

Universidade Federal do Paraná
Setor de Ciências Exatas
Departamento de Estatística
Programa de Especialização em *Data Science* e *Big Data*

Diógenes Antonio Falcade Pereira

Sistema de Recomendação de Restaurantes por Filtragem Baseada em Conteúdo com Análise de Sentimentos

**Curitiba
2025**

Diógenes Antonio Falcade Pereira

Sistema de Recomendação de Restaurantes por Filtragem Baseada em Conteúdo com Análise de Sentimentos

Monografia apresentada ao Programa de Especialização em *Data Science* e *Big Data* da Universidade Federal do Paraná como requisito parcial para a obtenção do grau de especialista.

Orientador: Prof. Marcos A. Zanata Alves

Curitiba
2025

Sistema de Recomendação de Restaurantes por Filtragem Baseada em Conteúdo com Análise de Sentimentos

Restaurant Recommendation System Based on Content-Based Filtering and Sentiment Analysis

Diógenes Antonio Falcade Pereira¹, Marco Antonio Zanata Alves²

¹Aluno do programa de Especialização em Data Science & Big Data, diogenesfalcade@gmail.com

²Professor Adjunto e Pesquisador Departamento de Informática - DINF/UFPR, mazalves@inf.ufpr.br

Este trabalho apresenta o desenvolvimento de um sistema de recomendação de restaurantes por meio da técnica de Filtragem Baseada em Conteúdo, com aprimoramento por análise de sentimentos aplicada a avaliações textuais. A solução utiliza dados reais de estabelecimentos gastronômicos obtidos por meio das APIs públicas do Google Maps e TripAdvisor, contendo atributos estruturados e comentários de usuários. Esses comentários foram classificados automaticamente quanto ao sentimento predominante utilizando um modelo de linguagem leve (LLM), o *Ollama 3.2 1B*, executado localmente. As recomendações são geradas por similaridade entre restaurantes, calculada com a distância do cosseno sobre componentes principais extraídos via *PCA*, ponderadas por sentimento agregado e nota média. Para otimização do desempenho em ambiente web, todas as recomendações são processadas previamente e armazenadas em um único arquivo, permitindo resposta instantânea na plataforma *Streamlit Community Cloud*. A aplicação busca combinar relevância e diversidade nas sugestões, promovendo a descoberta de novos locais com base em reputação, sentimento e número de avaliações.

Palavras-chave: Sistemas de recomendação, filtragem baseada em conteúdo, análise de sentimentos, LLM, aprendizado não supervisionado, restaurantes, Streamlit

This paper presents the development of a restaurant recommendation system based on Content-Based Filtering, enhanced by sentiment analysis of user reviews. The solution uses real-world data collected from the public APIs of Google Maps and TripAdvisor, including both structured attributes and user comments. These comments were automatically classified according to their predominant sentiment using a lightweight language model (LLM), *Ollama 3.2 1B*, executed locally. Recommendations are generated through cosine similarity computed over principal components extracted via *PCA*, and weighted by aggregated sentiment and average ratings. To optimize performance in a cloud environment, all recommendations are preprocessed offline and stored in a single file, enabling instant delivery through the *Streamlit Community Cloud* platform. The system aims to balance relevance and diversity by suggesting high-quality and lesser-known restaurants, based on sentiment, reputation, and number of reviews.

Keywords: Recommendation systems, content-based filtering, sentiment analysis, LLM, unsupervised learning, restaurants, Streamlit

1. Introdução

Com o crescimento exponencial de dados disponíveis online, os sistemas de recomendação tornaram-se ferramentas essenciais para auxiliar usuários na tomada de decisões, especialmente em contextos com muitas opções, como a escolha de restaurantes [1]. Esses sistemas utilizam diferentes abordagens para sugerir itens relevantes, destacando-se a filtragem baseada em conteúdo, a filtragem colaborativa e os métodos híbridos [2].

No contexto de recomendação de restaurantes, as avaliações e comentários públicos tornaram-se fontes ricas de informação sobre preferências dos usuários, qualidade do serviço e aspectos específicos da experiência gastronômica [2]. Enquanto abordagens tradicionais se baseiam apenas em notas ou atributos fixos, métodos mais recentes utilizam análise de sentimentos para extrair preferências implícitas dos usuários a partir de suas opiniões textuais [1, 3].

A análise de sentimentos é especialmente promissora nesse domínio, pois permite detectar nuances nas

avaliações, diferenciando aspectos como qualidade da comida, atendimento, ambiente e preço [4]. Apesar disso, muitos sistemas ainda ignoram o potencial da análise textual em profundidade, limitando-se a dados estruturados ou modelos genéricos. Trabalhos anteriores demonstram que o uso combinado de técnicas de processamento de linguagem natural (NLP) e aprendizado de máquina pode melhorar significativamente a precisão das recomendações [1, 3, 5].

Este trabalho propõe um sistema de recomendação de restaurantes baseado em similaridade de perfil entre estabelecimentos, enriquecido com a análise de sentimentos das avaliações de usuários. A coleta dos dados foi realizada por meio de chamadas às APIs públicas do Google Maps e TripAdvisor, seguidas de um processo de engenharia de atributos e armazenamento em banco de dados PostgreSQL local. Para a classificação dos sentimentos, foi utilizada uma LLM leve (Ollama 3.2 1B), com desempenho satisfatório na tarefa de rotulação de sentimentos.

O presente artigo está organizado da seguinte forma: a Seção 2 discute os conceitos e trabalhos relacionados; a Seção 3 descreve os materiais utilizados e os métodos empregados no desenvolvimento do sistema; a Seção 4 apresenta o sistema proposto; a Seção 5 aponta a validação do sistema; e a Seção 6 expõe as conclusões e possíveis direções futuras.

2. Discussão

Sistemas de recomendação têm sido amplamente utilizados como ferramentas para auxiliar na tomada de decisão em contextos com elevada oferta de opções. Essas ferramentas são especialmente relevantes no setor de serviços, como o gastronômico, onde a escolha de um restaurante envolve múltiplos critérios subjetivos e contextuais. Tradicionalmente, os sistemas de recomendação podem ser classificados em três categorias principais: colaborativos, baseados em conteúdo e híbridos [6].

Os sistemas colaborativos baseiam-se no comportamento e nas preferências de múltiplos usuários, recomendando itens que usuários semelhantes avaliaram positivamente. Já os sistemas baseados em conteúdo analisam as características dos itens e as preferências explícitas ou implícitas do próprio usuário, sugerindo itens similares aos que ele já gostou. Por fim, os sistemas híbridos combinam ambas as abordagens para superar limitações individuais, como o problema do usuário novo nos sistemas colaborativos

ou a falta de diversidade nas recomendações baseadas em conteúdo [6].

Este trabalho adota uma abordagem de filtragem baseada em conteúdo, na qual a recomendação é feita a partir da comparação entre os atributos dos próprios itens, que neste caso, são os restaurantes. Esse tipo de abordagem é particularmente útil quando não se dispõe de um histórico de interações por parte dos usuários, como avaliações explícitas ou comportamento de navegação. Ao invés disso, características dos estabelecimentos, como categorias culinárias, faixa de preço, horário de funcionamento e estrutura física, são utilizadas para estimar a similaridade entre eles.

A métrica de similaridade adotada neste projeto é a distância do cosseno, amplamente utilizada em aplicações de recomendação por seu desempenho na comparação de vetores de alta dimensionalidade [7]. Para tornar essa comparação mais eficiente e reduzir o ruído em atributos pouco relevantes, aplicou-se a técnica de Análise de Componentes Principais (*Principal Component Analysis* — PCA), que transforma o espaço de atributos originais em um subconjunto de componentes ortogonais que retêm a maior parte da variância dos dados [8].

Como aprimoramento do modelo, foi incorporada a análise de sentimentos das avaliações textuais fornecidas por usuários. A análise de sentimentos tem sido uma ferramenta valiosa em sistemas de recomendação, pois permite extrair informação subjetiva que complementa os dados estruturados dos itens [9]. Para este fim, utilizou-se um modelo de linguagem leve (LLM), o *Ollama 3.2 1B*, capaz de classificar comentários como positivos, neutros ou negativos com boa acurácia e baixo custo computacional, sendo executado localmente. O resultado da análise de sentimentos foi integrado ao modelo de recomendação como um fator adicional de ponderação, juntamente com a nota média dos restaurantes.

Além da relevância, a diversidade das recomendações também foi considerada, visando evitar sugestões redundantes e promover o descobrimento de novos estabelecimentos. Técnicas como penalização por número reduzido de avaliações e inclusão de sugestões aleatórias entre os estabelecimentos com melhor reputação e baixa visibilidade foram empregadas com esse objetivo.

A discussão dos conceitos aqui apresentados embasa as escolhas metodológicas detalhadas na seção seguinte e justifica a adoção de uma arquitetura baseada em pré-processamento offline, como estratégia

para garantir a leveza e a escalabilidade do sistema em ambiente de nuvem gratuita.

3. Materiais e métodos

Esta seção apresenta os dados utilizados, as ferramentas empregadas e o fluxo metodológico adotado para a construção do sistema de recomendação.

3.1. Coleta e Armazenamento dos Dados

A base de dados foi construída a partir da integração com as APIs públicas do *Google Maps Places* [11] e do *TripAdvisor* [12]. As duas APIs foram utilizadas para a identificação geográfica de restaurantes ativos na cidade de Curitiba, enquanto apenas a API do *TripAdvisor* forneceu informações detalhadas sobre os estabelecimentos, como:

- Categorias culinárias;
- Faixa de preço;
- Notas médias (geral, comida, serviço, valor);
- Número de avaliações;
- Horários de funcionamento;
- Comentários textuais de usuários (até 25 por restaurante).

O processo de coleta foi realizado entre janeiro e fevereiro de 2025, respeitando os limites de requisições das APIs gratuitas. Os dados foram armazenados localmente em um banco de dados PostgreSQL estruturado, permitindo acesso via SQL e integração com Python. Na Tabela 1 são exibidas as estatísticas da base de dados coletada.

Tabela 1: Estatísticas descritivas da base de dados

Métrica	Valor
Total de restaurantes	2.181
Total de avaliações	24.176
Avaliações com sentimento positivo	78,1%
Avaliações com sentimento negativo	19,4%
Avaliações com sentimento neutro	2,5%
Média total das notas das avaliações	3.05

3.2. Pré-processamento e Engenharia de Atributos

Após a coleta, os dados passaram por um processo de enriquecimento e transformação. A etapa de engenharia de atributos ou *feature engineering* incluiu:

- Codificação binária de categorias culinárias e características de serviço;

- Conversão de faixas de horário de funcionamento em variáveis temporais binárias;
- Conversão da faixa de preço para valores numéricos;
- Remoção de atributos com poucas ocorrências;

A base utilizada possuía em torno de 154 variáveis distintas advindas tanto da codificação binária quanto da conversão de faixa de horários. A classificação de cada restaurante depende das categorias que são adicionadas, o que eventualmente acarreta em culinárias extremamente específicas e que aparecem uma vez, ou até que têm nomes diferentes mas representam o mesmo aspecto. A Tabela 1 exemplifica como as categorias que foram binarizadas ficaram na base de dados.

Tabela 2: Variáveis binárias por restaurante

Restaurante	Italiana	Churrascaria	Manhã
Restaurante A	1	0	0
Restaurante B	1	0	1
Restaurante C	0	1	1

3.3. Redução de Dimensionalidade

Devido à alta dimensionalidade gerada pelo processo de *feature engineering*, foi aplicada a técnica de Análise de Componentes Principais (PCA) conforme proposto em [8], com preservação de 95% da variância explicada. A transformação visou:

- Reduzir redundâncias;
- Facilitar o cálculo de similaridade;
- Tornar o sistema mais eficiente computacionalmente.

Essa etapa foi realizada com o uso da biblioteca *scikit-learn*. O conjunto reduzido de componentes serviu de base para a matriz de similaridade por cosseno, detalhada posteriormente.

3.4. Análise de Sentimentos

As avaliações textuais foram classificadas com base em seu sentimento predominante, utilizando o modelo leve de linguagem *Ollama 3.2 1B*, executado localmente. O modelo foi instruído a não retornar explicações ou texto adicional, apenas uma saída em formato JSON do tipo "sentimento". O retorno deve ser um rótulo (positivo, negativo ou neutro) para cada comentário, que era armazenado no banco de dados.

Após a classificação individual, as avaliações foram agregadas por restaurante, gerando o score de sentimento:

$$\text{score} = P_{\text{positivo}} - P_{\text{negativo}}$$

(1)

onde P representa a proporção de comentários de cada classe. Essa variável agregada foi posteriormente utilizada no cálculo do *score* de recomendação, como será detalhado na próxima seção.

Tabela 3: Classificações reais de sentimento via LLM	
Comentário	Sentimento
“Melhor pizza de pepperoni da vida! Preço justíssimo, ótimo atendimento, ótimo cheddar e ótimos sabores. Super recomendo!”	Positivo
“Não indico esse restaurante. Demora para trazer a comida e as bebidas... péssimo atendimento... porções muitoooo pequenas e que não condizem com a foto anunciada. Preço alto por pouco! Não vale a pena!!.”	Negativo
“O lugar é ok. Nada de espetacular. Eu esperava mais. O atendimento era ok. A pizza era boa. Os preços estavam ok. Mas não sei se voltaria.”	Neutro

Inicialmente, também foi testado um classificador baseado em BERT utilizando embeddings gerados pelo modelo pré-treinado *bert-base-portuguese-cased*. No entanto, os resultados mostraram-se inconsistentes para as avaliações, com maior tendência a enviesar classificações. Apesar de fornecer um *score* numérico que facilitaria a classificação mais subjetiva, no fim, isso foi insuficiente. Além disso, o custo computacional e o tempo de inferência foram significativamente mais altos, o que inviabilizou seu uso no ambiente local para classificar 24 mil comentários. Por esse motivo, optou-se pelo modelo leve Ollama 3.2 1B, que apresentou desempenho mais estável e compatível com os requisitos do projeto, levando em média 10 horas para classificar todos os comentários.

3.5. Ferramentas Utilizadas

- **Linguagem:** Python 3.11
- **Bibliotecas:** Pandas, scikit-learn, SQLAlchemy, Streamlit

- **Banco de dados:** PostgreSQL
- **APIs utilizadas:** Google Maps Places, TripAdvisor
- **Modelo de LLM:** Ollama 3.2 1B (executado localmente)

4. Sistema Proposto

Esta seção descreve a lógica de geração das recomendações, incluindo as etapas de cálculo de similaridade, composição do *score* final, inclusão de recomendações diversificadas e otimização para entrega em ambiente web.

4.1. Resumo do Fluxo de Recomendação

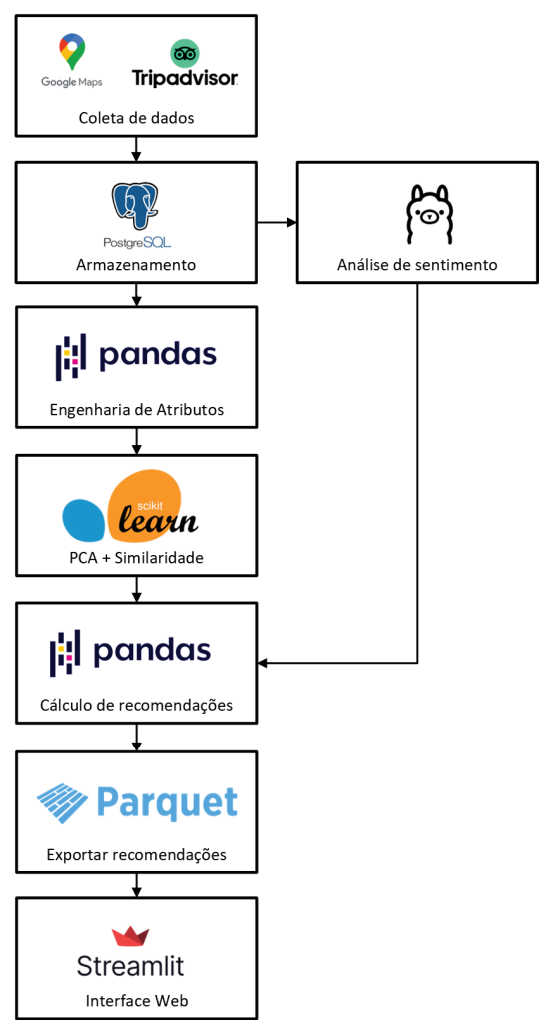


Figura 1: Fluxo resumido do sistema de recomendação

4.2. Cálculo de Similaridade entre Restaurantes

Após a aplicação do *PCA*, os restaurantes foram representados como vetores no novo espaço de com-

ponentes. A similaridade entre eles foi calculada utilizando a métrica da distância do cosseno:

$$\text{sim}(A, B) = \frac{A \cdot B}{|A| \cdot |B|} \quad (2)$$

O resultado é uma matriz densa de similaridade, na qual cada elemento indica o grau de proximidade entre dois restaurantes. Para cada restaurante da base, foram selecionados os 10 mais similares como candidatos à recomendação.

4.3. Cálculo do *Score* Ponderado de Recomendação

Cada um dos 10 candidatos foi reavaliado com base em um *score* composto, ponderando três critérios:

- Similaridade vetorial entre os restaurantes (70%);
- Sentimento médio agregado das avaliações (10%);
- Nota média geral (20%).

$$\text{score} = 0.7 \times \text{similaridade} + 0.1 \times \text{sentimento} + 0.2 \times \text{nota} \quad (3)$$

As notas e os sentimentos foram previamente normalizados (mínimo-máximo) para permitir comparação adequada. Apenas os restaurantes com pelo menos 5 avaliações foram elegíveis para compor o top 5 final.

4.4. Diversificação da Recomendação

Além dos cinco restaurantes com melhor *score*, foram adicionadas até três sugestões complementares com o objetivo de promover a descoberta de novos locais. As regras para diversidade foram:

- Dois restaurantes escolhidos aleatoriamente entre os 100 melhores no ranking geral, excluindo os já recomendados.
- Um restaurante com menos de 100 avaliações (considerado novidade), mas com *score* de sentimento positivo

Essas recomendações visam apresentar alternativas menos conhecidas, mas potencialmente interessantes, equilibrando popularidade e novidade.

4.5. Extração e Armazenamento

Para viabilizar o uso da aplicação em ambiente de nuvem com recursos limitados, como o *Streamlit Community Cloud*, o sistema foi projetado para funcionar em duas etapas:

1. Processamento offline: todas as recomendações foram geradas localmente, uma vez, para cada restaurante.
2. Armazenamento estruturado: os resultados finais foram armazenados em um único arquivo no formato `.parquet`, incluindo:

- ID, nome, nota, número de avaliações, faixa de preço e link do site do restaurante;
- As cinco recomendações principais;
- As recomendações diversas;
- *Score* de sentimento.

Esse arquivo consolidado é carregado instantaneamente no servidor, eliminando a necessidade de processamento em tempo real.

4.6. Interface e Otimização

A camada de interface foi construída com a biblioteca *Streamlit*, que permite a criação de dashboards interativos com poucos comandos em Python.

O usuário interage com a aplicação selecionando um restaurante que conhece e recebe como retorno:

1. Um conjunto de cinco recomendações principais, formatadas como cartões com link, nota, número de avaliações e preço;
2. Três sugestões adicionais, com foco em descoberta e diversidade;
3. Interface responsiva e carregamento imediato, mesmo em ambiente de nuvem gratuito.

Para ilustrar, a Figura 2 apresenta um exemplo da interface final em funcionamento. As recomendações são divididas em suas seções específicas para visualizar a separação entre similaridade e diversidade.

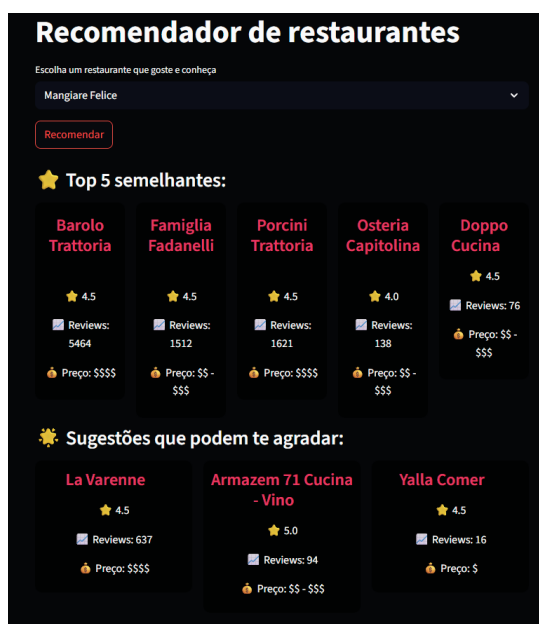


Figura 2: Interface web de recomendação construída com Streamlit

O carregamento do arquivo *parquet* é feito com cache nativo do Streamlit, o que evita reprocessamento entre interações do usuário. Novamente, como as recomendações já estão prontas, isso dispensa cálculos intensivos em tempo de execução. Essa escolha resultou em tempo de resposta inferior a 1 segundo por recomendação, mesmo na instância gratuita de nuvem do *Streamlit Community Cloud*.

5. Validação e Avaliação do Sistema

A fim de avaliar a qualidade e coerência das recomendações geradas, foi conduzida uma validação combinando métodos qualitativos e empíricos com base em dados históricos reais de usuários.

5.1. Validação Qualitativa

Nesta etapa, foi analisada a relevância semântica das recomendações geradas para restaurantes populares. O objetivo foi verificar se os estabelecimentos recomendados apresentavam similaridade plausível com o restaurante de origem, considerando atributos como tipo de culinária, faixa de preço, *score* de sentimento e reputação geral.

A Tabela 4 apresenta um exemplo desse tipo de validação para três restaurantes da base:

Tabela 4: Exemplo qualitativo de recomendações geradas

Restaurante Base	Recomendações
Mercearia Bresser	Pizzaria Dom Carmino, Piola, Baggio Pizzeria & Focacceria, Pizzaria Mercearia Bresser, Pizza Bis
Bar do Alemão	Cantinho do Eisbein, Batel Grill Churrascaria, Le Chef Confeitaria, Brioche Padaria, Coco Bambu
Lucca Cafés Especiais	Café do Mercado, Arte & Letra, Empório Kaveh Kanes, Café e Livraria Solar do Rosário, Padaria Família Farinha

Observa-se que, de modo geral, os restaurantes sugeridos compartilham características compatíveis com o restaurante de entrada, evidenciando a eficácia do cálculo de similaridade e do *score* composto. Em alguns casos, o sistema chegou, inclusive, a recomendar outras unidades da mesma rede do restaurante, demonstrando o alinhamento entre conteúdo e recomendação.

5.2. Validação com Histórico de Usuários

Para avaliar empiricamente a eficácia do sistema, foi utilizado um subconjunto de 24 usuários reais com histórico de avaliações em Curitiba, totalizando 174 comentários. Após filtragem, 28 comentários foram descartados por se referirem a restaurantes ausentes na base de dados, resultando em 146 avaliações para validação.

A estratégia adotada foi inspirada no método *Leave-One-Out Cross Validation*, adaptada ao contexto de sistemas de recomendação baseados em conteúdo. Para cada instância:

1. Selecionou-se um restaurante previamente avaliado pelo usuário.
2. Geraram-se as cinco principais recomendações com base nesse restaurante.
3. Verificou-se se algum dos restaurantes sugeridos também havia sido avaliado pelo mesmo usuário.

A Tabela 5 apresenta a distribuição da quantidade de recomendações coincidentes com o histórico dos usuários:

Tabela 5: Coincidência entre recomendações e locais já avaliados pelo usuário

Coincidência	Casos	Proporção (%)
3 recomendações	4	2,7%
2 recomendações	12	8,2%
1 recomendação	48	32,9%
Nenhuma	82	56,2%
Total	146	100%

Mesmo nos casos em que não houve sobreposição direta, foi possível observar que em 108 das 146 instâncias válidas (74%), ao menos uma das recomendações compartilhava categoria semelhante à de restaurantes previamente visitados pelo usuário. Isso reforça a capacidade do modelo em manter coerência semântica, mesmo sem perfil explícito do usuário.

Tabela 6: Cobertura semântica entre recomendações e histórico

Recomendações Similares	Quantidade
Uma ou mais	108
Nenhuma	38
Total	146

Apesar da ausência de perfis explícitos, a validação mostrou que o sistema é capaz de capturar preferências implícitas por meio da similaridade entre restaurantes, fornecendo recomendações relevantes, mesmo com dados limitados de usuário.

6. Conclusão

Este trabalho apresentou o desenvolvimento de um sistema de recomendação de restaurantes baseado em análise de similaridade de conteúdo e sentimentos extraídos de avaliações textuais. A solução foi construída a partir de dados reais coletados das APIs do Google Maps [11] e TripAdvisor [12], processados com técnicas de engenharia de atributos, redução de dimensionalidade com *PCA*, cálculo de similaridade por distância do cosseno e enriquecidos com análise de sentimentos utilizando um modelo de linguagem leve (*LLM*).

A principal contribuição do projeto está na integração eficiente de múltiplas fontes de dados, atributos estruturados, avaliações textuais e metainformações, para gerar recomendações relevantes e diversificadas. A arquitetura adotada, baseada em pré-processamento offline, demonstrou ser eficaz mesmo em ambientes

com recursos computacionais limitados, sem comprometer a experiência do usuário final.

O sistema conseguiu recomendar restaurantes similares com base em múltiplos critérios (nota, sentimento e perfil de serviço), mantendo coerência semântica nas sugestões, mesmo sem perfis explícitos dos usuários. A etapa de validação empírica, utilizando dados reais de usuários com múltiplas avaliações, reforçou a utilidade prática da abordagem e mostrou que há potencial para capturar preferências implícitas por meio da análise de conteúdo.

Como perspectivas futuras, destacam-se:

- A incorporação de elementos colaborativos, como feedback direto dos usuários e interações históricas;
- A expansão geográfica e temática do sistema para outras cidades ou setores (ex: hotéis, eventos, serviços);
- A expansão da base de comentários e perfis de usuários utilizando outros métodos, como web scrapping;
- A utilização de modelos de linguagem mais avançados, como *LLMs* otimizadas por *fine-tuning*, para refinar a análise semântica das avaliações;
- A automação completa do ciclo de coleta, atualização e entrega das recomendações, tornando o sistema autossustentável.

Os resultados obtidos demonstram que é possível combinar leveza computacional com profundidade analítica, resultando em um sistema funcional, escalável e aplicável a cenários reais de recomendação.

Referências

- [1] Asani, E., Vahdat-Nejad, H., & Sadri, J. (2021). Restaurant recommender system based on sentiment analysis. *Machine Learning with Applications*, 6, 100114.
- [2] Gao, Y., Yu, W., Chao, P., Zhang, R., Zhou, A., & Yang, X. (2015). A Restaurant Recommendation System by Analyzing Ratings and Aspects in Reviews. In *DASFAA 2015* (pp. 526–530). Springer.
- [3] Mahajan, K., et al. (2021). Restaurant Recommendation System Using Machine Learning. *International Journal of Advanced Trends in Computer Science and Engineering*, 10(3), 1671–1675.
- [4] Gomathi, R. M., Ajitha, P., Krishna, G. H. S., & Pranay, I. H. (2019). Restaurant Recommendation System for User Preference and Services Based on Rating and Amenities. *2nd ICCIDS*, IEEE.
- [5] Aye, Y. M., & Aung, S. S. (2019). Sentimental Analysis for Reviews of Restaurant in Myanmar Text. *2nd ICCIDS*, IEEE.

- [6] Ricci, F., Rokach, L., & Shapira, B. (2011). *Introduction to Recommender Systems*. Springer.
- [7] Huang, A. (2008). Measuring similarity between short texts. In *IEEE International Conference on Semantic Computing* (pp. 291–298).
- [8] Jolliffe, I. T. (2002). *Principal Component Analysis* (2nd ed.). Springer.
- [9] Feldman, R. (2013). Techniques and applications for sentiment analysis. *Communications of the ACM*, 56(4), 82–89.
- [10] Ollama (2024). *Ollama 3.2 1B: Local LLM for sentiment analysis*. Disponível em: <https://ollama.com>. Acesso em: 14/04/2025.
- [11] Google. (2025). *Google Maps Places API*. Disponível em: <https://developers.google.com/maps/documentation/places>. Acesso em: 22/12/2025.
- [12] TripAdvisor. (2025). *TripAdvisor API for Business Listings and Reviews*. Disponível em: <https://developer-tripadvisor.com>. Acesso em: 28/12/2024.
- [13] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- [14] PostgreSQL Global Development Group. (2025). *PostgreSQL Documentation*. Disponível em: <https://www.postgresql.org/docs/>. Acesso em: 13/01/2025.
- [15] Streamlit Inc. (2025). *Streamlit Documentation*. Disponível em: <https://docs.streamlit.io>. Acesso em: 02/06/2025.