomendação, destacamos contribuições significativas de outros autores, como [Adeniyi, D. A.] que explorou a abordagens de *clustering* e k-Nearest Neighbors (KNN) em um projeto de automatização de mineração e recomendação de dados de uso da web. Em suas pesquisas, [Adeniyi, D. A.] relata

um resultado bem-sucedido e transparente na classificação do k-Nearest Neighbors (KNN) com uma maneira simples de entender, com alta tendência a possuir qualidades desejáveis e bem fácil de implementar do que a maioria das outras técnicas de aprendizado de máquina disponíveis nos dias de hoje.

Já [H.R. Koosha] propôs um novo sistema de recomendação em duas etapas baseado nos dados demográficos e classificações de usuários nos conjuntos de dados públicos do MovieLens. Na primeira etapa, o clustering no conjunto de dados de treinamento é realizado com base nos dados demográficos, agrupando os clientes em grupos homogêneos aglomerados. No próximo passo, para cada cluster, são propostos dois sistemas de recomendação baseado em K-Nearest Neighbours (KNN) e Naïve Bayesian Classificação. Os resultados obtidos são avaliados com base em muitas medidas internas e externas como o índice Davies-Bouldin, precisão, exatidão, recall e medida F. Os resultados obtidos mostram a eficácia do K-means/FA/KNN em comparação com os outros modelos existentes.

Ao incorporar essas abordagens, esta pesquisa busca não apenas estende esses conhecimentos estabelecidos, mas também explorar novas sinergias entre clustering e KNN, visando aprimorar a qualidade das recomendações e proporcionar uma experiência mais individualizada aos usuários. A convergência desses insights contribuirá para a evolução contínua dos sistemas de recomendação, promovendo uma compreensão mais profunda e eficaz das preferências dos usuários.

4 RECOMENDAÇÃO DE PRATOS POR AVALIAÇÃO

A abordagem científica empregada na concepção deste projeto fundamentou-se na observação do funcionamento dos sistemas de recomendação. O processo envolveu a análise minuciosa das técnicas apresentadas em artigos acadêmicos, a seleção de um domínio específico para a implementação de um caso de estudo experimental e, por fim, a análise criteriosa dos resultados obtidos pelo modelo desenvolvido.

Este segmento destaca o sistema desenvolvido para fornecer recomendações de pratos com base nas avaliações dos consumidores, a recomendação combinada de conteúdo e colaboração demonstra eficácia, complementando-se mutuamente. A filtragem colaborativa compara as avaliações do usuário com as de outros consumidores, sugerindo pratos com maior avaliação.

O desenvolvimento do sistema abrange a inserção e análise estatística e exploratória do dataset, manipulação dos dados, avaliação do modelo e visualização de resultados . A utilização

da tabela ajustada e o emprego de algoritmos de aprendizado de máquina são centrais para avaliar a clusterização das avaliações e aprimorar a precisão das recomendações.

4.1 Tecnologias Utilizadas

Para o desenvolvimento do protótipo foi utilizado a linguagem de programação Python que fornece uma série de ferramentas para o desenvolvimento de um sistema de recomendação.

O *Jupyter* Notebook, que faz parte do distribuidor de linguagem Anaconda, foi o ambiente de desenvolvimento escolhido. A plataforma Anaconda já inclui várias bibliotecas essenciais, proporcionando um ambiente completo para a criação de aplicações de ciência de dados. O *Jupyter* Notebook é a ferramenta padrão nesse contexto, oferecendo não apenas um ambiente de desenvolvimento organizado, mas também facilitando a documentação do processo de desenvolvimento de maneira acessível e direta.

Neste projeto como mostra na Figura 9, foram empregadas as bibliotecas pandas, *numpy*, *matplotlib* e *sklearn*. Cada uma dessas bibliotecas desempenha funções fundamentais. A utilização da biblioteca pandas é fundamental para realizar análise e manipulação de dados, desempenhando um papel crucial nessas tarefas. Em conjunto, a biblioteca numpy mostrou eficiência ao manipular e reestruturar tabelas, além de realizar operações matemáticas matriciais nos dados. A principal contribuição da biblioteca *sklearn* foi fornecer os algoritmos fundamentais para o aprendizado de máquina. Por fim, a biblioteca *matplotlib* destacou-se como uma ferramenta valiosa para visualizar graficamente os resultados matemáticos derivados do aprendizado.

Figura 9: Bibliotecas

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import warnings
import random
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import silhouette_score
from sklearn.decomposition import PCA
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.utils.fixes import sklearn
from sklearn.metrics import silhouette_samples, silhouette_score
from scipy import sparse
from warnings import filterwarnings
from google.colab import drive
```

Fonte: Autoria própria

4.2 Pré-processamento dos Dados

O conjunto de dados presentes no *dataset* utilizado neste projeto estão disponíveis para *download* no site da kaggle.com e contém dados sobre todos os restaurantes da empresa de *delivery* iFood no período de 11 de novembro de 2020 a 27 de fevereiro de 2021. Na Figura 10 temos as estatísticas do *dataset*.

Figura 10: Dataset

```
#Estatísticas do Dataset#
O DataFrame tem 406399 linhas e 14 colunas.

A esparsidade dos dados é: 11.91%

Porcentagem de dados faltantes por coluna:
availableForScheduling    0.000000
avatar                    0.068160
category                  0.000000
delivery_fee              0.000000
delivery_time             0.000000
distance                  0.000000
ibge                      0.000000
minimumOrderValue         0.000000
name                      0.000000
paymentCodes              0.000492
price_range               0.000000
rating                    0.000000
tags                      0.000000
url                       0.000000
dtype: float64

Lista de Pratos únicos:
['Marmita' 'Açaí' 'Bebidas' 'Carnes' 'Brasileira' 'Lanches' 'Congelados'
 'Pastel' 'Indiana' 'Árabe' 'Doces & Bolos' 'Salgados' 'Mercado'
 'Italiana' 'Pizza' 'Típica do Norte' 'Hambúrguer' 'Coreana' 'Japonesa'
 'Frangos' 'Cafeteria' 'Sorvetes' 'Padaria' 'Variada' 'Peixes' 'Saudável'
 'Frutos Do Mar' 'Cozinha Rápida' 'Nordestina' 'Conveniência' 'Mexicana'
 'Portuguesa' 'Chinesa' 'Tapioca' 'Vegetariana' 'Africana' 'Mineira'
 'Sopas & Caldos' 'Argentina' 'Contemporânea' 'Mediterrânea' 'Vegana'
 'Peruana' 'Panqueca' 'Crepe' 'Yakisoba' 'Francesa' 'Alemã'
 'Congelados Fit' 'Espanhola' 'Baiana' 'Presentes' 'Asiática' 'Colombiana'
 'Casa de Sucos' 'Tailandesa' 'Gaúcha' 'Paranaense' 'Xis' 'Grega'
 'Marroquina']
```

Fonte: Autoria própria

4.2.1 Tratativa dos Dados

Como vemos na Figura 11 para um melhor processamento e organização dos dados foi feito uma tratativa dos mesmos mantendo apenas o atributo relevante para esse projeto que é o '**chosendish**' que contêm o nome dos pratos vendidos, foi incluído também um atributo chamado **"num_star"** onde foi adicionadas avaliações com notas de 1 a 10 para cada prato vendido do *dataset*.

Figura 11: Dataset tratado

```
dfr = df.rename(columns={'category': 'chosendish'})
num = [random.randint(1, 10) for _ in range(406399)]
dfr['num_star'] = num
remover = ['name','availableForScheduling','avatar','delivery_fee','delivery_time',
           'distance','ibge','minimumOrderValue','paymentCodes',
           'price_range','rating','tags','url' ]
dfrr = dfr.drop(columns=remover)
ordem = ['num_star', 'chosendish']
df = dfrr[ordem]
info = df.info()
print(info)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 406399 entries, 0 to 406398
Data columns (total 2 columns):
 #   Column      Non-Null Count   Dtype
---  ------      --------------   -----
 0   num_star    406399 non-null  int64
 1   chosendish  406399 non-null  object
dtypes: int64(1), object(1)
memory usage: 6.2+ MB
None
```

Fonte: Autoria própria

Continuando a tratativa dos dados, para que possamos processar a recomendação do prato foi feita uma média de todas as avaliações que estão no atributo **“num_star”** que foram dadas pelos clientes para cada prato deste período contido nesse *dataset*, na Figura 12 podemos ver uma amostra dessa atrativa.

Figura 12: Média das Avaliações

```
media_por_categoria = df.groupby('chosendish')['num_star'].mean().round(1)
df_new = pd.DataFrame({'chosendish': media_por_categoria.index,
                       'num_star': media_por_categoria.values})
df = pd.DataFrame(df_new)
df['num_star'].plot(kind='line', figsize=(8, 4), title='Avaliações')
plt.gca().spines[['top', 'right']].set_visible(False)
df
```

	chosendish	num_star
0	Africana	5.3
1	Alemã	5.7
2	Argentina	5.4
3	Asiática	5.6
4	Açaí	5.5
...	...	...
56	Vegana	5.7
57	Vegetariana	5.6
58	Xis	4.8
59	Yakisoba	5.3
60	Árabe	5.5

61 rows × 2 columns

Fonte: Autoria própria

4.2.2 Normalização dos dados

Um ponto importante para alguns algoritmos de *machine learning*, especialmente aquelas sensíveis à escala dos dados é normalizar os dados usando um *scaler* em Python onde refere-se em aplicar uma transformação aos dados para que fiquem em uma escala padrão ou normalizada. Isso é comumente feito para garantir que diferentes *features* ou variáveis em um conjunto de dados estejam na mesma escala, mesmo nosso projeto não precisando de uma normalização em um primeiro momento pois os dados da avaliação foram imputados manualmente a opção foi deixada para uma futura necessidade como vemos na Figura 13.

Figura 13: Normalização de dados

```
scaler = StandardScaler()
df['Avaliacao_Norm'] = scaler.fit_transform(df[['num_star']])
```

Fonte: Autoria própria

4.2.3 Número de Clusters

A determinação do número de clusters foi através do método Silhouette onde tivemos o resultado de 2 clusters como vemos a seguir na Figura 14.

Figura 14: Método Silhouette

```
max_clusters = 5
best_silhouette_score = -1
best_num_clusters = 2

for nk in range(2, max_clusters + 1):
    kmeans = KMeans(n_clusters=nk, random_state=0)
    cluster_labels = kmeans.fit_predict(df[['Avaliacao_Norm']])
    silhouette_avg = silhouette_score(df[['Avaliacao_Norm']], cluster_labels)

    if silhouette_avg > best_silhouette_score:
        best_silhouette_score = silhouette_avg
        best_num_clusters = nk

print(f"Melhor número de clusters com Silhouette: {best_num_clusters}")
```

```
Melhor número de clusters com Silhouette: 2
```

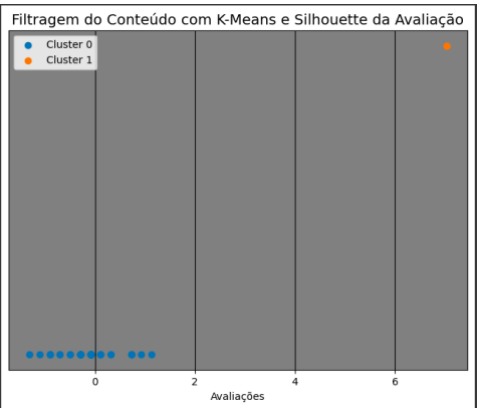
Fonte: Autoria própria

4.2.4 Plotagem do Número de Clusters

Foi realizado uma redução de dimensionalidade usando análise de componentes principais (PCA) na coluna com as avaliações, reduzindo os dados para uma única dimensão, em seguida, visualizamos os resultados usando um gráfico de dispersão, onde cada ponto representa um dado transformado pela PCA, colorido de acordo com o cluster original ao qual o dado pertence assim como mostra a Figura 15:

Figura 15: Plotagem do Número de Clusters

```
pca = PCA(n_components=1)
df_pca = pd.DataFrame(pca.fit_transform(df[['Avaliacao_Norm']]),
                      columns=['PC1'])
df_pca['Cluster'] = df['Cluster']
plt.figure(figsize=(8, 6))
for cluster in df_pca['Cluster'].unique():
    plt.scatter(df_pca[df_pca['Cluster'] == cluster]['PC1'],
                [cluster] * len(df_pca[df_pca['Cluster'] == cluster]),
                label=f'Cluster {cluster}')

plt.title('Filtragem do Conteúdo com K-Means e Silhouette da Avaliação',
          fontsize = 14, color = 'black')
plt.xlabel('Avaliações')
plt.yticks([])
plt.grid(color = 'black')
plt.legend()
ax = plt.gca()
ax.set_facecolor('grey')
plt.show()
```



Fonte: Autoria própria

A plotagem é útil para visualizar como os dados originais foram reduzidos a uma dimensão usando PCA e como esses dados transformados estão distribuídos em relação aos clusters originais.

4.3 Avaliação do Modelo

O primeiro ponto da avaliação do modelo é a divisão do conjunto de dados em treinamento e teste, para isso vamos usar a função train_test_split da biblioteca scikit-learn como mostra a Figura 16.

Figura 16: treinamento e teste

```
X_train, X_test, y_train, y_test = train_test_split(X, df['chosendish'], test_size=0.2, random_state=42)
```

Fonte: Autoria própria

Essa divisão é fundamental para treinar o modelo em uma parte do conjunto de dados e avaliá-lo em outra parte para verificar o desempenho do modelo em dados não vistos durante o treinamento.

Próximo passo é o treinamento do modelo K-Means com o melhor número de clusters, uma vez com o número de cluster podemos aplicar o método de clusterização K-Means para agrupamento dos pratos com base nas avaliações normalizadas como mostra a seguir a Figura 17.

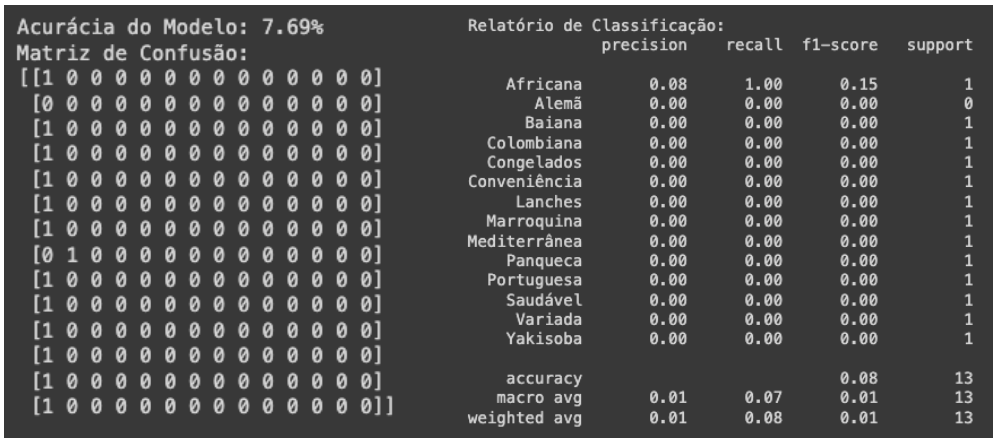
Figura 17: Código K-Means

```
kmeans = KMeans(n_clusters=best_num_clusters, random_state=0)
df['Cluster'] = kmeans.fit_predict(df[['Avaliacao_Norm']])
```

Fonte: Autoria própria

Na sequência temos a avaliação da acurácia do modelo usando métricas como matriz de confusão e relatório de classificação, essa parte do projeto é essencial para entender como o modelo K-Means está se saindo na tarefa de agrupar os dados de teste e comparar as previsões com as categorias reais, na Figura 18 temos o resultado dessa avaliação e classificação.

Figura 18: Métricas de Avaliação



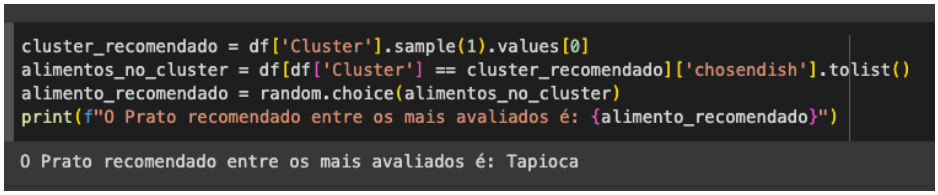
```
Acurácia do Modelo: 7.69%
Matriz de Confusão:
[[1 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 1 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 0]]

Relatório de Classificação:
                precision    recall  f1-score   support

     Africana       0.08      1.00      0.15         1
        Alemã       0.00      0.00      0.00         0
       Baiana       0.00      0.00      0.00         1
   Colombiana       0.00      0.00      0.00         1
   Congelados       0.00      0.00      0.00         1
Conveniência        0.00      0.00      0.00         1
      Lanches       0.00      0.00      0.00         1
    Marroquina      0.00      0.00      0.00         1
Mediterrânea        0.00      0.00      0.00         1
     Panqueca       0.00      0.00      0.00         1
   Portuguesa       0.00      0.00      0.00         1
     Saudável       0.00      0.00      0.00         1
      Variada       0.00      0.00      0.00         1
     Yakisoba       0.00      0.00      0.00         1

     accuracy                           0.08        13
    macro avg       0.01      0.07      0.01        13
weighted avg        0.01      0.08      0.01        13
```

Fonte: Autoria própria

4.4 Visualização de Resultados

A geração da recomendação do prato é feita com base nos clusters criados pelo modelo K-Means. Em resumo, o código seleciona aleatoriamente um cluster identificado pelo modelo K-Means, extrai a lista de alimentos pertencentes a esse cluster e, finalmente, escolhe aleatoriamente um alimento dessa lista para recomendar ao usuário. Essa abordagem proporciona sugestões personalizadas, levando em consideração as características semelhantes entre os alimentos no mesmo cluster, oferecendo uma experiência de recomendação diversificada e adaptada às preferências do usuário. O resultado é impresso, apresentando o alimento recomendado com base nesse processo de agrupamento, que como vemos na Figura 19 o prato recomendado é “Tapioca”.

Figura 19: Escolhendo um prato.



```
cluster_recomendado = df['Cluster'].sample(1).values[0]
alimentos_no_cluster = df[df['Cluster'] == cluster_recomendado]['chosendish'].tolist()
alimento_recomendado = random.choice(alimentos_no_cluster)
print(f"O Prato recomendado entre os mais avaliados é: {alimento_recomendado}")
```

```
O Prato recomendado entre os mais avaliados é: Tapioca
```

Fonte: Autoria própria

5 CONCLUSÃO

Na conclusão deste trabalho, é possível destacar os avanços significativos alcançados, o projeto teve como principal objetivo criar um sistema que oferecesse sugestões de pratos de forma personalizada e eficaz aos usuários, considerando as avaliações dos consumidores e a colaboração entre eles no ambiente da base de dados utilizada do iFood.

Durante o desenvolvimento, observou-se que a abordagem adotada, utilizando o K-Means, proporcionou resultados promissores na tarefa de recomendação. A análise detalhada das avaliações dos consumidores, a escolha criteriosa das características relevantes dos pratos e a implementação efetiva do algoritmo contribuíram para a eficácia do sistema..

Em termos práticos, o sistema desenvolvido oferece uma solução viável para o desafio de recomendação de pratos no contexto da base de dados do iFood. A capacidade de personalização e a eficácia do sistema na identificação de padrões de preferências dos consumidores indicam um potencial significativo para melhorar a experiência do usuário na plataforma. No entanto, é crucial reconhecer que a busca por aprimoramentos contínuos, tanto na qualidade dos dados quanto nas métricas utilizadas para avaliação, é um aspecto essencial para o desenvolvimento futuro do sistema de recomendação e até mesmo para novas funcionalidades como uma personalização na recomendação por usuários, entre outras. Este trabalho serve não apenas como um ponto culminante deste projeto, mas também como um ponto de partida para pesquisas futuras e refinamentos na área de sistemas de recomendação alimentar.

REFERÊNCIAS

D.A. Adeniyi, Z. Wei, Y. Yongquan **“Automated web usage data mining and recommendation system using K-Nearest Neighbor (KNN) classification method”**; 2017. Disponível em :
< https://www.sciencedirect.com/science/article/pii/S221083271400026X?via%3Dihub> Acessado em 29 nov. 2023.

H.R. Koosha; Z. Ghorbani; R. Nikfetrat **“A Clustering-Classification Recommender System based on Firefly Algorithm”**; 2021. Disponível em :
< https://jad.shahroodut.ac.ir/?_action=article&au=24486&_au=H.R.++Koosha> Acessado em 29 nov. 2023.

TACHINARDI, Ricardo **“iFood Restaurants Data”**; 2021. Disponível em :
< https://www.kaggle.com/datasets/ricardotachinardi/ifood-restaurants-data> Acessado em 29 nov. 2022.

SOUZA, Igor **“Descubra como o algoritmo KNN pode classificar seus dados”**; 2023. Disponível em :
< https://medium.com/@igor1245souza/descubra-como-o-algoritmo-knn-pode-classificar-seus-dados-886d04042576> Acessado em 29 nov. 2023.

RICCI, et al., Recommender Systems Handbook, edição 2011, NewYork: Springer Science+Business Media, 2010.

OLIVEIRA. “**Clusterização ou Agrupamento de Dados. IME: IME Unicamp”**, 2012. 17 slides. Disponível em: https://www.ime.unicamp.br/~wanderson/Aulas/Aula6/MT803-Aula06- Clusterizacao.pdf , Acessado em 29 nov. 2023

FONTANA, Éliton. “**Introdução aos Algoritmos de Aprendizagem Supervisionada”**, 2020. Disponível em: https://fontana.paginas.ufsc.br/files/2018/03/apostila_ML_pt2.pdf, Acessado em 29 nov. 2023

IBM. “**O que é aprendizado supervisionado”**, 2020. Disponível em: https://www.ibm.com/br-pt/topics/supervised-learning, Acessado em 29 nov. 2023