Instituto Superior
de Engenharia
Politécnico de Coimbra

DEPARTAMENTO DE INFORMÁTICA E SISTEMAS

Exploração de Informação Textual na Recomendação de Pontos de Interesse

Trabalho de Projeto para a obtenção do grau de Mestre em Engenharia Informática

Especialização em Análise Inteligente de Dados

Autor

Karla Giovanna Jiménez Enríquez

Orientadora

Professora Ana Oliveira Alves

Professora do Departamento de Informática e Sistemas

Instituto Superior de Engenharia de Coimbra

INSTITUTO POLITÉCNICO
DE COIMBRA

INSTITUTO SUPERIOR
DE ENGENHARIA
DE COIMBRA

Coimbra, Fevereiro, 2025

"Challenge yourself, it's the only path which leads to growth."

Morgan Freeman

RESUMO

O aumento do volume de dados e o uso de serviços de localização tornaram os sistemas de recomendação de Pontos de Interesse (POIs) uma ferramenta essencial em áreas como turismo e navegação urbana. As principais funções dos Sistemas de Recomendação são a análise das diversas ações do utilizador do sistema. Com essa análise é possível extrair informações úteis para futuras predições, fornecendo recomendações de diferentes itens (por exemplo, sugestões de músicas, filmes, comércio eletrónico, etc). Existem diferentes variantes nos sistemas de recomendação, nomeadamente, sistemas de filtragem colaborativa de classificações (*ratings*), filtragem baseada em conteúdo dos itens (descrição, características, localização) ou de filtragem híbrida (que combinam as duas variantes anteriores), tendo todas por objetivo a seleção de conteúdos de interesse tendo em conta os padrões de consumo dos utilizadores.

Este estudo propõe um sistema de recomendação híbrido que integra filtragem colaborativa e filtragem por conteúdo, aproveitando informações textuais dos POIs, como descrições, nome e categorias, obtidas de uma rede social baseada em comunicação.

Com foco em Portugal como caso de estudo, foram desenvolvidos métodos para transformar e integrar variáveis textuais, utilizando técnicas de Processamento de Linguagem Natural (PLN) e aprendizado computacional. A implementação do sistema baseou-se na biblioteca LightFM, ampliando sua aplicação com métricas como precisão, *recall* e AUC para avaliar o desempenho.

A metodologia envolveu desde a recolha de dados, filtragem e transformação até a avaliação detalhada do modelo. O sistema demonstrou ser eficaz em fornecer recomendações mais personalizadas e relevantes, validando a importância de integrar dados textuais em sistemas de recomendação de Pontos de Interesse (POIs). As limitações, como a dependência de dados de uma rede social baseada em comunicação e o foco em Portugal, abrem caminho para estudos futuros que explorem sua aplicação em diferentes contextos geográficos e linguísticos.

Palavras-chave: Recomendação de Pontos de Interesse (POI); Filtragem Colaborativa; Filtragem por Conteúdo; Sistemas de Recomendação Híbridos; Processamento de Linguagem Natural (PLN); Aprendizagem computacional; Informação textual

ABSTRACT

The increase in data volume and the use of location-based services have made Point of Interest (POI) recommendation systems an essential tool in areas such as tourism and urban navigation. The primary functions of recommendation systems involve analyzing various user interactions within the system. Through this analysis, valuable insights can be extracted to support future predictions, providing recommendations for different items (e.g., music, movies, e-commerce suggestions, etc.). There are different variants of recommendation systems, including collaborative filtering based on user ratings, content-based filtering that considers item descriptions, features, and locations, and hybrid filtering, which combines both approaches. All these methods aim to deliver relevant content based on user consumption patterns.

This study proposes a hybrid recommendation system that integrates collaborative filtering and content-based filtering, leveraging textual information from POIs, such as descriptions, names, and categories, obtained from a location-based social network.

Focusing on Portugal as a case study, methods were developed to transform and integrate textual variables using Natural Language Processing (NLP) techniques and machine learning. The system implementation was based on the *LightFM* library, extending its application with evaluation metrics such as precision, recall, and AUC to assess its performance.

The methodology encompassed data collection, filtering, transformation, and an in-depth evaluation of the model. The system proved effective in providing more personalized and relevant recommendations, validating the importance of incorporating textual data into POI recommendation systems. Limitations, such as dependency on data from a location-based social network and the focus on Portugal, pave the way for future research exploring its application in different geographical and linguistic contexts.

Keywords: Recommendation of Points of Interest (POI); Collaborative Filtering; Content-Based Filtering; Hybrid Recommendation Systems; Natural Language Processing (NLP); Machine Learning; Textual Information

DEDICATÓRIA

A minha querida avó, Paulina. Querida abuelita, sei que me olhas de onde estás, e sei que um dia nos voltaremos a encontrar. Este projeto é para ti.

AGRADECIMENTOS

A conclusão desta tese marca o fim de um capítulo desafiante, mas gratificante, da minha vida académica e pessoal. Quero expressar a minha profunda gratidão às pessoas que, de diferentes formas, estiveram ao meu lado e me ajudaram a alcançar este objetivo.

Agradeço ao meu esposo, João, pelo seu apoio incondicional, por ser a minha força nos momentos de dúvida e por acreditar em mim própria quando eu vacilava. À minha mãe Rosa Elvia e às minhas irmãs, Nidia e Angie, pelo amor e incentivo constantes. À pequena Ema, que desde que nasceu tem visto a mamã com frequência em frente ao computador nas horas "vagas", obrigado por seres a minha alegria e inspiração diária.

Às minhas amigas mexicanas que encontrei em Portugal, obrigada pela amizade e por me fazerem sentir em casa mesmo longe da terra natal. À minha grande amiga Nanashi, és e serás sempre um pilar fundamental na minha vida.

Não poderia deixar de agradecer à minha orientadora, Ana Alves, pela sua orientação, confiança e motivação ao longo deste percurso. A sua orientação foi crucial para a concretização deste trabalho.

ÍNDICE

ÍNDICE DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS E ACRÓNIMOS

API	*Application Programming Interface*
AUC	*Area Under the Curve*
BPR	*Bayesian Personalized Ranking*
CBSN	*Communication-Based Social Network*
CSR matrix	*Compressed Sparse Row matrix*
DBSCAN	*Density-Based Spatial Clustering of Applications with Noise*
IA	Inteligência Artificial
IDF	*Inverse Document Frequency*
LBSN	*Location-Based Social Network*
MRR	*Mean Reciprocal Rank*
NDCG	*Normalized Discounted Cumulative Gain*
PLN	Processamento de Linguagem Natural
POI	*Points of Interest*
P-DBSCAN	*Popularity-based DBSCAN*
RMSE	*Root Mean Square Error*
SVM	*Support Vector Machine*
TF	*Term Frequency*
TF-IDF	*Term Frequency-Inverse Document Frequency*

1 INTRODUÇÃO

O aumento de dados e serviços de localização tornaram a recomendação de pontos de interesse (ou a sua denominação em inglês, *Points-Of-Interest* - POI) uma prática indispensável em várias áreas, desde o turismo até a navegação urbana e a descoberta de locais para atividades sociais. Os utilizadores confiam cada vez mais em sistemas de recomendação para orientá-los na escolha dos melhores POIs para suas necessidades e preferências.

A recomendação de pontos de interesse é uma das tarefas essenciais em redes sociais baseadas em localização (*Location-Based Social Networks - LBSNs*) para ajudar os utilizadores a descobrirem novos locais interessantes com base na aplicação generalizada de tecnologia de informação móvel e serviços de localização (Qiao et al., 2018).

Nas LBSNs, a recomendação de POIs é uma tarefa crucial, utilizando dados de *check-ins*[1], fotografias, geolocalização e outros atributos multimodais. Esses sistemas também consideram características dos utilizadores e atributos dos POIs, como sentimentos expressos em comentários e descrições textuais. No entanto, muitos sistemas tradicionais ainda se limitam a informações geográficas e históricas, subutilizando o contexto textual, que pode fornecer *insights*[2] valiosos sobre as características e atratividade dos locais.

Este projeto propõe uma abordagem que integra informações em formato texto em sistemas de recomendação de POIs, explorando descrições e categorias dos locais em Portugal como prova de conceito. Espera-se que esta integração melhore a precisão e relevância das recomendações, oferecendo sugestões mais alinhadas com as preferências dos utilizadores.

1.1 Definição de Objetivos

A problemática central deste estudo reside na subutilização de dados textuais nos sistemas de recomendação de POIs. Apesar de avanços nas técnicas de filtragem colaborativa e por conteúdo, muitos sistemas tradicionais se limitam a informações geográficas e históricos de *check-ins*, ignorando a informação textual, como descrições e categorias dos locais. Este estudo investiga a inclusão de dados textuais para melhorar a relevância das recomendações e personalizar melhor as sugestões para os utilizadores.

---

[1] O ato ou processo de comunicar que chegou a um hotel, aeroporto, etc. https://www.britannica.com/dictionary/check%E2%80%93in (último acesso janeiro 2025)

[2] Percepções, entendimentos ou descobertas valiosas. https://www.britannica.com/dictionary/insights (último acesso janeiro 2025).

O objetivo geral deste projeto é desenvolver um sistema de recomendação híbrido que integre a filtragem colaborativa e a filtragem por conteúdo, explorando informações textuais dos POIs. Para tal, o projeto será estruturado em diferentes etapas, incluindo a análise das abordagens existentes e a implementação do sistema. As metas específicas incluem:

- Estudar os sistemas de recomendação actuais de POIs, analisando as metodologias mais utilizadas, as suas vantagens e desafios, com base na revisão do estado da arte.
- Implementar um sistema de recomendação híbrido.
- Integrar variáveis textuais, como o nome, a categoria, a categoria raiz e a descrição dos POIs, extraídas da plataforma Foursquare.
- Avaliar a eficácia do sistema proposto, considerando métricas como precisão e relevância das recomendações em comparação inclusive com trabalhos do estado da arte.

1.2 Relevância do Estudo e Metodologia

- Este caso de estudo adota uma abordagem de pesquisa baseada em análise de dados e técnicas de aprendizagem computacional. A metodologia segue as seguintes etapas:
- Coleta de Dados**:** Utilização de rede social baseada em comunicação (LBSN) para obter dados e coletar informações dos POIs.
- Desenvolvimento do Sistema: A implementação do sistema de recomendação foi fortemente inspirada no tutorial "Recommendation System in Python". O código base foi adaptado para integrar informações textuais dos POIs, ajustando o formato e a estrutura dos dados.
- Avaliação e Métricas: A eficácia do sistema será medida usando métricas como precisão, *recall*, auc_score e MRR a serem explicadas no capítulo 2, para analisar o desempenho e a relevância das recomendações.

1.3 Estrutura do Relatório

Este relatório está organizado em sete capítulos, que estruturam a pesquisa, o desenvolvimento e a avaliação do sistema de recomendação proposto.

- Capítulo 2 – Conceitos Fundamentais: Apresenta os principais conceitos relacionados aos sistemas de recomendação, incluindo os métodos de filtragem baseada em conteúdo, filtragem colaborativa e abordagem híbrida. Além disso, são discutidos os fundamentos do processamento de linguagem

natural, medidas de avaliação e a aplicação do algoritmo de *clustering*[3] no contexto de POIs.

- Capítulo 3 – Estado da Arte: Fornece uma revisão da literatura sobre sistemas de recomendação de POIs, analisando estudos recentes e identificando as principais abordagens adotadas. São apresentados métodos utilizados, métricas de avaliação e comparações de desempenho entre diferentes estratégias.
- Capítulo 4 – Tecnologias e Ferramentas: Descreve as tecnologias, bibliotecas e ferramentas utilizadas no desenvolvimento do sistema, e outras *frameworks*[4] aplicadas no processamento de dados e na modelagem do sistema de recomendação.
- Capítulo 5 – Sistema de Recomendação: Apresenta a arquitetura geral do sistema proposto, detalhando o conjunto de dados utilizado, o processo de enriquecimento dos dados e as diferentes estratégias de recomendação adotadas, como a filtragem baseada em conteúdo e a filtragem colaborativa.
- Capítulo 6 – Resultados e Discussões: Explora a avaliação do desempenho do sistema de recomendação, analisando as métricas de avaliação aplicadas e discutindo os impactos da abordagem multilíngue e do enriquecimento dos dados na qualidade das recomendações.
- Capítulo 7 – Conclusão e Perspetivas Futuras: Fornece um resumo das principais contribuições do estudo, abordando as limitações encontradas e propondo direções para trabalhos futuros na área de recomendação de POIs.

[3] Algoritmo de aprendizado de máquina não supervisionado que organiza e classifica diferentes objetos. https://www.ibm.com/br-pt/think/topics/clustering (último acesso janeiro 2025)

[4] Ambiente de trabalho, para desenvolvimento de algum programa. https://ebaconline.com.br/blog/framework-seo (último acesso janeiro 2025).

2 CONCEITOS FUNDAMENTAIS

Este capítulo explora os sistemas de recomendação e a sua integração com o PLN, Inicialmente, são apresentados os tipos de sistemas de recomendação, como os baseados em conteúdo, a filtragem colaborativa e os sistemas híbridos, destacando as suas vantagens e desvantagens.

O capítulo também discute o papel da aprendizagem computacional na evolução dos sistemas de recomendação, tornando-os mais adaptáveis e precisos. Especificamente para recomendações de POI, é destacada a importância das informações de contexto e históricas. A secção final enfoca a informação textual, sublinhando a subutilização destes dados em recomendações e a relevância do PLN na análise e no treino de modelos de recomendação. Conclui-se com exemplos da aplicação de sistemas de recomendação em empresas como Amazon, Netflix e Spotify, ilustrando a sua importância prática e o impacto nas vendas e na satisfação do cliente.

2.1 Sistemas de Recomendação

O foco do desenvolvimento de um sistema de recomendação, normalmente utilizado por sites de comércio eletrónico, agências de viagens ou plataformas de música e filmes, consiste em fornecer sugestões de itens na mesma plataforma. Um dos casos mais elementares em sistemas de recomendação envolve algoritmos baseados em factorização, nos quais se consideram utilizadores e itens para criar recomendações com base nas avaliações (ou "*ratings*") dos utilizadores.

Os sistemas de recomendação são ferramentas que desempenham um papel fundamental em auxiliar e facilitar os utilizadores na pesquisa e descoberta de opções e elementos de interesse que aprimoram sua experiência. Globalmente, esses sistemas empregam critérios e dados dos utilizadores para realizar previsões e sugerir elementos que possam ser úteis. Eles coletam dados diretamente ou indiretamente relacionados com o histórico do utilizador e transformam essas informações em conhecimento ou recomendações.

A escolha das técnicas para o desenvolvimento de um sistema de recomendação está intimamente ligada ao tipo de informação que será utilizada. Em alguns casos, temos acesso apenas a um identificador para cada item, enquanto em outros, dispomos de informações mais detalhadas sobre os itens, incluindo descrições, classificações, avaliações, número de *check-ins* e outros atributos relevantes.

A identificação de dados representa uma fase essencial no processo de criação de um sistema de recomendação. Em geral, quanto mais sofisticada for a representação dos itens, mais eficiente será o processo de seleção para a sua recomendação.

Conforme detalhado (Aggarwal, 2016), os sistemas de recomendação podem ser formulados a partir de duas abordagens principais, que definem o problema em mãos:

- Versão de Previsão do Problema: Esta abordagem busca prever os valores de avaliação para combinações de utilizadores e itens, com base em dados de treino que representam preferências observadas. Este problema é conhecido como o problema de completação de matriz, uma vez que o modelo deve preencher valores ausentes em uma matriz de preferências parcialmente preenchida.
- Versão de Ranking do Problema: Focada em identificar os *top-k*[5] itens mais relevantes para um utilizador, esta abordagem não se preocupa em prever valores absolutos de avaliação, mas sim em ordenar as sugestões com base na relevância percebida. Essa formulação é amplamente utilizada em sistemas onde o objetivo é apresentar aos utilizadores as opções mais interessantes ou úteis, sem necessariamente associar valores numéricos às recomendações.

A escolha entre essas abordagens depende das metas do sistema e do tipo de dados disponíveis. Por exemplo, sistemas que visam otimizar a experiência do utilizador em termos de relevância e personalização podem priorizar a formulação de ranking, enquanto problemas mais específicos de previsão podem exigir a abordagem de completação de matriz. Essas formulações servem como base para as técnicas abordadas nas próximas seções deste capítulo.

Um mecanismo de recomendação pode agregar valor para as empresas e para utilizadores (Aggarwal, 2016), para alcançar o objetivo empresarial mais amplo de aumentar a receita, os objetivos operacionais e técnicos comuns dos sistemas de recomendação incluem:

1. **Relevância**: o objetivo mais evidente é recomendar itens relevantes para o utilizador. Os utilizadores têm maior probabilidade de consumir itens que considerem interessantes. No entanto, a relevância isolada não é suficiente, e por isso, objetivos secundários também são importantes.
2. **Novidade**: os sistemas de recomendação são úteis quando sugerem itens que o utilizador não viu antes. Por exemplo, filmes populares de um género preferido raramente serão novidade. Recomendações repetitivas podem reduzir a diversidade de vendas.
3. **Serendipidade**: este conceito refere-se à recomendação de itens inesperados, proporcionando uma descoberta agradável. Serendipidade difere da novidade porque as recomendações surpreendem o utilizador, revelando interesses latentes que ele mesmo desconhecia. Por exemplo, recomendar um novo restaurante indiano para um fã de comida indiana seria novidade, mas recomendar comida etíope poderia ser serendipidade. Embora essas recomendações possam às vezes ser irrelevantes, os benefícios estratégicos de longo prazo frequentemente superam as desvantagens de curto prazo.

[5] Refere-se aos k itens mais relevantes para um utilizador em um sistema de recomendação (Aggarwal, 2016).

4. **Aumentar a Diversidade nas Recomendações**: recomendação de listas mais diversificadas aumenta a probabilidade de um utilizador gostar de pelo menos um dos itens. Além disso, evita a monotonia de sugestões repetitivas.

É importante notar que a eficácia dos sistemas de recomendação depende da qualidade dos algoritmos subjacentes e da transparência na seleção de recomendações. Além disso, as questões de privacidade são cruciais, e a coleta de dados pessoais deve ser feita com o consentimento e em conformidade com regulamentações de proteção de dados.

Quanto à Figura 1, ela ilustra o processo de um sistema de recomendação, enfatizando a importância dos dados de entrada do utilizador na definição do modelo de recomendação. Isso destaca como a personalização baseada nas ações e preferências do utilizador é fundamental para o funcionamento desses sistemas.

Figura 1 - Esquema Sistema de recomendação

Os sistemas de recomendação podem ser classificados com base nas técnicas que utilizam para fazer as recomendações, dependendo dos dados disponíveis e da abordagem adotada.

No próximo subcapítulo, serão abordadas as técnicas mais comuns para o desenvolvimento de sistemas de recomendação, que podem ser divididas em três categorias: baseadas em conteúdos, filtragem colaborativa ou híbridas.

A compreensão dessas categorias é fundamental para a análise e seleção da técnica mais apropriada ao contexto de aplicação, uma vez que cada uma delas apresenta abordagens distintas para a geração de recomendações (Isinkaye et al., 2015).

2.2 Filtragem por Conteúdo

A filtragem por conteúdo é uma abordagem fundamental nos sistemas de recomendação, onde se utilizam os atributos descritivos dos itens para gerar recomendações personalizadas (Aggarwal, 2016). Esta é uma das abordagens mais populares em sistemas de recomendação, utilizando as características do conteúdo para sugerir itens semelhantes. Na filtragem por conteúdo, a recomendação de itens é feita com base na semelhança entre os itens a serem recomendados e aqueles que o utilizador já avaliou. Cada item é descrito por um conjunto de características que o definem, como palavras-chave ou categorias que descrevem categorias personalizadas (Aggarwal, 2016).

Segundo (Lops, P; de Gemmis, M; Semeraro, 2011) precisaram de técnicas adequadas para representar os itens e gerar o perfil do utilizador, além de estratégias para comparar o perfil do utilizador com a representação do item. A arquitetura de alto nível de um sistema de recomendação baseado em conteúdo é ilustrada na Figura 2. O processo de recomendação é realizado em três etapas, cada uma das quais é tratada por um componente separado:

- **Analisador de Conteúdo**: Este componente pré-processa informações não estruturadas, como texto, para extrair dados relevantes e representá-los de forma adequada. Ele transforma itens (ex.: documentos, páginas da web, descrições de produtos) em uma forma estruturada, como vetores de palavras-chave, que serão utilizados nas etapas seguintes.
- **Aprendiz do Perfil**: Aqui, os dados sobre as preferências do utilizador são coletados e generalizados para criar um perfil do utilizador. Técnicas de aprendizagem computacional são usadas para construir esse perfil com base nos itens que o utilizador gostou ou não no passado, criando um modelo que reflete seus interesses.
- **Componente de Filtragem**: Utilizando o perfil do utilizador, este módulo sugere itens relevantes, comparando o perfil com os itens disponíveis. A relevância dos itens é avaliada por métricas de similaridade, como a similaridade do cosseno, e gera uma lista classificada de itens que são potencialmente interessantes para o utilizador.

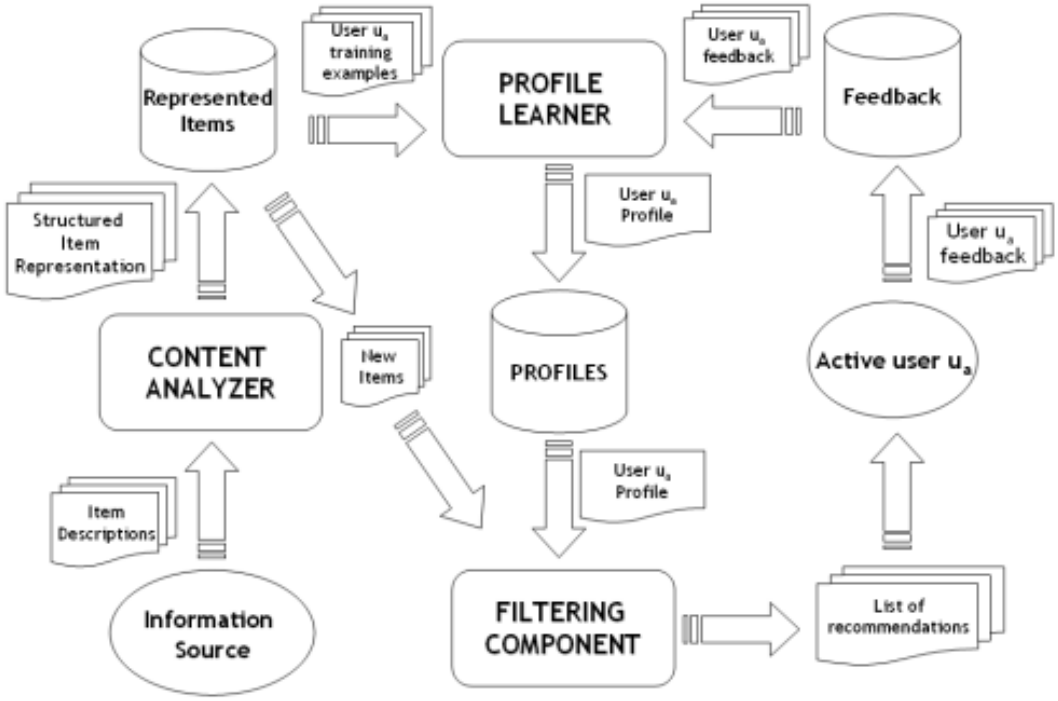


Figura 2 - Arquitetura de alto nível de um sistema de recomendação baseado em conteúdo (Lops, P; de Gemmis, M; Semeraro, 2011)

Num sistema baseado em conteúdo, o utilizador fornece, de forma implícita ou explícita, as suas preferências e restrições, e o sistema compara essas descrições com os itens contidos num catálogo de itens.

No caso da abordagem explícita, podemos referir-nos ao feedback que o utilizador fornece a um item, como uma avaliação de um restaurante. Esta abordagem facilita o processo de recomendação do sistema, embora seja importante notar que nem todos os utilizadores avaliam os itens, o que pode resultar em informações incompletas.

Já no caso implícito, é analisado o comportamento do utilizador. Alguns exemplos incluem o tempo que um utilizador passa num site, o histórico de compras, o histórico de navegação, entre outros. A Plataforma Last.FM[6] baseia-se na quantidade de vezes que um utilizador ouve uma canção para criar recomendações.

No entanto, a filtragem por conteúdo apresenta algumas desvantagens. Segundo (Aggarwal, 2016), uma delas é a tendência para fornecer recomendações óbvias, devido ao uso exclusivo de palavras-chave e descrições. Se um utilizador nunca consumiu um item com determinadas palavras-chave, esse item dificilmente será recomendado, o que pode reduzir a diversidade das recomendações.

[6] https://www.last.fm/ (último acesso em janeiro 2025).

2.3 Filtragem Colaborativa

A filtragem colaborativa é uma técnica de recomendação que utiliza informações das preferências e comportamentos de vários utilizadores para fazer previsões ou recomendações para um utilizador específico. Esta abordagem explora padrões de comportamento semelhantes entre os utilizadores para inferir as preferências individuais, permitindo que sistemas de recomendação sugiram itens ou conteúdos que são mais prováveis de interessar ao utilizador com base em comparações com outros utilizadores similares (Aggarwal, 2016).

A eficiência de um algoritmo colaborativo depende da precisão com que consegue encontrar a vizinhança do utilizador alvo. Tradicionalmente, os sistemas baseados em filtragem colaborativa enfrentam o problema do *cold-start* [7] (que descreve a dificuldade de fazer recomendações quando os utilizadores ou os itens são novos), e questões de privacidade, devido à necessidade de partilhar dados dos utilizadores (Roy & Dutta, 2022).

As abordagens colaborativas dividem-se em dois tipos principais:

1. Estratégias Baseadas na Memória - Recomendam novos itens considerando as preferências da vizinhança do utilizador e utilizam directamente a matriz de utilidade para previsões:

   - A construção do modelo baseia-se numa função da matriz de utilidade, onde o perfil do utilizador está contido nesta matriz. Segundo(Roy & Dutta, 2022), dada pela equação (2.1):

$$Modelo = f(Matriz\ de\ Utilidade) \qquad (2.1)$$

   - As recomendações são feitas com base no modelo gerado e no perfil do utilizador, onde este pertence à matriz de utilidade. Conforme (Roy & Dutta, 2022) é expressa pela equação (2.2) :

$$Recomendação = f(Modelo\ Definido, Perfil\ do\ Utilizador) \qquad (2.2)$$

Abordagens baseadas em memória dividem-se em:

- Filtragem Colaborativa Baseada em Utilizadores - A classificação de um novo item é calculada identificando outros utilizadores na vizinhança que já classificaram esse mesmo item. Se o item recebeu classificações positivas da vizinhança, será recomendado.

[7] Problema recorrente em Sistemas de Recomendação quando um novo item é adicionado ao sistema e não possui nenhuma avaliação prévia, (Mario Toledo, 2022).

- Filtragem Colaborativa Baseada em Itens - Cria uma vizinhança de itens semelhantes aos que o utilizador já classificou. A classificação de um novo item é prevista calculando a média ponderada das classificações dos itens semelhantes.

2. Estratégias Baseadas em Modelos - Estas abordagens utilizam algoritmos de mineração de dados e de aprendizagem computacional para criar um modelo que prevê as classificações de itens ainda não avaliados pelo utilizador. Extraem características do conjunto de dados para construir o modelo, sem depender da matriz de utilidade completa durante a recomendação.

   O processo também envolve dois passos:

   - Construir o modelo a partir dos dados disponíveis segundo (Roy & Dutta, 2022), é dada pela equação (2.3):

$$Modelo = f(Dados) \tag{2.3}$$

   - Fazer previsões com base no modelo e no perfil do utilizador segundo (Roy & Dutta, 2022), é dada pela equação (2.4):

$$Previsão = f(Modelo\ Definido, Perfil\ do\ Utilizador) \tag{2.4}$$

   Onde o perfil do utilizador não pertence à matriz de utilidade.

   As técnicas baseadas em modelos não requerem a adição do perfil de utilizador de um novo utilizador à matriz de utilidade antes de fazer previsões. Pode ser feito recomendações mesmo a utilizadores que não estão presentes no modelo. Os sistemas baseados em modelos são mais eficientes para recomendações em grupo. Eles podem recomendar rapidamente um grupo de itens utilizando o modelo pré-treinado. A precisão desta técnica depende em grande parte da eficiência do algoritmo de aprendizagem subjacente utilizado para criar o modelo (Roy & Dutta, 2022).

2.4 Factorização de Matrizes

Algoritmos baseados em factorização matricial são as implementações de maior sucesso dos modelos de factorização latente[8]. Em termos gerais, esta técnica é capaz de caracterizar tanto itens quanto utilizadores através de vetores de variáveis latentes inferidas de padrões de avaliação dos itens (Aggarwal, 2016).

[8] Dimensões escondidas que representam características dos utilizadores e dos itens, (Aggarwal, 2016).

No modelo básico de factorização de matriz (Aggarwal, 2016) a matriz R, que contém

as avaliações, é aproximadamente decomposta em duas matrizes: U, de dimensão $m \times k$, V, de dimensão $n \times k$, onde m representa o número de utilizadores e n o número de itens, assim como também k o número de fatores latentes. Essa relação pode ser expressa segundo (Aggarwal, 2016) e dada pela equação (2.5):

$$R \approx UV^t \tag{2.5}$$

Esta técnica permite representar de maneira compacta a relação entre utilizadores e itens, facilitando a geração de recomendações com base nos padrões de interação observados.

Cada coluna de U ou Vé um vetor latente e cada linha é um fator latente. A i-ésima linha u_i de U é chamada fator do utilizador e contém k entradas correspondentes a afinidade do utilizador i em relação aos k conceitos na matriz de notas. Segundo (Aggarwal, 2016), cada nota em r pode ser expressa como o produto escalar do i-ésimo fator do utilizador com o j-ésimo fator do item, tal como se apresenta na equação (2.6)

$$r_{ij} \approx \overline{u_i} \cdot \overline{v_j} \tag{2.6}$$

O modelo SVD (*Singular Value Decomposition* o seu acrónimo em inglês) representa tanto utilizadores quanto itens num espaço de fatores latentes de dimensão reduzida. A interação entre um utilizador u e um item i é modelada como o produto interno dos vetores latentes que os representam segundo (Aggarwal, 2016. Na equação (2.7) a previsão de uma avaliação $\hat{r}_{ui}$ é dada por:

$$\hat{r}_{u^i} = \mu + b_u + b_i + q_i^T p_u \tag{2.7}$$

Onde:

- u: média global das avaliações.
- $b_u \; e \; b_i$: enviesamentos associados ao utilizador e ao item.
- $q_i \; e \; p_u$: vetores latentes que capturam as características do item e as preferências do utilizador.

O modelo SVD++ expande o SVD básico ao incorporar informações implícitas, como itens que os utilizadores interagiram, mas não avaliaram explicitamente. Este modelo é particularmente eficaz para melhorar as previsões em cenários onde o

feedback implícito é mais prevalente. Segundo (Aggarwal, 2016), a previsão é feita tal como se mostra na equação (2.8):

$$\hat{r}_{ui} = u + b_u + b_i + q_i^T \left(p_u + \frac{1}{\sqrt{|\mathrm{Re}(u)|}} \right) \sum_{j \in R(u)} yj \tag{2.8}$$

Onde:

- $R(u)$: conjunto de itens com os quais o utilizador u interagiu.
- yj: vetores que capturam os fatores latentes associados aos itens no conjunto $R(u)$.

A integração do feedback implícito torna o SVD++ uma extensão robusta do SVD, capaz de lidar melhor com dados esparsos e melhorar a diversidade nas recomendações.

Os modelos SVD e SVD++ demonstram a capacidade da factorização de matriz em capturar padrões complexos nas interações entre utilizadores e itens. Enquanto o SVD fornece uma base sólida para a previsão de avaliações, o SVD++ amplia essa capacidade ao integrar feedback implícito, tornando-o uma escolha preferida em cenários com interações incompletas.

2.5 Filtragem Híbrida

A filtragem híbrida refere-se à combinação de múltiplas abordagens ou técnicas de recomendação em um único sistema. Essa combinação permite aproveitar as vantagens de diferentes métodos para melhorar a precisão e a qualidade das recomendações. Também é utilizada para superar as limitações individuais de cada técnica, proporcionando uma solução mais robusta e adaptável aos diferentes cenários e tipos de dados disponíveis em sistemas de recomendação (Aggarwal, 2016). Entre esses desafios estão o *cold-start* um problema comum tanto na filtragem colaborativa quanto nos sistemas baseados no conteúdo.

Tal como menciona (Aggarwal, 2016), existem três principais abordagens para criar sistemas de recomendação híbridos:

- **Design de Conjunto** - Neste modelo, os resultados de algoritmos já treinados são combinados para formar uma saída única e mais robusta. Por exemplo, combinar as avaliações de um sistema baseado em conteúdo com as de um sistema colaborativo resulta em uma recomendação conjunta mais eficaz.
- **Design Monolítico** - Aqui, um único algoritmo integrado de recomendação é desenvolvido, utilizando diferentes tipos de dados. Às vezes, as partes distintas do algoritmo (como conteúdo e colaboração) não

são claramente separadas, integrando-se mais profundamente as diversas fontes de dados.

- **Sistemas Mistos** - Similar aos designs de conjunto, esses sistemas utilizam múltiplos algoritmos de recomendação como caixas-pretas, porém apresentam os itens recomendados lado a lado, provenientes de diferentes sistemas.

O termo "sistema híbrido" é utilizado de forma mais abrangente do que "sistema de conjunto". Enquanto todos os sistemas de conjunto são, por definição, híbridos, o inverso nem sempre é verdadeiro.

Embora os sistemas de recomendação híbridos sejam frequentemente projetados para combinar diferentes abordagens, como filtragem baseada em conteúdo e filtragem colaborativa, também podem integrar múltiplos modelos dentro de uma mesma abordagem para otimizar o desempenho. No caso de sistemas baseados em conteúdo, que operam essencialmente como classificadores de texto, a aplicação de técnicas de aprendizagem automática em conjunto (*ensemble learning*) pode melhorar significativamente a precisão da classificação. Ao combinar os resultados de diferentes algoritmos, esta abordagem permite capturar com maior precisão as relações semânticas entre os conteúdos e as preferências dos utilizadores, resultando em recomendações mais relevantes e personalizadas.

A Figura 3 Ilustra uma categorização hierárquica desses diferentes tipos de sistemas:

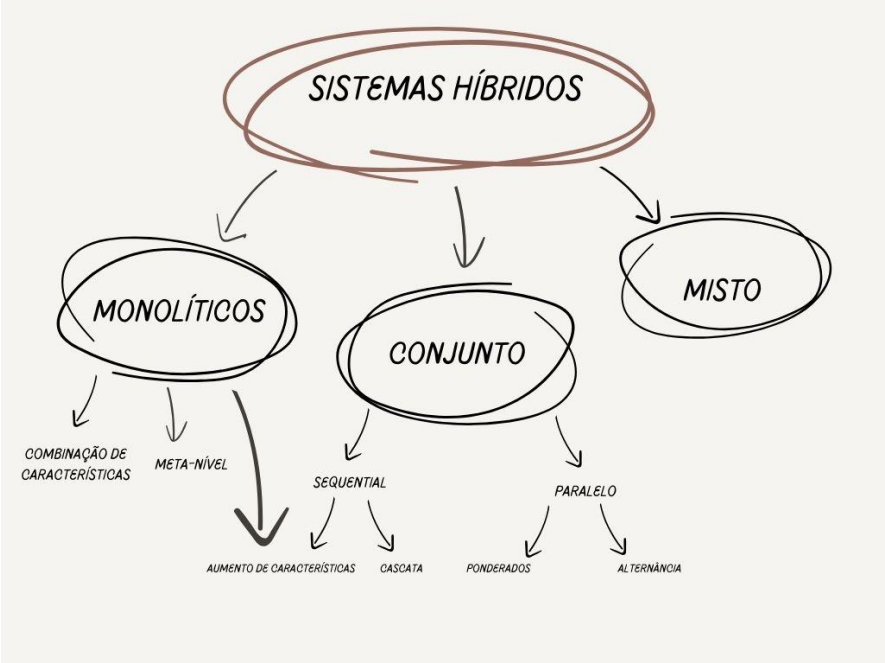


Figura 3 - Taxonomia dos sistemas híbridos

Neste contexto, (Aggarwal, 2016) descreve que os sistemas de recomendação híbridos podem ser classificados nos seguintes tipos:

- **Ponderados -** Neste caso, os *scores*[9] de vários sistemas de recomendação são combinados num único *score* unificado através do cálculo de médias ponderadas ou utilizando modelos estatísticos formais para determinar a contribuição de cada componente.
- **Alternância -** O sistema, alterna entre diferentes sistemas de recomendação dependendo das necessidades atuais. Por exemplo, pode-se começar com um sistema baseado em conhecimento (que recorrem a regras definidas, relações lógicas e informações contextuais para recomendar os itens mais adequados), para lidar com problemas de *cold start* e depois mudar para um sistema baseado em conteúdo ou filtragem colaborativa à medida que mais dados se tornam disponíveis.
- **Cascata -** Neste cenário, um sistema de recomendação refina as recomendações fornecidas por outro. Isso pode envolver o *boosting*[10], onde modelos subsequentes são treinados para corrigir ou melhorar as saídas dos modelos anteriores.
- **Aumento de Características** - Aqui, a saída de um sistema de recomendação é usada como características de entrada para o próximo sistema. Este método se assemelha ao *stacking*[11] na classificação, onde as previsões de um classificador são usadas como características para outro.
- **Combinação de Características** - Características de diferentes fontes de dados são combinadas dentro de um único sistema de recomendação. Esta abordagem é considerada monolítica pois não integra explicitamente saídas de múltiplos sistemas, mas sim consolida características de entrada.
- **Meta-nível** - O modelo usado por um sistema de recomendação informa o modelo de outro sistema. Por exemplo, um sistema de filtragem colaborativa pode incorporar características baseadas em conteúdo para refinar as definições de grupos de pares, melhorando a precisão das previsões.
- **Misto** - Recomendações de múltiplos motores são apresentadas simultaneamente aos utilizadores sem combinar explicitamente *scores*. Este método é distinto porque não agrega previsões, mas oferece várias recomendações, frequentemente adequadas para domínios de itens complexos ou em combinação com sistemas baseados em conhecimento.

---

[9] Resultado; pontuação, pontos. https://www.infopedia.pt/dicionarios/ingles-portugues/score (último acesso janiero 2025)

[10] Método usado em aprendizagem computacional para reduzir erros na análise preditiva de dados. https://aws.amazon.com/pt/what-is/boosting/ (último acesso janiero 2025)

[11] Técnica de sistemas de ensemble. https://machine-learning.martinsewell.com/ensembles/stacking/ (último acesso janiero 2025).

As primeiras quatro categorias mencionadas são classificadas como sistemas de *ensemble*[12], as duas seguintes como sistemas monolíticos, e a última como sistema misto. A categoria de sistemas mistos não pode ser claramente categorizada como monolítica ou de *ensemble* porque apresenta múltiplas recomendações como uma entidade composta.

Essas abordagens híbridas oferecem flexibilidade na geração de recomendações e podem ser adaptadas a diferentes cenários e necessidades. Elas são essenciais para superar os desafios encontrados em técnicas de recomendação individuais.

2.6 Funcionamento dos Sistemas de Recomendação

Para compreender a dinâmica dos sistemas de recomendação, é útil observar como os diferentes tipos de sistemas, modelos e métodos interagem para oferecer recomendações personalizadas. A **Erro! Fonte de referência não encontrada.** ilustra o fluxo geral de funcionamento desses sistemas, desde a recolha e processamento dos dados de utilizadores e itens, até à geração das recomendações finais.

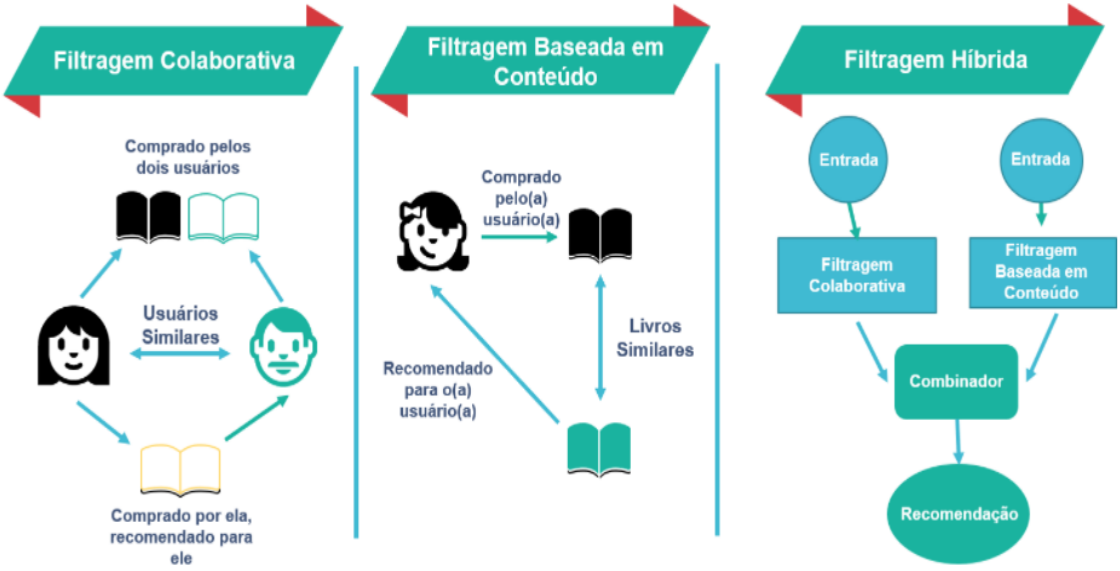


Figura 4 - Tipos de Sistemas de Recomendação (Viniski A., 2021)

Para complementar a explicação sobre os diferentes tipos de sistemas de recomendação, foi elaborada a Tabela 1, que apresenta uma comparação entre eles. Nesta tabela, destacam-se as características principais de cada abordagem, com ênfase no tipo de objetivo e entrada necessária.

[12] Métodos que combinam múltiplos modelos para melhorar previsões. https://didatica.tech/metodos-ensemble/ (último acesso janeiro 2025).

Ressalta-se que a construção da Tabela 1, é baseada na análise e compreensão do tema pelo autor, integrando informações teóricas e práticas discutidas ao longo do capítulo. O objetivo é proporcionar uma visão clara e objetiva das diferenças e semelhanças entre os sistemas, facilitando o entendimento do seu funcionamento e das suas aplicações.

Tabela 1 - Comparação de Tipos de Sistemas de Recomendação

Tipo	**Objetivo**	**Input**
Filtragem colaborativa	Oferece recomendações obtidas das iterações dos utilizadores.	Pontuação do utilizador
Baseada em conteúdo	Oferece recomendações baseadas nas descrições dos itens do sistema.	Pontuação do utilizador e atributos dos itens
Filtragem híbrida	Oferece recomendações combinando múltiplos métodos para melhorar a precisão.	Pontuação do utilizador, atributos dos itens e/ou interações de outros utilizadores

2.7 Aprendizagem Computacional

À medida que o tempo avança, os sistemas de recomendação evoluem, e grande parte dessa evolução é impulsionada pela aprendizagem computacional (ou *Machine Learning* em inglês) um ramo da inteligência artificial que foi definido por (Mitchell, 1997) como estudo de algoritmos de computador que permitem que programas de computador melhorem automaticamente através da experiência (Joshi, 2020). No fundo, de acordo com (Portugal et al., 2018), a aprendizagem computacional utiliza computadores para simular a aprendizagem humana e permite a esses computadores identificar e adquirir conhecimentos sobre o mundo real, e melhorar a performance de algumas tarefas baseadas nesse novo conhecimento. Os humanos aprendem naturalmente com a experiência por causa da sua capacidade de raciocinar. Em contraste, os computadores não aprendem raciocinando, mas sim com algoritmos (Portugal et al., 2018).

Com a aprendizagem computacional, os sistemas de recomendação podem considerar uma variedade de fatores, incluindo históricos de utilização, informações demográficas e feedback explícito dos utilizadores. Isso resulta em recomendações altamente personalizadas e relevantes, melhorando significativamente a experiência do utilizador.

A aprendizagem computacional desempenha um papel fundamental na eficácia e na adaptabilidade dos sistemas de recomendação, tornando-os ferramentas poderosas para plataformas online, como lojas virtuais, serviços de entretenimento e redes sociais.

Essa abordagem em constante evolução continua a moldar o campo dos sistemas de recomendação e desempenha um papel central na capacidade de fornecer recomendações de alta qualidade aos utilizadores.

2.8 Tipos de Aprendizagem Computacional

De acordo com o (Joshi, 2020) existem vários tipos de algoritmos de aprendizagem, no entanto os três mais relevantes são: *Supervised Learning* (aprendizagem supervisionada), *Unsupervised Learning* (aprendizagem não supervisionada) e o *Reinforcement Learning* (Aprendizagem por Reforço).

- **Aprendizagem supervisionada -** O modelo é treinado com dados históricos que contêm entradas e saídas conhecidas. Um exemplo clássico é a classificação, onde, com base em dados rotulados (como medições de flores), o modelo aprende a identificar ou classificar novos dados.
- **Aprendizagem não supervisionada -** Não há rótulos nos dados. Um exemplo é o agrupamento, onde o sistema organiza os dados em grupos ou padrões, como classificar flores em diferentes categorias, sem saber quais são os rótulos exatos. O objetivo é identificar estruturas subjacentes nos dados.
- **Aprendizagem por reforço -** O sistema interage com o ambiente, buscando aprender a tomar as melhores decisões com base em feedback contínuo, como recompensas ou punições. Não há dados rotulados, e o objetivo é maximizar a recompensa ao longo do tempo por meio de ações bem-sucedidas.

A aplicação de aprendizagem computacional em sistemas de recomendação oferece inúmeros benefícios que aprimoram a experiência do utilizador e a eficácia do sistema:

- **Melhoria da Precisão -** A aprendizagem computacional melhora a precisão das recomendações ao aprender continuamente com novos dados e ajustar os modelos de acordo.
- **Personalização Aprimorada -** Os modelos de aprendizagem computacional podem captar as preferências individuais dos utilizadores de forma mais detalhada, proporcionando uma experiência personalizada.
- **Capacidade de Previsão -** Algoritmos avançados de aprendizagem computacional conseguem prever tendências e comportamentos futuros dos

utilizadores, permitindo que o sistema de recomendação se antecipe às necessidades do utilizador.

- **Automatização -** O uso de aprendizagem computacional permite a automatização de processos complexos de análise de dados e geração de recomendações, reduzindo a necessidade de intervenção manual e melhorando a eficiência operacional.

2.9 Medidas de Avaliação em Sistemas de Recomendação

Os sistemas de recomendação empregam diversas métricas para avaliar o êxito das recomendações e algoritmos de previsão. Em pesquisas de filtragem colaborativa, a gestão de métricas desempenha um papel crucial na determinação precisa, rigorosa e exata da semelhança entre utilizadores e itens. Essas métricas foram projetadas para se adequar às características, particularidades e restrições específicas de cada sistema.

De acordo com o estudo conduzido por (Werneck et al., 2020) é importante destacar que, embora a precisão seja uma métrica amplamente priorizada, ela não abrange totalmente a complexidade dos sistemas de recomendação, especialmente no que diz respeito aos POIs. A precisão, que mede a exatidão das recomendações feitas, é uma métrica central, mas frequentemente desconsidera outras dimensões importantes da qualidade das recomendações, como a novidade e a diversidade[13]. Estas dimensões são cruciais para avaliar a eficácia de um sistema, pois não basta apenas recomendar itens que o utilizador já possa gostar com base no seu histórico. A verdadeira riqueza de um sistema de recomendação reside também na sua capacidade de expandir os interesses dos utilizadores, sugerindo novos e diversos POIs que eles possam gostar, mas que ainda não conhecem.

É relevante mencionar que a avaliação de sistemas de recomendação não se restringe apenas à precisão. Outras métricas, como a *Recall,* a *F1-Score* e a Acurácia, desempenham um papel importante na compreensão global do desempenho de um sistema de recomendação. Essas métricas permitem uma avaliação mais completa, considerando não apenas a precisão, mas também a abrangência e a capacidade de apresentar recomendações diversas e relevantes. A seguir, é detalhado as fórmulas das métricas e as variáveis envolvidas:

- Acurácia (*accuracy)*: mede a proporção de previsões corretas em relação ao total de previsões, tal como descreve (Aggarwal, 2016), dada pela equação (2.9).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.9}$$

[13] Explicado no subcapítulo 2.1

Onde:

TP (True Positive): POIs relevantes corretamente recomendados.

TN (True Negative): POIs irrelevantes corretamente não recomendados.

FP (False Positive): POIs irrelevantes que foram recomendados.

FN (False Negative): POIs relevantes que não foram recomendados.

- Precisão: mede a proporção de recomendações relevantes entre todas as recomendações feitas, (Aggarwal, 2016), dada pela equação (2.10).

$$Precision = \frac{TP}{TP + FP} \tag{2.10}$$

- Precisão@*k*: mede a proporção de itens relevantes entre as principais recomendações *k*. Segundo (Pullakandam Krishna, 2024), foca na qualidade das recomendações e é dada pela equação (2.11).

$$Precision@k = \frac{Relevantes@K}{K} \tag{2.11}$$

Onde: $Relevantes@k$ é o número de itens relevantes dentro do top-K recomendações e *k* é o número total de recomendações consideradas.

- *Recall*: mede a proporção de POIs relevantes corretamente recomendados entre todos os POIs relevantes, (Aggarwal, 2016), e é dada pela equação (2.12).

$$Recall = \frac{TP}{TP + FN} \tag{2.12}$$

- *Recall@K*: mede a proporção de itens relevantes que foram recuperados dentro dos *k* itens recomendados, em relação ao total de itens relevantes disponíveis, segundo (Pullakandam Krishna, 2024) e é dada pela equação (2.13).

$$Recall@k = \frac{Relevantes@K}{K} \tag{2.13}$$

- F1-Score: é a média harmônica entre a Precisão e o *Recall*, balanceando ambos os aspectos. É útil quando há necessidade de um equilíbrio entre as duas métricas (Aggarwal, 2016), e é dada pela equação (2.14).

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (2.14)$$

Além disso, é essencial considerar as métricas em relação às informações e ao objetivo específico do sistema de recomendação. Cada métrica pode destacar diferentes aspectos do desempenho, e a escolha da métrica adequada depende das metas e das necessidades do sistema em questão.

2.10 Informações de Tipo Textual

Geralmente, os sistemas de recomendação têm o objetivo de sugerir os itens mais importantes e relevantes para os utilizadores sem inicialmente considerar a informação contextual (Adomavicius & Tuzhilin, 2005).

O tipo de informação, pode assumir várias formas, incluindo contexto geográfico, histórico, preferências do utilizador, dados sociais, entre outros. Neste caso de estudo, concentramo-nos nos tipos de dados textuais e geográficos para complementar os dados históricos, como os *check-ins* e as preferências dos utilizadores, por exemplo.

O (Werneck et al., 2020) destaca uma lacuna significativa na pesquisa de sistemas de recomendação, evidenciando uma subutilização do uso de dados textuais. Dos 73 estudos selecionados para a análise, apenas 7 deles incorporavam dados textuais como parte do processo de recomendação. Esses resultados sugerem que, apesar do vasto potencial dos dados textuais para enriquecer a qualidade e a personalização das recomendações, existe uma escassez de projetos que aproveitam plenamente esse recurso.

Esta falta de utilização dos dados textuais pode ser atribuída a várias razões, como a complexidade na análise de texto não estruturado, a necessidade de técnicas avançadas de processamento de linguagem natural e a limitada disponibilidade de conjuntos de dados textualmente ricos. No entanto, é crucial reconhecer o valor que os dados textuais podem acrescentar aos sistemas de recomendação.

No âmbito deste estudo, contemplamos a filtragem por conteúdo, onde foram analisados projetos semelhantes (descritos na seção 3.5) para extrair ideias para a implementação de um sistema híbrido.

2.11 Áreas de Aplicação

Existem diversas áreas de aplicação para sistemas de recomendação, abrangendo uma grande diversidade de itens a serem recomendados. Embora a utilidade típica dos sistemas de recomendação seja na área do comércio eletrónico, como na compra

e venda de produtos, livros, filmes, viagens e outros serviços, esses sistemas têm se expandido para outras áreas e contextos.

Aqui estão alguns exemplos de empresas que utilizam mecanismos de recomendação:

- **Amazon.com**[14]: A Amazon.com utiliza recomendações de filtragem colaborativa item a item na maioria das páginas do seu site, bem como em campanhas por e-mail. a evolução do algoritmo de recomendação da Amazon, que começou com abordagens simples baseadas em classificações de produtos e evoluiu para modelos avançados de aprendizagem computacional. O algoritmo utiliza dados como o histórico de compras e visualizações para oferecer recomendações personalizadas. Este desenvolvimento reflete o foco da Amazon em melhorar a experiência do cliente através da inovação tecnológica(Larry Hardesty, 2019).
- **Netflix**[15]**:** A Netflix é outra empresa orientada por dados que utiliza sistemas de recomendação para aprimorar a satisfação do cliente. A Netflix também promoveu competições de ciência de dados, como o Netflix Prize (Maroto Mariana, 2021), onde o algoritmo de recomendação de filmes mais preciso poderia ganhar um prêmio de $1.000.000.
- **Spotify**[16]**:** O Spotify é uma aplicação que cria listas de reprodução personalizadas para cada assinante, incluindo a conhecida "Discover Weekly". Esta lista de reprodução é composta por 30 músicas personalizadas com base nos gostos musicais únicos de cada utilizador, como explicado por (Velardo V., 2019) na sua publicação.

Esses exemplos destacam a ampla gama de aplicações para sistemas de recomendação em várias indústrias e como eles desempenham um papel significativo na satisfação do cliente e no aumento das vendas. A precisão e eficácia desses sistemas são fatores críticos para o sucesso das empresas que os utilizam.

2.12 DBSCAN

O DBSCAN[17] (Density-Based Spatial *Clustering* of applications with Noise o significado do seu acrónimo em inglês) é um algoritmo de *clustering* baseado na

[14] Marketplace global para compras online e serviços digitais. https://www.amazon.com/ (último acesso em janeiro 2025)
[15] Plataforma de vídeos e entretenimento sob demanda. https://www.netflix.com/ (último acesso em janeiro 2025)
[16] Serviço de música e podcasts online. https://open.spotify.com/ (último acesso em janeiro 2025)
[17] Algoritmo para agrupamento de dados https://scikit-learn.org/1.5/modules/generated/sklearn.cluster.DBSCAN.html (último acesso em janeiro 2025).

densidade amplamente utilizado para segmentação de dados espaciais. Diferente de métodos como K-Means, que exigem um número pré-definido de *clusters*, o DBSCAN identifica agrupamentos naturais nos dados ao considerar a densidade dos pontos em um determinado espaço. Ele agrupa pontos que estão suficientemente próximos entre si e marca os restantes como ruído. Essa característica torna o algoritmo particularmente útil para conjuntos de dados espaciais irregulares, onde os *clusters* podem ter formas arbitrárias.

Este algoritmo funciona com dois parâmetros principais:

- **Epsilon (ε)**: distância máxima entre dois pontos para que sejam considerados vizinhos.
- **MinPts**: número mínimo de pontos necessários para que uma região seja considerada um *cluster*.

O DBSCAN tem sido amplamente aplicado na análise de dados geoespaciais e na segmentação de locais em sistemas de recomendação. Estudos prévios (Wang et al., 2017), explicado no subcapítulo 3.6 demonstraram que a incorporação de agrupamentos espaciais pode melhorar a qualidade das recomendações, permitindo segmentar POIs em áreas de interesse comuns.

No contexto deste estudo, o DBSCAN foi utilizado para agrupar os POIs com base na sua proximidade geográfica, facilitando a organização e análise dos dados. Ao aplicar o algoritmo às coordenadas de latitude e longitude, os POIs foram segmentados em grupos que representam áreas de maior concentração de locais de interesse. Esta abordagem foi essencial para a melhoria da qualidade das recomendações, pois permitiu que o sistema considerasse não apenas as preferências do utilizador, mas também a distribuição espacial dos locais sugeridos.

Nos capítulos subsequentes, serão apresentados estudos que utilizam o DBSCAN no contexto da recomendação de POIs, permitindo uma comparação com os resultados obtidos neste projeto.

3 ESTADO DA ARTE

Neste capítulo, apresenta-se uma revisão dos estudos relacionados a sistemas de recomendação de POIs, baseada em uma pesquisa detalhada de trabalhos científicos e propostas recentes na área. Foram analisados os métodos adotados em diferentes abordagens, destacando as estratégias utilizadas para aprimorar a precisão e relevância das recomendações.

Existem diversos sistemas de recomendação de POIs propostos, baseados em diferentes tipos de dados, que aplicam métodos aprimorados e novos algoritmos com o objetivo de melhorar a eficiência dessas recomendações. Como principais tendências de pesquisa, destaca-se a convergência do reconhecimento de dados sociais (de grande importância devido à partilha de interesses comuns, amigos, familiares, etc.), dados temporais e dados geográficos como elementos essenciais para a recomendação de POIs.

Um dos principais focos da investigação neste projeto é o contexto em formato textual, como mencionado anteriormente. É importante observar que há poucos sistemas que fazem uso efetivo desses dados textuais (Werneck et al., 2020). No entanto, é possível que esses dados textuais desempenhem um papel fundamental para tornar as recomendações mais precisas, pois complementam as informações desejadas. Além das preferências e interesses pessoais, o comportamento do utilizador é influenciado e, em muitos casos, limitado por preferências locais e contextuais (Aliannejadi & Crestani, 2018). Os dados textuais, ao capturarem descrições detalhadas do contexto, podem complementar essas preferências, permitindo recomendações mais personalizadas e ajustadas às necessidades específicas do utilizador.

Na criação de um sistema de recomendação, é de máxima importância destacar o tipo de dado que será utilizado e a fonte dessas informações. Diversas plataformas podem ser utilizadas para a aquisição de dados. Autores como (Xiong et al., 2020) mencionam as redes heterogéneas denominadas LBSN (Redes Sociais Baseadas em Localização, em inglês, *Location-Based Social Networks*) e CBSN (Redes Sociais Baseadas em Comunicação, em inglês, *Communication-Based Social Networks*). A diferença fundamental entre essas redes sociais, segundo os autores, é que as LBSNs concentram-se principalmente nas informações do perfil do utilizador e nos registos de *check-ins*, o que pode resultar em uma escassez de dados sobre as preferências do utilizador e que torna as redes CBSN mais confiáveis, em termos de informações, é que além dos dados históricos e preferências, há a possibilidade de extrair dados de relacionamentos próximos com outros utilizadores (informações de amigos confiáveis). Portanto, (Xiong et al., 2020) utilizam o Foursquare como uma rede LBSN, e o Facebook e o Twitter como redes CBSN. A confiabilidade das informações desempenha um papel crucial, uma vez que as redes sociais se baseiam em comportamentos e interações do utilizador, que geram comentários subjetivos sobre um POI para expressar as suas preferências. Eles explicam que a

recomendação de POIs em LBSNs concentra-se principalmente em reproduzir vários fatores extraídos dos perfis dos utilizadores e dos registos de *check-in*.

Com base numa análise aprofundada do estado da arte em sistemas de recomendação de POIs, este capítulo apresentará uma pesquisa abrangente de casos de estudo. A investigação será focada na identificação e discussão das principais características implementadas em cada projeto, com ênfase nos métodos, algoritmos e variáveis utilizadas. Este levantamento permitirá compreender as abordagens inovadoras, os desafios enfrentados e as soluções aplicadas, fornecendo uma visão estruturada sobre o desenvolvimento de sistemas de recomendação.

3.1 Recomendação de POIs

Quando se trata de sistemas de recomendação de Pontos de Interesse, é importante considerar que as recomendações genéricas muitas vezes se baseiam na popularidade e, na maioria dos casos, no histórico ou na distância dos *check-ins* (Guo et al., 2017a) e (Chang et al., 2018a).

Essas recomendações genéricas tendem a não ser personalizadas, levando à sugestão do mesmo POI para todos os utilizadores. No entanto, as recomendações personalizadas são adaptadas de acordo com as necessidades e preferências individuais de cada utilizador, resultando em sugestões de diferentes POIs para cada um (Medeiros, 2020).

Normalmente, um sistema de recomendação tem acesso a um conjunto de itens e utilizadores, e o seu principal objetivo é sugerir itens que provavelmente interessarão aos utilizadores. No entanto, adaptar sistemas de recomendação para recomendações de localização requer algumas premissas. Por exemplo, um utilizador que está de férias numa cidade que não é a sua cidade natal terá interesses que variam com base na geolocalização e no motivo da visita àquela cidade (negócios, lazer, etc.).

Essas informações formam um conjunto de dados sobre a qual as recomendações serão construídas, permitindo ao sistema compreender e prever as preferências dos utilizadores de forma precisa. A seguir alguns elementos-chave que os conjuntos de dados devem incluir:

- Descrições Detalhadas dos POIs: Informações detalhadas sobre cada ponto de interesse, incluindo localização, descrição, categoria (como museus, restaurantes, parques, etc.), horário de funcionamento e características únicas.
- Avaliações e Classificações dos Utilizadores: Avaliações e classificações fornecidas por utilizadores anteriores sobre os POIs. Estas avaliações podem ajudar a determinar a qualidade e a popularidade de um POI, influenciando assim as recomendações feitas pelo sistema.
- Histórico de Visitas dos Utilizadores: Registos históricos das visitas dos utilizadores a diferentes POIs. Isto permite ao sistema entender as

preferências individuais dos utilizadores e oferecer recomendações personalizadas com base nos seus interesses e comportamentos passados.

- Informações Contextuais: Informações contextuais, como a época do ano, o clima, eventos locais e preferências sazonais, podem ser incorporadas ao algoritmo de recomendação para fornecer sugestões relevantes e oportunas.

Ao garantir que o conjunto de dados esteja praticamente completo, o sistema de recomendação pode oferecer sugestões precisas e relevantes de POIs, proporcionando assim uma experiência satisfatória aos utilizadores.

3.2 Objetivos

Nesta secção, exploram-se os objetivos dos estudos relacionados com sistemas de recomendação de POIs. A influência geográfica é destacada como uma das características mais importantes para estes sistemas, sendo amplamente abordada nos artigos analisados. A maioria dos estudos selecionados concentrou-se em enfrentar o desafio tradicional de melhorar a precisão das recomendações, utilizando diversos tipos de informações contextuais e heterogéneas (Gao et al., 2015a), (Guo et al., 2017b), (Wang et al., 2017), (Chang et al., 2018b), (Aliannejadi & Crestani, 2018).

Um dos principais desafios identificados foi a escassez de dados provenientes da interação dos utilizadores com os POIs, especialmente em cenários fora das áreas habituais de visita do utilizador. Esta escassez torna-se ainda mais relevante em contextos onde o utilizador não possui histórico de visitas numa nova localização. Neste sentido, muitos estudos enfatizam a necessidade de considerar preferências locais e contextuais, além das preferências pessoais do utilizador.

(Guo et al., 2017) investigam a recomendação de POIs com base na influência geo-social, destacando a integração de dados provenientes de *check-ins*, distâncias geográficas e conexões sociais. O modelo proposto foi desenvolvido para superar as limitações dos métodos tradicionais de filtragem colaborativa, especialmente em cenários caracterizados por dados esparsos. Além disso, o estudo sublinha a importância de abordar a heterogeneidade dos dados para melhorar a precisão das recomendações.

A geração de palavras-chave ou tags foi um dos objetivos dos estudos de (Liang & Wang, 2018) e (Xiong et al., 2020) para complementar a informação de preferências do utilizador. Isso está relacionado ao tema das pesquisas de rotas ótimas, onde os autores utilizaram o modelo proposto por (Zeng et al., 2015), que consiste em um algoritmo de otimização baseado em programação dinâmica para gerar palavras-chave a partir de dados textuais e contextuais dos utilizadores. Este modelo foi projetado para identificar termos mais relevantes para descrever POIs, levando em consideração a frequência e a relevância semântica das palavras em diferentes contextos.

(Carusotto et al., 2021) consideraram a informação mais atualizada após a pandemia (Covid-19) e a preferência do utilizador como fator central, mas não o único aspeto crítico. Outros fatores, como regulamentações locais, cautela do utilizador ou aglomeração de POIs, começam a surgir devido a preocupações com a saúde.

3.3 Plataformas de Dados

Na pesquisa sobre recomendação de POIs, várias plataformas são amplamente utilizadas para a aquisição de dados. Entre as escolhas mais tradicionais estão Yelp[18], Foursquare[19] e Gowalla[20], conhecidas por disponibilizarem informações valiosas para esse domínio (Guo et al., 2017), (Guo et al., 2017), (Wang et al., 2017), (Aliannejadi & Crestani, 2018), (Xiong et al., 2020).

Estudos significativos, como o de (Aliannejadi & Crestani, 2018) preferiram utilizar o Foursquare e o Yelp para coletar informações cruciais, como avaliações, categorias e palavras-chave, No entanto, o projeto TREC-CS 2016 (Biega et al., 2020) introduziu dados adicionais não comuns em outras plataformas, como o contexto da viagem (por exemplo, negócios ou lazer) e as preferências contextuais dos utilizadores por exemplo, motivo da visita: o utilizador está em viagem de trabalho ou lazer, esses dados permitiram um ajuste mais preciso das recomendações, levando em conta fatores como a finalidade da visita e o ambiente local, enriquecendo assim as sugestões de POIs.

Carusotto et al., (2021), exploraram o Yelp Open Dataset[21], uma coleção abrangente que inclui dados sobre negócios, avaliações e informações de utilizadores, destacando a relevância dessa informação para pesquisas contemporâneas.

Além das plataformas tradicionais, pesquisas recentes como as de Guo et al., (2017) e Wang et al. (2017) combinaram o uso do Yelp com o Foursquare devido às semelhanças dessas plataformas em termos de dados disponíveis, incluindo informações de utilizador, POI e check-in.

[18] Plataforma de avaliações de negócios locais. https://www.yelp.com (último acesso janeiro 2025)
[19] ecomendação e análise de dados de localização. https://www.foursquare.com (último acesso janeiro 2025)
[20] App de check-ins e descoberta de locais. https://www.gowalla.com/ (último acesso janeiro 2025)
[21] Dados abertos de avaliações e negócios. https://business.yelp.com/data/resources/open-dataset/ (último acesso janeiro 2025).

Outras abordagens foram analisadas por (Xiong et al., 2020) e (Chang et al., 2018), que analisaram redes sociais como Facebook[22], Twitter (atualmente X[23]) e Instagram[24] para extrair informações relevantes, tais como opiniões sobre experiências em locais específicos, atividades realizadas e preferências dos utilizadores. Estas plataformas facilitam a análise de características de POIs através do conteúdo gerado nas publicações, fornecendo dados como categorias dos POIs e informações contextuais dos utilizadores. Por falta de bases de dados adequadas, foi construído um novo conjunto contendo textos extraídos do Instagram, uma plataforma destacada por (Chang et al., 2018) pela sua popularidade e pelo volume diário significativo de *check-ins* e conteúdos textuais associados a localizações.

3.4 Tipos de Dados

Na construção de sistemas de recomendação de POIs, diversos tipos de dados são utilizados. Alguns autores optam por utilizar palavras-chave, cinco deles (Gao et al., 2015), (Guo et al., 2017), (Aliannejadi & Crestani, 2018), (Chang et al., 2018) e (Xiong et al., 2020) em particular, sobressaem ao utilizarem esse recurso com o objetivo de otimizar a identificação e seleção dos interesses dos utilizadores.

Aliannejadi & Crestani, (2018), propõem uma abordagem probabilística na qual mapeiam palavras-chave de geolocalização tais como pontos turísticos, nomes de cidades ou locais populares, para as *tags*[25] utilizadas pelos utilizadores. Eles identificaram uma correlação entre o conteúdo do local e as tags do utilizador. Em contrapartida, (Chang et al., 2018) utilizaram conteúdo de redes sociais do tipo CBSN, como o conjunto de dados adequado ao modelo não estava disponível, eles construíram um novo conjunto de dados contendo conteúdo de texto relacionado a POIs, com base no texto escrito pelos utilizadores no Instagram.

No mesmo estudo de (Chang et al., 2018), foram analisadas as relações entre palavras e POIs para determinar a importância do conteúdo textual na representação desses POIs em sistemas de recomendação. Esse conjunto de dados abrange a cidade de Nova York e contem POIs filtrados com um mínimo de 5 *check-ins*, este critério foi utilizado como um filtro para garantir que, os POIs incluídos no conjunto de dados sejam significativamente frequentados pelos utilizadores, totalizando 2.216.631 *check-ins* em 13.187 POIs de 78.233 utilizadores. Para avaliar a influência geográfica, foram criados dois conjuntos de informações históricas. O primeiro conjunto

[22] https://www.facebok.com/ Rede social para conexão, compartilhamento e comunicação. (último acesso janeiro 2025)

[23] https://x.com/ Plataforma de microblogging para publicações curtas e interações em tempo real. (último acesso janeiro 2025)

[24] https://www.instagram.com/ (último acesso janeiro 2025)

[25] Palavras-chave ou etiquetas usadas para categorizar. https://www.tecmundo.com.br/navegador/2051-o-que-e-tag-.htm (último acesso janeiro 2025).

utilizava todos os *check-ins* de cada utilizador, enquanto o segundo continha todos os *check-ins* sequenciais de cada POI, organizados por dia.

(Liang & Wang, 2018) adotaram uma função de cobertura para medir o grau em que cada consulta correspondia a uma rota e às categorias dos locais. Eles aplicaram um método de (Zeng et al., 2015) que também se dedicou a resolver problemas de planeamento de rotas e deu destaque especial às palavras-chave associadas a POIs georreferenciados, localização e texto.

Os dados geográficos são amplamente utilizados para capturar padrões e relações em sistemas de recomendação (Guo et al., 2017), propuseram o uso de um grafo heterogéneo para representar as interações entre utilizadores e locais, atribuindo pesos às arestas baseados em critérios como frequência de visitas e proximidade geográfica. Por outro lado, (Wang et al., 2017) dividiram o espaço geográfico em regiões e modelaram padrões de mobilidade dos utilizadores através de distribuições multinominais, permitindo compreender como eles interagem com diferentes áreas ao longo do tempo.

(Gao et al., 2015b) utilizaram análise de sentimento dos utilizadores e, por meio de uma escala para ações de *check-in* observadas, aumentaram ou diminuíram a importância com base numa escala de indicação positivo ou negativo. Essa escala era fundamentada em uma pontuação obtida de um dicionário léxico de sentimentos.

(Wilson et al., 2005), como parte de seu conjunto de dados, extraíram características das avaliações e aplicaram o método de (Zeng et al., 2015) , que considerava as preferências dos utilizadores na busca por frases e introduziram uma função de cobertura de palavras-chave que ampliava a precisão das recomendações.

Do mesmo modo, (Aliannejadi & Crestani, 2018) introduziram o conceito de relevância de uma categoria de POI para uma dimensão contextual que varia de -1 a +1, com valores positivos e negativos. Eles distinguiram duas classes de características: objetivas, como a proximidade de um restaurante a uma estação de transporte público, que pode influenciar diretamente a decisão de um utilizador de o visitar; e subjetivas, como a atmosfera acolhedora ou a decoração de um restaurante, que dependem principalmente da opinião e das preferências pessoais do utilizador.

O conjunto de dados da Yelp consiste em vários ficheiros JSON (negócio, utilizador, avaliação, *check-in*, opinião e foto). (Carusotto et al., 2021) se concentraram nos ficheiros relacionados a "negócio" e "avaliação". Dado que o ficheiro de avaliações continha textos em diferentes idiomas, eles utilizaram o "*langdetect*", extensão da biblioteca Spacy para detetar a linguagem da variável (explicado no subcapítulo 4.3.1), para selecionar apenas o conteúdo em inglês. Após o processamento de dados e uma filtragem, foi extraído um conjunto de dados que incluía as avaliações dos dez utilizadores mais ativos, ou seja, aqueles que publicaram um maior número de avaliações.

Por último (Xiong et al., 2020) utilizaram três conjuntos de dados, sendo um do tipo LBSN baseado na plataforma Foursquare, que incluía características temporais como horário dos *check-ins* e coordenadas geográficas. Os outros dois conjuntos de dados pertenciam ao género CBSN e provinham do Twitter e do Facebook. Os dados do Twitter abrangiam informações de 28.553 utilizadores e 186.589 comentários relacionados. No caso do Facebook, foram coletados dados de 52.772 utilizadores e 378.117 comentários relevantes sobre POIs. Este artigo adotou uma metodologia que aparentemente invadia informações de privacidade. Portanto, todos os nomes de POIs e de utilizadores foram convertidos em números, assim como as coordenadas de latitude e longitude dos POIs foram ajustadas para coordenadas relativas. A conexão entre essas redes deu origem a duas redes heterogéneas, a rede Foursquare-Twitter e a rede Foursquare-Facebook, por meio de links âncora caso os utilizadores tivessem contas interligadas no Foursquare.

3.5 Métodos Adotados

Os estudos relacionados propõem alterações ou melhorias nos modelos de POI, com destaque para as técnicas de filtragem colaborativa e filtragem baseada em conteúdo, que são amplamente utilizadas. Mesmo quando esses métodos não são usados no processo de recomendação, são frequentemente empregues para fins de comparação na fase de avaliação, como será explicado na seção 3.6.

Os estudos selecionados para este capítulo do estado da arte fazem uso do tipo de dado textual, e o método de filtragem baseada em conteúdo predomina na maioria dos casos. No entanto, isso não significa que o método de filtragem colaborativa tenha sido negligenciado. Na verdade, ambos os métodos foram aplicados em trabalhos que serão brevemente descritos a seguir.

(Gao et al., 2015b) enfatizaram que diferentes tipos de conteúdo de informação disponíveis em LBSN podem estar relacionados a diferentes aspetos da ação de *check-in*. Para atender às exigências do processo de recomendação de POIs, foi utilizado a filtragem de conteúdo, combinada com a aplicação de um método de factorização de matrizes de menor dimensão (por exemplo, os utilizadores e os POIs), o que permitiu melhorar a precisão e a personalização das recomendações.

(Guo et al., 2017) combinaram técnicas de filtragem colaborativa com abordagens baseadas em grafos para recomendar POIs. Utilizaram filtragem colaborativa baseada em utilizadores (UCF o seu acrónimo em inglês) e filtragem colaborativa baseada em itens (ICF o seu acrónimo em inglês), onde as recomendações foram geradas com base na similaridade entre utilizadores e entre itens. Para personalizar ainda mais as recomendações, aplicaram técnicas de análise de grafos, especificamente através de uma adaptação do método *Personalized PageRank (*PPR o seu acrónimo em inglês*),* que considera as relações de utilizadores e POIs numa matriz de conexões. Além disso, exploraram relações sociais e padrões de *check-ins*

para ajustar as recomendações, utilizando a similaridade de comportamento e as conexões espaciais entre os utilizadores.

(Wang et al., 2017) adotaram uma abordagem híbrida para a recomendação de locais de interesse, que combina técnicas de filtragem colaborativa com análise de sentimentos e influências geográficas. A abordagem integra a filtragem colaborativa tanto baseada em utilizadores como em itens, ajustando as recomendações com base na localização geográfica dos utilizadores, considerando tanto os interesses individuais como as preferências coletivas de utilizadores em diferentes áreas.

(Xiong et al., 2020) adotam o método de filtragem colaborativa, tanto baseada em utilizadores quanto baseada em itens, explorando as interações sociais e os comportamentos dos utilizadores em relação aos POIs. Além disso, adotam a análise de conteúdo textual, especificamente de comentários e interações nas redes sociais, como parte do processo de recomendação.

3.6 Métricas de Avaliações

Quando é avaliado um sistema de recomendação, o objetivo principal é determinar o quão próximas as previsões estão das escolhas reais dos utilizadores (Gao et al., 2015a). Para além das métricas tradicionais, como Precisão, *Recall*, Acurácia ou AUC, também são utilizadas abordagens baseadas em distância geográfica, especialmente em sistemas de recomendação de POIs.

A maioria dos estudos selecionados utilizou métricas populares, como Acurácia, Precisão e *Recall***,** para avaliar o desempenho dos sistemas de recomendação (Gao et al., 2015a), (Guo et al., 2017b),(Wang et al., 2017) e (Xiong et al., 2020).

No caso de (Guo et al., 2017b) destacaram a importância de integrar a distância geográfica na avaliação. Os autores utilizaram métricas como *Precisão@* k *e Recall@* k*,* que avaliam a proporção de POIs relevantes (por exemplo, próximos ao local visitado). Essas métricas permitiram medir tanto a relevância quanto a proximidade geográfica dos POIs recomendados, essenciais em sistemas que integram contexto espacial.

No estudo de (Aliannejadi & Crestani, 2018) a eficácia das recomendações, a redução da dimensionalidade para a recomendação e a eficácia dos métodos de previsão de tags do utilizador foram avaliadas. Para garantir comparações justas, foi escolhido um protocolo de avaliação oficial, o *TREC-CS*[26] (Text REtrieval Conference Contextual Suggestion a sua denominação em inglês), e considerou as seguintes métricas de avaliação: Precisão_$@k$, nDCG_$@k$ (*Normalized Discounted Cumulative Gain* significado do seu acrónimo em inglês) e MRR (*Mean Reciprocal Rank* significado do seu acrónimo em inglês). Precisão_$@k$ já explicado na equação (2.11) mas calculado por (Aliannejadi & Crestani, 2018) de forma particular na equação (3.1), e

[26] https://sites.google.com/site/treccontext/ (último acesso em janeiro 2025).

nDCG_@ (Aliannejadi & Crestani, 2018) , foi calculado na k-ésima posição tal como se mostra na equação (3.2):

$$P_u@k = \frac{\#hits_u@k}{k} \tag{3.1}$$

$$nDCG_u@k = Z_u \sum_{i=1}^{k} \frac{2^{r_u^i} - 1}{\log(1+i)} \tag{3.2}$$

Onde:

u: o conjunto de utilizadores,

Z_u: um fator de normalização, que garante que o valor máximo do DCGseja 1

$\#hits_u@k$: o número de localizações para o utilizador nas primeiras k localizações da lista classificada.

$\sum_{i=1}^{k}$: soma dos termos relacionados à relevância dos k itens recomendados.

r_u^i: grau de relevância do item na i-ésima posição para o utilizador u.

$log(\mathbf{1}+\boldsymbol{i})$: função de penalização logarítmica que reduz o peso de itens relevantes em posições mais baixas da lista.

(Aliannejadi & Crestani, 2018), calcularam MRR como se descreve na equação (3.3):

$$MRR = \frac{1}{|U|} \sum_{u=1}^{|U|} \frac{1}{rank_u} \tag{3.3}$$

Onde:

$rank_u$: é a classificação do primeiro local relevante para o utilizador u.

(Guo et al., 2017) consideraram utilizar *Precisão*@k *e, Recall*@k onde k é o tamanho da lista de classificação de POI recomendada (*top*-k),

Onde foi definido:

k: 5, 10 e 20.

A métrica média de todos os utilizadores é adotada como resultado.

A mesma estratégia também foi realizada no caso do (Gao et al., 2015b), mas desta vez definiram:

k*:* 5 e 10.

No caso de (Chang et al., 2018) que também optaram por *Recall*@k and Mean Reciprocal Rank (MRR), onde definiram:

k = 1, 5, e 10.

(Liang & Wang, 2018) utilizaram as métricas: Ganho (*Gain* em inglês) para eficácia como se mostra na equação (3.4), tempo de execução da CPU e espaço de busca (num número de rotas abertas).

$$Gain = \sum_{i=1}^{N} \frac{r_i}{log_2(1+i)} \tag{3.4}$$

Onde:

N: número total de rotas ou itens avaliados.

r_i: relevância do item *i* (geralmente definida como 1 para itens relevantes e 0 para irrelevantes).

$log_2(1+i)$: penalização logarítmica para posições mais baixas no *ranking*, atribuindo maior peso a itens no topo da lista.

(Xiong et al., 2020) implementaram métricas de avaliação do tipo Acurácia (*Accuracy* $@k$) de acordo com uma estrutura metodológica (Hu & Martin, 2013a) com a utilização de validação cruzada (*cross-validation*[27]) de 10 vezes, calculado na equação (3.5):

$$Accuracy@k = \frac{hit@k}{D_{test}} \tag{3.5}$$

Onde $hit@k$[28] e D_{test} representam o número de ocorrências no conjunto de teste e o número de todos os casos de teste, respetivamente. Especificamente, foi dividido o conjunto de dados em subconjuntos com base nos *clusters* após o agrupamento de locais. O *clustering* foi realizado com base em características geográficas (latitude e longitude), utilizando algoritmos como *k-means* ou *DBSCAN*, que são comuns em sistemas de recomendação geoespacial. Um subconjunto com um número comparável de localizações envolve apenas *clusters* adjacentes, garantindo que as recomendações sejam consistentes dentro de áreas geográficas específicas. Esta métrica também foi utilizada por (Wang et al., 2017).

[27] Técnica para avaliação de desempenho de modelos de aprendizagem computacional. https://medium.com/@edubrazrabello/cross-validation-avaliando-seu-modelo-de-machine-learning-1fb70df15b78 (último acesso janeiro 2025)

[28] https://pykeen.readthedocs.io/en/stable/api/pykeen.metrics.ranking.HitsAtK.html (último acesso janeiro 2025)

3.7 Algoritmos Propostos

Neste subcapítulo apresentam-se os principais algoritmos propostos para sistemas de recomendação de POIs, destacando as abordagens inovadoras que integram informações contextuais, geográficas e textuais. Cada um dos modelos analisados reflete estratégias específicas para otimizar a recomendação, adaptando-se às necessidades e preferências dos utilizadores.

(Aliannejadi & Crestani, 2018) apresentam um modelo probabilístico que mapeia tags de utilizadores para palavras-chave de preferência de localização. Foram propostos quatro modelos que utilizam diferentes combinações de *ratings* de similaridade entre utilizador-POI:

- UT-CRF (*User tag - Conditional Random Fields* significado do seu acrónimo em inglês) prevê *tags* de utilizador utilizando um modelo *tagger* sequencial baseados em recursos binários que são extraídos do texto (Lafferty et al., 2001). Então, para cada par utilizador-POI, ele calcula a similaridade entre as tags do utilizador previstas e o perfil do utilizador. Depois é substituída a pontuação inicial pela pontuação de similaridade calculada.
- UT-SVM (*User tag - Support Vector Machines* significado do seu acrónimo em inglês) Este modelo prevê *tags* de utilizador dado um par utilizador-POI usando um modelo de *tagging* baseado em SVM. A pontuação é substituída pela pontuação de similaridade entre o perfil do utilizador e as *tags* previstas.
- UT-ML (*User tag – Machine Learning* significado do seu acrónimo em inglês) Neste modelo, a pontuação é baseada nas *tags* de utilizador previstas seguindo o critério de probabilidade máxima.
- Factorização Tensorial n-dimensional (nDTF o seu acrónimo em inglês) (Karatzoglou et al., 2010) Este modelo generaliza a factorização de matrizes para permitir a integração de vários recursos contextuais no modelo.

Em (Chang et al., 2018) Propõem o CAPE (o acrónimo em inglês de *content-aware* POI *embedding*) que utiliza conteúdo de texto para capturar informações sobre as características de um POI. O CAPE consiste em uma camada de contexto de *check-in* e uma camada de conteúdo textual.

(Liang & Wang, 2018) apresentam um algoritmo de busca de rotas PACER, (***P****refix b****A****sed* ***C****ompact stat****E****s g****R****owth* o significado do acrónimo em Inglês), que incorpora programação dinâmica e estratégias de redução baseadas em custo e ganho em um sistema unificado para busca de rotas.

(Xiong et al., 2020), propõem um modelo de geração probabilística latente denominado HI-LDA (acrónimo de *Heterogeneous Information based LDA*), que que captura as palavras dos utilizadores numa CBSN, considerando informações geográficas, relacionamentos sociais e comentários.

(Guo et al., 2017) propõem o modelo AGSG (acrónimo de *Aspect-aware Geo-Social in-fuence Graph*) que une várias relações entre utilizadores, POIs e aspetos das avaliações dos utilizadores, explorando uma estrutura heterogénea de informação.

(Wang et al., 2017) propõem um modelo generativo probabilístico latente chamado LSARS (acrónimo de *Location-Sentiment-Aware Recommender System*) que otimiza o processo de tomada de decisão dos *check-ins* dos utilizadores, levando em consideração a influência geográfica, conteúdo do POI e opinião do utilizador.

(Gao et al., 2015a) propõem o algoritmo CAPRF (*Content-Aware POI Recommendation Framework* o significado do acrónimo em Inglês), uma estrutura de recomendação que leva em consideração propriedades do POI, interesses do utilizador e indicações de sentimento.

(Carusotto et al., 2021) propõem dois algoritmos de aprendizagem: PV-DM (*Paragraph Vector - Distributed Memory* o significado do acrónimo em Inglês) e PV-DBOW (*Paragraph Vector - Distributed Bag of Word*s o significado do acrónimo em Inglês). variantes do Doc2Vec, que é uma extensão do Word2Vec para representar documentos ou frases em um espaço vetorial contínuo. Enquanto o Word2Vec foca na representação de palavras, o Doc2Vec é projetado para lidar com textos maiores, como parágrafos ou documentos inteiros, capturando a semântica global do texto (Le & Mikolov, 2014).

Além disso, (Carusotto et al., 2021) conduzem experiências variando parâmetros da abordagem Doc2Vec, como o tamanho da janela de contexto, a dimensionalidade dos vetores e o número de iterações de treino, para calcular a similaridade semântica entre as avaliações dos utilizadores. Essa abordagem permite capturar melhor as relações entre os perfis dos utilizadores e os POIs recomendados.

3.8 Avaliações e Métodos para Comparação

Nesta fase, identificam-se os aspetos considerados em métodos de avaliação de usabilidade e experiência, com base em métricas e métodos. Também são mencionados os métodos utilizados para a análise de desempenho.

No caso de estudo de (Aliannejadi & Crestani, 2018), os quatro métodos propostos foram avaliados com o conjunto de dados TREC-CS. Comparativamente, foram confrontados com modelos GeoSoCa (Zhang & Chow, 2015) que exploram correlações geográficas, sociais e categóricas para recomendar POIs, e LinearCatRev (Biega et al., 2020) que extraíram informações de diferentes LBSNs para calcular pontuações baseadas em revisões. A análise também envolveu o uso de Análise de Componente Principal (ACP) para reduzir a dimensionalidade das palavras-chave de localização e componentes de *tags* de utilizadores. Além disso, a avaliação incluiu métodos como CRF (Sang et al., 2012) e SVM-based *Tagger* (Kudo & Matsumoto, 2003).

A filtragem colaborativa foi um método de comparação usado em vários casos de estudo, como o método UPS-CF (*User Preference, Proximity and Social-Based Collaborative Filtering* significado do acrónimo em inglês) em (Xiong et al., 2020), que explora recomendações de localização para utilizadores móveis em LBSN. Outros métodos comparados incluem *ST-LDA LDA* (Yuan et al., 2015) JIM (Hu & Martin, 2013b) e CKNN (Bao et al., 2012). Em alguns casos, a filtragem colaborativa teve desempenho inferior, reforçando a hipótese da dispersão de dados. (Guo et al., 2017) , (Wang et al., 2017) e (Gao et al., 2015a) também utilizaram a filtragem colaborativa em diferentes aspetos, como filtragem colaborativa baseada em factorização de matrizes, mas sem sucesso, recorrendo a outros métodos baseados em grafos.

O método proposto por (Guo et al., 2017), CAPRF, apresentou o melhor desempenho entre todas as abordagens, destacando a importância da informação de conteúdo em LBSNs para a recomendação de POIs. A combinação dos três tipos de conteúdo em formação, ou seja, CAPRF + Indicações de Sentimento + Conteúdo de Interesse do Utilizador + POI-Conteúdo da Propriedade, obteve o melhor desempenho entre todos os outros métodos, destacando o efeito complementar entre os três tipos de informação de conteúdo.

Para comparar os diferentes métodos utilizados nos trabalhos relacionados, foi criada a Tabela 2 que sintetiza os principais resultados das métricas de avaliação reportadas em cada estudo. A Tabela 2, apresenta os resultados de métricas como *Precision*@k, *Recall*@k, MRR, nDCG@k, e Accuracy, bem como outras métricas utilizadas para avaliar os modelos propostos.

Tabela 2 - Resultados das Métricas de Avaliação por Trabalho

Métricas	(Gao et al., 2015a)	(Guo et al., 2017)	(Wang et al., 2017)	(Aliannejadi & Crestani, 2018)	(Chang et al., 2018)	(Xiong et al., 2020)	(Carusotto et al., 2021)
Precision@5	0.85	0.74	0.72	0.81	0.82	0.78	
Precision@10	0.81	0.7	0.68	0.77	0.78	0.74	
Precision@20	0.76	0.66	0.64	0.73	0.74	0.7	
Recall@5	0.67	0.62	0.63	0.68	0.69	0.65	
Recall@10	0.75	0.65	0.67	0.72	0.71	0.69	
Recall@20	0.82	0.71	0.7	0.76	0.73	0.72	
MRR		0.48		0.32	0.35	0.36	
nDCG@5				0.74	0.77	0.73	
Outras Métricas							Similarity, Jaccard

Os resultados refletem a diversidade de métodos e métricas utilizadas nos estudos relacionados. cada modelo prioriza diferentes aspectos, como a Precisão e *Recall*, dependendo da abordagem adotada. Modelos contextuais e híbridos demonstram melhor desempenho geral em rankings e recuperação de POIs relevantes.

Com o fim de aproveitar a máxima informação dos artigos selecionados, foi criada a Tabela 3 com as características principais dos estudos relacionados.

Tabela 3 - Características principais dos artigos selecionados

Características			Trabalhos Relacionados						
			(Gao et al., 2015a)	(Guo et al., 2017)	(Wang et al., 2017)	(Aliannejadi & Crestani, 2018)	(Chang et al., 2018)	(Xiong et al., 2020)	(Carusotto et al., 2021)
Plataformas para obter conjuntos de dados	LBSN	Forsquare	■		■	■		■	
		Yelp		■	■	■			■
	CBSN	Facebook						■	
		Twitter						■	
		Instagram					■		
		outros				TREC			
Tipo de Dado utilizado	Palavras-chave		■	■		■	■	Inf. Sensível	
	Categoria do POI		■	■		■		■	■
	Review/opinião		■	■	■	■	■	■	■
	Check-ins		■	■	■	■	■	■	■
	Contexto Temporal				■	■	■		
	Geolocalização			■	■	■	■	■	
	Avaliação utilizador								■
Tipo de filtragem	Filtragem colaborativa		■	■	■		■	■	■
	Filtragem por conteúdo		■	■	■	■	■	■	■
Métricas de avaliação	Precisão_k@		5,10	5,10 e 20	■	5			
	Recall@k		5,10	■			1,5 e 10		
	MRR					■	■		
	Normalized Discounted					5			
	Accuracy				■			■	
	Outras *(Tempo de execução da CPU e espaço de pesquisa)*							■	Similaridade Cosine «/Jaccard Coefficient

(Gao et al., 2015a)	(Guo et al., 2017)	(Wang et al., 2017)	(Chang et al., 2018)	(Aliannejadi & Crestani, 2018)	(Xiong et al., 2020)	(Carusotto et al., 2021)
Content-Aware Point of Interest Recommendation on Location-Based Social Networks	Aspect-aware Point-of-Interest Recommendation with Geo-Social Influence Qing	A location-sentiment-aware recommender system for both home-town and out-of-town users	Content-Aware Hierarchical Point-of-Interest Embedding Model for Successive POI Recommendation Buru	Personalized Context-Aware Point of Interest Recommendation	Where to go: An effective point-of-interest recommendation framework for heterogeneous social networks	User Profiling for Tourist Trip Recommendations using Social Sensing

A Tabela 3 contribui para uma compreensão mais rápida e estruturada das diferentes abordagens e recursos explorados nos estudos analisados, destacando claramente as preferências e tendências atuais no desenvolvimento de sistemas de recomendação para Pontos de Interesse.

4 TECNOLOGIAS E FERRAMENTAS

Neste capítulo são apresentadas as ferramentas e linguagens que tiveram um papel fundamental na elaboração deste trabalho, destacando as aplicações e plataformas amplamente utilizadas e que contribuíram significativamente para alcançar os resultados pretendidos.

Esta secção funciona como um guia para compreender as tecnologias e ferramentas específicas empregues ao longo do processo de desenvolvimento do projeto, abordando detalhadamente a relevância de cada uma e o seu impacto na concretização dos objetivos definidos.

4.1 Python

Neste projeto, a linguagem escolhida para o desenvolvimento do sistema foi o Python[29], uma das linguagens mais amplamente utilizadas no contexto de aprendizagem computacional. Python é uma linguagem de programação de alto nível, dinâmica, interpretada, modular e multiplataforma.

Python é conhecido pela sintaxe relativamente simples e fácil compreensão, o que o torna popular entre profissionais da indústria tecnológica que podem não ser programadores dedicados, como engenheiros, matemáticos, cientistas de dados, pesquisadores e outros.

Uma das maiores vantagens é sua vasta biblioteca padrão e a disponibilidade de bibliotecas de terceiros. Isso torna Python uma linguagem altamente difundida e útil em uma ampla variedade de setores, incluindo desenvolvimento web, análise de dados, aprendizagem computacional e inteligência artificial (IA).

Python se tornou a linguagem de escolha para muitos profissionais que buscam desenvolver soluções em aprendizagem computacional devido à sua facilidade de uso e à riqueza de recursos disponíveis para essa finalidade. No contexto deste projeto, o Python é fundamental, sendo a principal linguagem utilizada para implementar os algoritmos de recomendação e o processamento PLN. A sua vasta gama de bibliotecas, como o *LightFM* para sistemas de recomendação e ferramentas de PLN como SpaCy explicadas em detalhe no subcapítulo 4.3, facilita a análise eficiente de grandes volumes de dados textuais e numéricos. O ecossistema robusto do Python permite a integração de diferentes abordagens de recomendação e a personalização do sistema, otimizando o desempenho e a precisão das recomendações de POIs.

[29] Linguagem de programação que permite trabalhar rapidamente e integrar sistemas de forma mais eficaz. https://www.python.org/ (último acesso em janeiro 2025.

4.2 Biblioteca LightFM

Foi escolhida a biblioteca *LightFM* [30] devido à sua capacidade de combinar filtragem colaborativa e filtragem por conteúdo, proporcionando um modelo híbrido robusto. A *LightFM* é especialmente eficaz para lidar tanto com feedback implícito quanto com feedback explícito, melhorando a qualidade das recomendações ao aproveitar os pontos fortes de ambas as abordagens (Kapadia, 2020).

Além disso, a *LightFM* oferece suporte para incorporar múltiplas características dos itens e dos utilizadores (*features*), como descrições textuais, categorias ou outros atributos, que enriquecem ainda mais o processo de recomendação. Isso é particularmente útil em contextos onde as preferências do utilizador podem ser complexas ou onde os itens possuem várias dimensões de similaridade.

Também, o uso das suas métricas de avaliação, como AUC, *Precision*@K e Recall@K, foi essencial para medir a precisão e relevância das recomendações, contribuindo para a melhoria contínua do sistema.

A *LightF_Modelevaluation*[31] oferece várias funções para avaliar o desempenho dos modelos de recomendação:

- AUC (Área sob a Curva ROC): Mede a capacidade do modelo em classificar corretamente os itens relevantes acima dos irrelevantes.
- *Precision*@K: Avalia a proporção de itens relevantes entre os top-K itens recomendados.
- *Recall*@K: Mede a proporção de itens relevantes que aparecem nos top-K resultados.
- Reciprocal Rank: Avalia a posição do primeiro item relevante nas recomendações.

Essas métricas são implementadas através de funções *como auc_score, precision_at_k, recall_at_k, e reciprocal_rank*, sendo essenciais para uma análise detalhada da performance do sistema.

4.3 Spacy

SpaCy é uma biblioteca avançada de processamento de linguagem natural desenvolvida por (Honnibal et al., 2020) em Python e Cython. Concebida com base

[30] Biblioteca de recomendação híbrida em Python
https://making.lyst.com/lightfm/docs/home.html (último acesso em janeiro 2025)
[31] Métodos de avaliação de modelos no LightFM
https://making.lyst.com/lightfm/docs/lightfm.evaluation.html (último acesso em janeiro 2025).

nas mais recentes pesquisas da área, foi projetada desde o início para ser utilizada em aplicações reais.

A biblioteca inclui pipelines pré-treinados e suporta atualmente a tokenização e o treino em mais de 70 idiomas. Destaca-se pela sua alta velocidade e pelos seus modelos de rede neuronal de última geração, que abrangem tarefas como marcação, análise sintáctica, reconhecimento de entidades nomeadas, classificação de texto e muito mais. Além disso, spaCy permite a aprendizagem multitarefa utilizando transformadores pré-treinados, como BERT[32] *(Bidirectional Encoder Representations from Transformers* significado do acrónimo em inglês), e oferece um sistema de treino pronto para produção. Facilita também o empacotamento de modelos, a sua implementação e a gestão de fluxos de trabalho, tornando-se uma ferramenta robusta e eficiente para projetos de processamento de linguagem natural.

No projeto, esta biblioteca desempenha um papel fundamental no processamento dos textos contidos no conjunto de dados, utilizado para a transformação de texto em vetores numéricos de alta dimensionalidade, capturando propriedades semânticas das palavras, como significados e relações entre elas.

O SpaCy oferece diferentes modelos, variando no tamanho do vocabulário e na complexidade das representações vetoriais. O número de dimensões do vetor de palavras depende do modelo utilizado:

- Modelos Pequenos (_sm) – Não possuem embeddings vetoriais pré-treinados e dependem apenas de regras sintáticas e gramaticais.
- Modelos Médios (_md) – Incluem embeddings de 300 dimensões, baseados na técnica GloVe, oferecendo um bom equilíbrio entre precisão e eficiência computacional.
- Modelos Grandes (_lg) – Também utilizam 300 dimensões, mas com um vocabulário maior, garantindo melhor cobertura para diferentes domínios.
- Modelos Baseados em Transformers (_trf) – Utilizam embeddings contextuais com 768 ou mais dimensões, permitindo uma melhor compreensão do significado das palavras em diferentes contextos.

No código desenvolvido, os modelos de linguagens **en_core_web_md** e **pt_core_web_md** do spaCy foi carregado e utilizado para processar a informação textuais tais como, nome, descrição, categoria raiz e categoria especifica. Cada valor da informação textual é convertido em um vetor de 300 dimensões, permitindo uma análise mais profunda das semelhanças semânticas entre as informações textuais.

Para cada valor presente no conjunto de dados, foi criado um objeto *doc* com o spaCy, que armazena o vetor associado à palavra, além de outras informações como o valor *norm* (a magnitude do vetor). Se os dados possuíam um vetor associado, esse

[32] https://ubiai.tools/from-words-to-vectors-a-dive-into-spacy-transformers-for-embeddings/ (último acesso em janeiro 2025).

vetor foi armazenado em uma matriz específica. Adicionalmente, o valor **norm** de cada vetor foi armazenado em uma coluna à parte.

4.3.1 Langdetected

A extensão *Langdetected*[33] para a biblioteca spaCy oferece capacidades de deteção automática de idioma no texto processado. Ideal para aplicações que necessitam processar textos em múltiplos idiomas, garantindo que o processamento subsequente é adequado ao idioma identificado.

Pode ser utilizada em diversas áreas, desde análise de sentimentos e classificação de texto até na recomendação de conteúdos e tradução automática.

No projeto o uso da biblioteca *langdetect* desempenha um papel crucial na divisão do dataset com base na identificação de idiomas, particularmente no caso de textos em português e inglês como se descreve no subcapítulo 5.3.2

4.4 Modelo all-MiniLM-L6-v2

O modelo *all-MiniLM-L6-v2* [34] um modelo de transformação de frases baseado em redes neuronais que gera representações vetoriais densas com 384 dimensões. Esse modelo foi escolhido devido ao seu excelente equilíbrio entre eficiência computacional e precisão semântica, tornando-o adequado para processar grandes volumes de texto sem comprometer a performance do sistema.

Diferente de abordagens que segmentam os dados por idioma, optou-se por aplicar o *all-MiniLM-L6-v2* a todo o conjunto de dados disponível, garantindo uma maior cobertura dos POIs recomendáveis e aprimorando a generalização das recomendações. Com isso, o sistema foi capaz de fornecer sugestões mais relevantes, aproveitando a diversidade linguística dos dados sem a necessidade de pré-segmentação.

A escolha deste modelo deve-se à sua capacidade de generalização, que permite lidar eficientemente com descrições textuais em diferentes idiomas e contextos.

4.5 Sweetviz

Sweetviz[35] é uma biblioteca Python de código aberto que gera visualizações apelativas e de alta densidade para impulsionar a Análise Exploratória de Dados

[33] https://pypi.org/project/spacy-langdetect/ (último acesso em janeiro 2025)
[34] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2 (último acesso em janeiro 2025)
[35] https://pypi.org/project/sweetviz/ (último acesso em janeiro 2025).

(EDA - *Exploratory Data Analysis* em inglês) com apenas duas linhas de código. O resultado é uma aplicação HTML totalmente autónoma.

O sistema foi desenvolvido para permitir a visualização rápida de valores-alvo e a comparação de conjuntos de dados. O seu principal objetivo é facilitar a análise rápida de características-alvo, a comparação entre dados de treino e teste, bem como outras tarefas de caracterização de dados.

No âmbito deste projeto, a biblioteca Sweetviz foi utilizada para realizar uma análise e fornecer um relatório detalhado sobre as variáveis do conjunto de dados. O principal objetivo foi compreender a estrutura e distribuição das variáveis para fundamentar a criação de intervalos (bins) explicado no capítulo 5.4, e a transformação de variáveis de informação textual.

4.6 Postman API

O Postman [36]é uma ferramenta popular para testar, desenvolver e documentar APIs. No contexto do sistema de recomendação desenvolvido, o Postman foi utilizado para interagir com a API da Foursquare.

Com esta API, foi possível enviar requisições *HTTP* para acessar os pontos de acesso da API, permitindo a coleta de dados de forma eficiente e automatizada. A plataforma Foursquare oferece uma vasta gama de informações detalhadas sobre os POIs, que foram integradas ao sistema para gerar recomendações personalizadas.

4.7 Google Colab

Para efeitos de partilha Google Colab foi a melhor opção como ferramenta de ambiente colaborativo.

Google Colab[37] é uma ferramenta em nuvem que permite criar e executar códigos na linguagem Python. Com ele, pode executar os programas diretamente do navegador, de forma simples e rápida.

Essa ferramenta oferece um ambiente bastante semelhante ao do software de código aberto Jupyter Notebook[38], com a praticidade de não necessitar configurações já que funciona inteiramente online. Por isso, os códigos criados em ambos são chamados *notebooks*, e são estruturados como um conjunto de células.

---

[36] https://www.postman.com/foursquareapis/foursquare-places-api-v3-public/collection/z7negrl/places-api (último acesso em janeiro 2025)
[37] https://colab.google/ (último acesso em janeiro 2025)
[38] Plataforma open-source que permite a criação e execução de notebooks interativos https://jupyter.org/ (último acesso em janeiro 2025).

As células de um *notebook* podem conter texto explicativo ou código executável, e é possível executar o código de uma célula separadamente ou todas as células de uma só vez. O resultado gerado pela execução é apresentado logo abaixo da célula correspondente, o que torna o estudo ainda mais objetivo, (Noleto C., 2022)

4.8 Spyder (Anaconda)

Spyder[39] (*Scientific Python Development Environment* significado do acrónimo em inglês), é um ambiente de desenvolvimento integrado (IDE o acrónimo em inglês) gratuito que está incluído no Anaconda. Inclui edição, testes interativos, depuração e recursos de introspeção.

No contexto deste projeto, a utilização do Spyder abrange diversas etapas, desde a Aquisição de dados até a implementação de APIs, como a do Foursquare. Isso envolveu a utilização da biblioteca *requests*[40] para enviar solicitações HTTP e receber dados em formato JSON. Os resultados das chamadas de API foram processados e armazenados em estruturas de dados adequadas, permitindo uma análise mais detalhada. Também foi fundamental na fase de pré-processamento dos textos, Scripts foram desenvolvidos para utilizar bibliotecas como *langdetec*t, a fim de identificar o idioma dos textos. Essa informação foi crucial para segmentar os dados em português e inglês.

Assim como também, salvaguarda de Dados resultados, do tipo *dataframes* com as informações processadas, foram guardados em formatos apropriados (CSV, Excel, etc.) para uso posterior ou compartilhamento com outras partes interessadas no projeto.

[39] https://www.spyder-ide.org/ (último acesso em janeiro 2025)
[40] vhttps://pypi.org/project/requests/ (último acesso em janeiro 2025).

5 SISTEMA DE RECOMENDAÇÃO

Neste capítulo, é descrito todo o sistema de recomendação desenvolvido no projeto, incluindo as metodologias aplicadas, ferramentas utilizadas e os passos seguidos para alcançar a solução final. A descrição inclui a arquitetura do sistema, as técnicas de pré-processamento aplicadas, o enriquecimento dos dados e a integração de métodos híbridos no modelo de recomendação.

Adicionalmente, é apresentada a avaliação do sistema com as métricas utilizadas para medir o desempenho das recomendações e garantir que atendam às necessidades dos utilizadores. Resumidamente, este capítulo constitui a base teórica e prática que sustenta o desenvolvimento do sistema de recomendação.

5.1 Arquitetura

A arquitetura proposta é composta por várias etapas integradas. Estas etapas incluem a recolha, processamento, enriquecimento e análise dos dados, bem como a aplicação de algoritmos híbridos para a recomendação POIs. A Figura 5 apresenta a arquitetura do sistema de recomendação desenvolvida neste projeto.

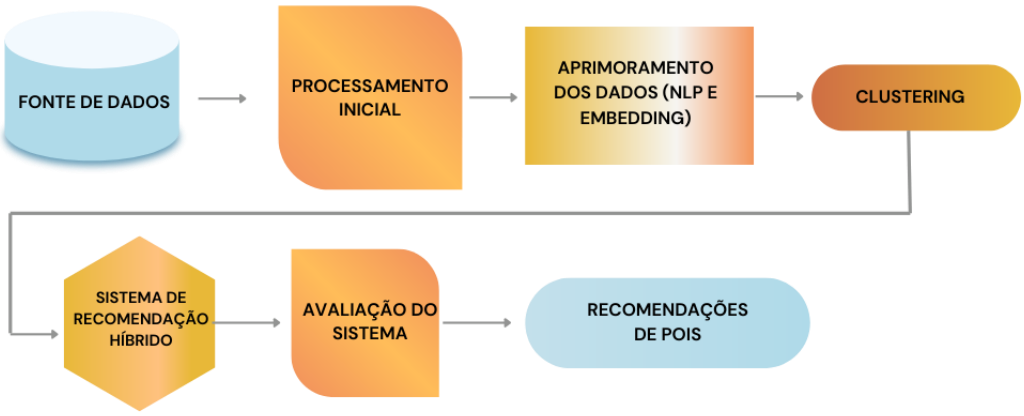


Figura 5 - Arquitetura do sistema

Com base na Figura 5, é possível determinar de forma sequencial as etapas que foram realizadas neste trabalho, e é isso que será explicado neste capítulo. O primeiro passo consiste em recolher os dados, provenientes da plataforma Foursquare, que oferece informações detalhadas sobre POIs, incluindo *check-ins*, descrições, categorias e localizações. Esta escolha foi estratégica devido à ampla cobertura geográfica e diversidade de informações da plataforma.

Após a recolha, os dados passam por uma série de técnicas de processamento inicial, que incluem limpeza, filtragem e organização. Estas técnicas foram aplicadas para garantir que os dados fossem relevantes e consistentes, como a remoção de

categorias irrelevantes e a separação dos textos em português e inglês. Em seguida, foi utilizada a técnica de clusterização espacial (DBSCAN) para agrupar POIs próximos e identificar padrões espaciais.

Posteriormente, os dados foram submetidos a um processo de aprimoramento por meio de técnicas de PLN e Embedding, no qual as informações textuais, como "Nome do Local", "Categoria", "Categoria Raiz" e "Descrição", foram convertidas em vetores numéricos. Esta transformação foi realizada com a biblioteca SpaCy e o modelo pré-treinado *all-MiniLM-L6-v2*, que permitiram capturar as nuances semânticas e criar representações vetoriais consistentes entre idiomas.

Com os dados processados e enriquecidos, foi implementado o sistema de recomendação híbrido, que combina duas abordagens principais:

- Recomendação baseada em conteúdo: Comparação das características dos POIs com as preferências dos utilizadores.
- Filtragem colaborativa: Uso do histórico de interações dos utilizadores para identificar padrões de comportamento.
- A implementação foi feita com a biblioteca *LightFM*, que permite integrar as duas abordagens e explorar os pontos fortes de ambas. O sistema gera recomendações personalizadas de POIs, levando em conta as preferências dos utilizadores.
- Por fim, o sistema foi avaliado com base em diversas métricas, incluindo *Precision@k, Recall@k*, MRR e AUC_ROC, que medem a relevância, recuperação e qualidade das recomendações. Este processo foi essencial para validar o desempenho do sistema e garantir que ele atendesse às expectativas dos utilizadores.

5.2 Conjunto de Dados

O conjunto de dados utilizado neste projeto foi obtido a partir da plataforma Foursquare 5.2.1, Os dados recolhidos da API apresentaram uma riqueza de detalhes sobre os Pontos de Interesse. Cada local incluído no conjunto de dados foi descrito de maneira abrangente, contemplando informações essenciais:

- **id_local**: identificador único para cada local.
- **nome**: nome do POI.
- **categoria**: classificação específica do POI (como restaurante, museu, parque, etc.).
- **descrição**: texto descritivo sobre o POI, disponível em vários idiomas.
- **rating_geral**: classificação agregada baseada nas avaliações dos utilizadores.

- **latitude** e **longitude**: coordenadas geográficas para localização.
- **verificado**: indicador booleano que identifica se o local foi oficialmente verificado na plataforma.

A API forneceu dados específicos associadas aos POIs, permitindo uma representação geral dos espaços, assim como informações sobre horários de funcionamento, contactos e links para sites oficiais.

Segundo (D. Yang et al., 2019) e (Di. Yang et al., 2020) , os dados foram recolhidos ao longo de um período de aproximadamente 22 meses (de abril de 2012 a janeiro de 2014), abrangendo 28.342 registos de POIs em Portugal. Este conjunto de dados revelou-se particularmente valioso para a análise e desenvolvimento do sistema de recomendação descrito neste projeto. Detalhes adicionais sobre os dados podem ser encontrados no subcapítulo a seguir 5.2.1.

Contudo, para assegurar a qualidade e relevância das recomendações, foi necessário realizar um processo rigoroso de limpeza, filtragem e enriquecimento, conforme descrito nas seções subsequentes.

5.2.1 Foursquare

Foursquare é um serviço web focado na tecnologia de geolocalização, une as funcionalidades do sistema de GPS com a interatividade das redes sociais. Na Foursquare pode informar a sua localização atual e ao mesmo tempo fazer comentários sobre locais visitados. (Amaral, 2017)

A rede social tem o objetivo claro de aproveitar a ascensão dos smartphones e da internet móvel ao redor do mundo.

Foursquare é muito popular em diversos locais do mundo. Na altura, (Amaral, 2017) descreve alguns dados:

- 10 bilhões de *check-ins* realizados no total;
- 600 milhões de fotos compartilhadas no total;
- 87 milhões de avaliações;
- 100 milhões de locais registados;
- 1,3 milhões de páginas de negócios;
- 55 milhões de utilizadores ativos em 2015;

Por meio do website Dingqi YANG[41] foi obtida informações da plataforma Foursquare, (D. Yang et al., 2019) e (Di. Yang et al., 2020), mais especificamente um conjunto abrangente de dados globais que inclui:

[41] https://sites.google.com/site/yangdingqi/home (último acesso em janeiro 2025).

- Conjunto de Dados de Check-ins Globais: Este conjunto de dados contém informações sobre os check-ins realizados pelos utilizadores da Foursquare em uma escala global durante o período especificado. Inclui um total de 22,809,624 check-ins, feitos por 114,324 utilizadores, em 3,820,891 locais diferentes.
- Dados das Redes Sociais dos Utilizadores: Além dos dados de check-ins, também estão incluídos dois instantâneos das redes sociais dos utilizadores, capturados antes e depois do período de recolha de dados dos check-ins. Esses dados das redes sociais contêm informações sobre as conexões de amizade entre os utilizadores. O conjunto de dados "antigo" contém 363,704 amizades, enquanto o conjunto de dados "novo" contém 607,333 amizades.
- Conjunto de Dados Brutos de Check-ins: Para atender a pedidos frequentes, também foi incluído na base de dados o conjunto de dados bruto de check-ins, que contém um volume ainda maior de informações. Este conjunto de dados inclui um total de 90,048,627 check-ins, feitos por 2,733,324 utilizadores, em 11,180,160 locais diferentes.

Estes conjuntos de dados fornecem uma visão abrangente das interações dos utilizadores com os locais na plataforma Foursquare, bem como das suas relações sociais associadas. São uma fonte valiosa de dados para análises e pesquisas em várias áreas, como mobilidade urbana, comportamento do utilizador e recomendação de locais.

5.2.2 Limpeza e Filtragem dos Dados

Realizou-se uma minuciosa limpeza e filtragem dos dados, restringindo-os exclusivamente aos locais de Portugal. Obtendo 28 342 locais mencionados no capítulo anterior, utilizando a API da Foursquare e efetuando solicitações HTTP, foram recolhidos os detalhes textuais dos locais, os quais foram armazenados na variável *foursquare_venues*[42], o script utilizado pode ser visualizado no Apêndice A . O processo de limpeza e filtragem incluiu:

- Remoção de categorias irrelevantes - Foi realizada a eliminação de categorias como *"Assisted Living", "Home (private)", "Housing Development", "Residential Building (Apartment / Condo)", "Trailer Park", "City", "County", "Country", "Neighborhood", "State", "Town", "Village", "States & Municipalities"* que não contribuem diretamente para o contexto das recomendações de POIs. Estas categorias estão mais relacionadas com informações geográficas e administrativas ou referem-se a locais sem um apelo direto para atividades recreativas ou comerciais. A Foursquare utiliza uma taxonomia hierárquica de categorias para classificar os seus POIs, organizando-os em categorias específicas que pertencem a uma categoria raiz mais abrangente tal como se

[42] https://api.foursquare.com/v2/venues/{}/tips? (último acesso em janeiro 2025).

explica no subcapítulo5.3.5. Por exemplo, um restaurante de sushi pertence à categoria específica "Restaurante de Sushi", que, por sua vez, está subordinada à categoria raiz *"Restaurantes"*. Esta estrutura facilita a organização e a navegação pelos diferentes tipos de locais, garantindo que as recomendações sejam mais precisas e relevantes.

Assim, foram mantidas apenas as categorias que, posteriormente, foram organizadas segundo a taxonomia da Foursquare, sendo distribuídas pelas suas respetivas categorias raiz, conforme destacado na Figura 6.

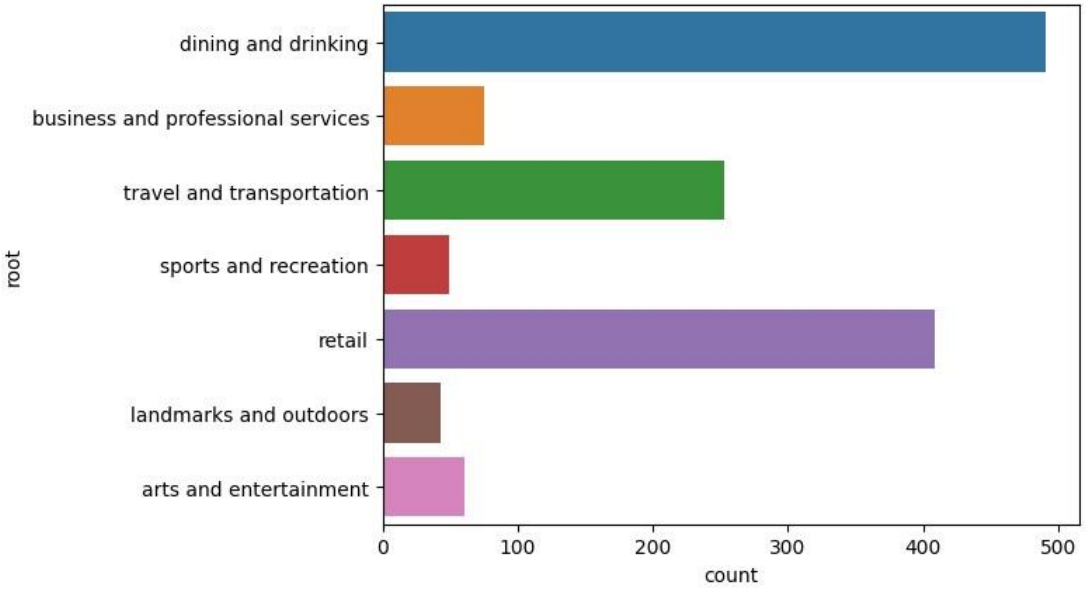


Figura 6 - Categorias raiz, dados distribuídos

- Exclusão de registos incompletos: Locais sem descrições ou *check-ins* associados foram descartados.
- Clusterização espacial: Utilização do algoritmo DBSCAN para identificar *clusters* de POIs com base na proximidade geográfica, substituindo as colunas de latitude e longitude pela variável "*Cluster*" explicada a seguir no subcapítulo 5.3.4 e como se pode visualizar no Apêndice H.

Como resultado, o conjunto de dados dos locais, contendo descrições e *check-ins*, após os procedimentos, ficou com 1379 registos. Este conjunto inclui variáveis essenciais do tipo *"id_local, nome, categoria, descrição, rating_geral, latitude, longitude e verificado",* sendo estes dados fundamentais para a implementação do sistema de recomendação descrito neste trabalho.

5.2.3 Divisão Linguística dos Dados

Para assegurar uma utilização eficiente do conjunto de dados e explorar as diferenças contextuais entre os idiomas, foi realizada uma divisão linguística baseada nas descrições textuais dos POIs. O conjunto de dados do *Foursquare* obtido para POIs em Portugal revelou que muitos locais, sobretudo em regiões turísticas, possuíam

nomes e descrições predominantemente em inglês, refletindo a forte presença de visitantes estrangeiros. Um exemplo disso é o restaurante português *Agapito*[43], localizado em Odeceixe, que no *Foursquare* apresenta a sua descrição em inglês: *'Traditional seafood restaurant with ocean views'*. Esta tendência verifica-se em muitos estabelecimentos turísticos, onde o idioma da descrição pode não corresponder ao idioma local.

A biblioteca SpaCy, com a extensão *langdetected* tal como (Carusotto et al., 2021), foi utilizada para identificar automaticamente o idioma predominante nas descrições textuais. Esta abordagem facilitou a separação dos dados em dois grupos principais: descrições em português e descrições em inglês. A utilização do SpaCy não apenas garantiu uma classificação confiável, mas também possibilitou a aplicação de modelos linguísticos apropriados para cada idioma.

Esta divisão foi essencial para testar os modelos em diferentes cenários linguísticos, aproveitando modelos treinados para português (como pt_core) e para inglês (como en_core). Além disso, a segmentação permitiu uma análise mais robusta, tanto individualmente para cada idioma quanto em um contexto multilíngue. Essa abordagem reflete a diversidade linguística dos dados e contribui para a avaliação da eficiência do sistema de recomendação em cenários reais.

5.3 Enriquecimento dos Dados

Neste capítulo apresenta-se uma visão geral sobre as abordagens utilizadas para enriquecer os dados, abrangendo desde a recolha de informações complementares, a aplicação de técnicas de processamento de linguagem natural até à integração de contexto geográfico. Essas técnicas visam otimizar os dados fornecidos ao sistema, assegurando que ele possa oferecer recomendações mais precisas, contextualizadas e personalizadas.

5.3.1 Conjunto de Dados *Ranking*

Outro conjunto de dados crucial é o "*Rankings*". O objetivo principal foi construir uma métrica personalizada que quantificasse o interesse de um utilizador em um determinado local com base no histórico de *check-ins*.

Este conjunto de dados foi criado para permitir que o sistema de recomendação incorporasse uma medida de preferência individual dos utilizadores, complementando as informações gerais dos locais já fornecidas pelo Foursquare. Foi necessário calcular o "*Ranking*" para:

1. Capturar Preferências Individuais: Criar uma métrica que reflete os locais mais visitados e, consequentemente, de maior relevância para cada utilizador.

[43] https://pt.foursquare.com/agapitoo7146989 (último acesso em janeiro 2025)

2. Melhorar a Personalização: Fornecer uma base de dados mais rica para o sistema, ajudando-o a recomendar POIs que realmente atendam aos interesses do utilizador.

Tal como o conjunto de dados "local_metadata", o "*Ranking*" também foi dividido em subconjuntos com base nos idiomas (português e inglês), resultando em:

- Ranking_geral: Inclui todas as interações entre utilizadores e POIs.
- Ranking_portugues: Subconjunto com dados de locais cujas descrições estão em português.
- Ranking_ingles: Subconjunto com dados de locais cujas descrições estão em inglês.

Assim, com base nas informações mencionadas anteriormente, este conjunto foi criado utilizando a seguinte equação (5.1):

$$Ranking = \frac{total\ check\text{-}ins\ do\ POI * utilizador}{total\ de\ check\text{-}ins * categoria * utilizador} \quad (5.1)$$

Onde:

- $total\ check\text{-}ins\ do\ POI * utilizador$: Refere-se ao número total de *check-ins* realizados em um POI específico, ponderado pelo utilizador que realizou os *check-ins*.
- $total\ de\ check\text{-}ins * categoria * utilizador$: Representa o total de *check-ins* em todas as categorias do conjunto de dados, ajustado pela categoria do POI e pelo utilizador.

Para calcular esta relação, foi necessário agregar os dados dos *check-ins*, criando um conjunto de dados onde cada linha representa um utilizador, uma categoria e o total de *check-ins* desse utilizador na respetiva categoria, foi realizada utilizando um *Group By*, como mostrado no Apêndice E.

Para exemplificar o cálculo, considera-se um utilizador que realizou:

- 10 *check-ins* num restaurante específico (POI).
- 50 *check-ins* no total de restaurantes da base de dados.

Aplicando equação (5.1):

$$Ranking = \frac{10}{50} = 0.2 \quad (5.2)$$

Isso significa que, para este utilizador, 20% de todos os seus check-ins em restaurantes foram realizados nesse POI específico. Esse valor indica um alto nível de envolvimento com o local, sugerindo que é um destino frequente ou preferido pelo utilizador. Assim, a criação deste conjunto de *rankings* foi fundamental para o

funcionamento do modelo de recomendação, pois permite criar uma matriz de interações utilizador-POI adequada para algoritmos de recomendação e capturar a preferência relativa do utilizador por locais dentro de cada categoria, tornando a recomendação mais personalizada.

5.3.2 Conjuntos de Dados Finais

Após os processos de recolha, limpeza, filtragem e enriquecimento descritos anteriormente, foram gerados seis conjuntos de dados principais, que formam a base para o desenvolvimento do sistema de recomendação. Estes conjuntos foram definidos para atender aos diferentes objetivos e cenários linguísticos do projeto, garantindo uma análise detalhada e flexível.

- **Local_metadata (geral):** Contém informações detalhadas sobre todos os locais incluídos no estudo (1.393 registos), com variáveis como identificador, nome, categoria, descrição, rating, coordenadas geográficas e status de verificação. Este conjunto é a base para a análise global.
- **Local_description_portugues**: Subconjunto com 1.045 locais do Local_metadata, contendo apenas descrições em português, permitindo análise específica para este idioma.
- **Local_description_ingles**: Similar ao conjunto anterior, mas com 298 locais cujas descrições estão em inglês, usado para avaliações direcionadas a este idioma.
- **Ranking_geral**: Inclui 11003 interações entre utilizadores e locais, capturando a avaliação geral de cada local pelos utilizadores.
- **Ranking_portugues**: Subconjunto do Rating_geral, com 5582 interações apenas relacionadas a locais com descrições em português. Focado na análise de utilizadores que interagem com este idioma.
- **Ranking_ingles**: Análogo ao Rating_portugues, com 923 interações de locais descritos em inglês, usado para testes multilíngues.

5.3.3 Word Embeddings

Neste subcapítulo apresenta-se o uso de técnicas de PLN para transformar dados textuais dos POIs em representações numéricas (vetores). Essas representações foram cruciais para capturar as propriedades semânticas e contextuais dos textos, permitindo uma análise mais precisa e recomendações personalizadas.

No projeto, foram utilizados dois modelos principais de embeddings para representar os textos associados aos POIs. O SpaCy com os modelos *en_core_web_md* e *pt_core_web_md*, que geraram vetores de 300 dimensões, foi utilizado para transformar colunas textuais ("Nome do Local", "Categoria", "Categoria Raiz" e "Descrição") Por exemplo, a descrição “Restaurante à beira-mar especializado em

frutos do mar" foi transformada em um vetor simplificado como [0.45, 0.78, 0.12, ...].

Para manter a coerência entre idiomas, o modelo *all-MiniLM-L6-v2* gerou embeddings multilíngues de 384 dimensões. Esta abordagem permitiu uma análise global, onde descrições em diferentes idiomas foram comparadas.

5.3.4 Contexto Geográfico

Após a Aquisição do conjunto de dados local_metadata, que contém diversas características de cada POI, foram destacadas duas colunas fundamentais relacionadas à localização: latitude e longitude. Estas colunas foram processadas utilizando o algoritmo DBSCAN para identificar *clusters* de POIs com base na proximidade geográfica.

O uso do DBSCAN permitiu agrupar POIs próximos em *clusters*, enquanto os pontos considerados como ruído (fora de qualquer cluster) foram isolados. Este procedimento resultou na criação de uma nova variável, denominada *Cluster*, que identifica os agrupamentos geográficos. Como resultado, as colunas de latitude e longitude foram eliminadas, sendo substituídas pela variável *Cluster*, simplificando assim a representação espacial no conjunto de dados.

A implementação detalhada deste procedimento, incluindo o código e os parâmetros ajustados, encontra-se documentada no Apêndice H. Além disso, os *clusters* geográficos ajudam a revelar padrões espaciais, como áreas de maior concentração de POIs ou locais frequentemente visitados em conjunto. A seguir mostra-se na Figura 7 com a tal concentração do treino com POIs com descrições em inglês (um dos exemplos):

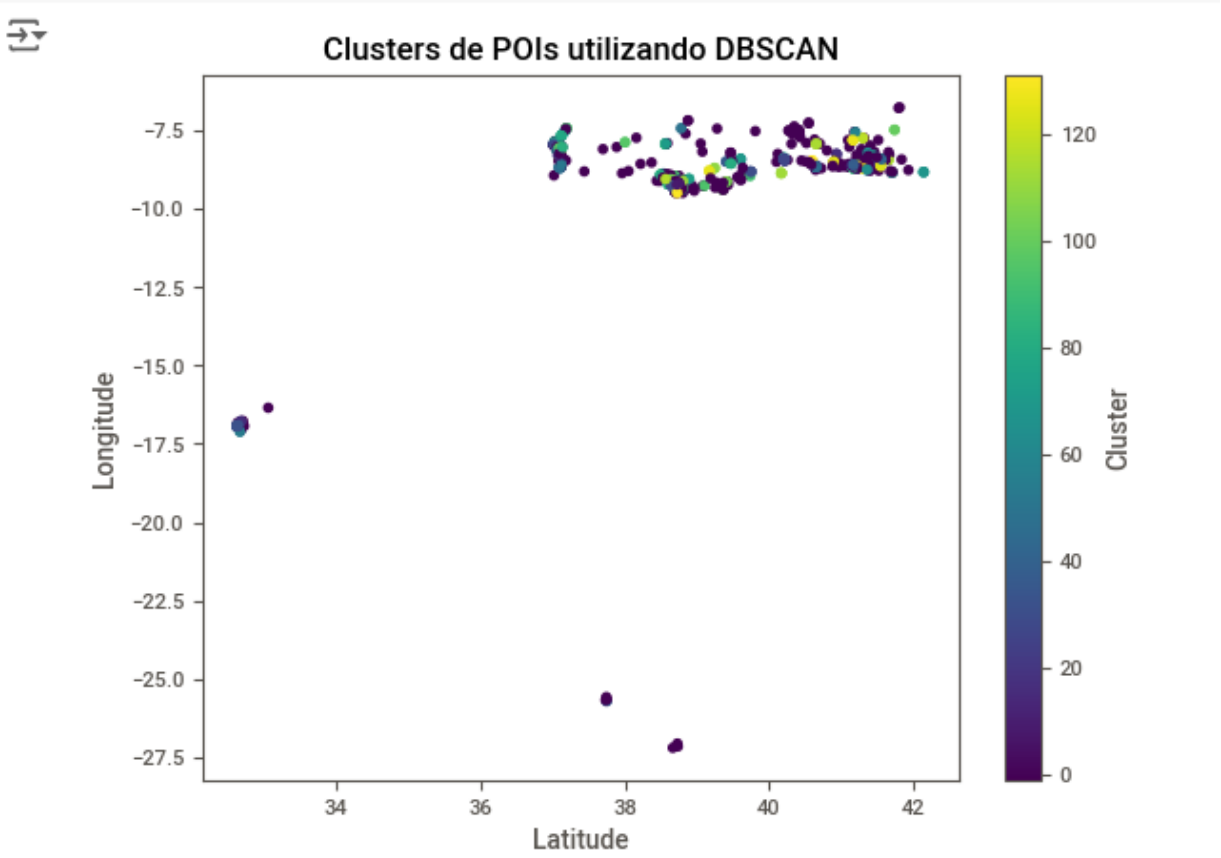


Figura 7 - *Clusters* POIs DBSCAN

5.3.5 Taxonomia de POIs

Relativamente às características do conjunto de dados dos POIs (local_metadata), as categorias desempenham um papel central no processo de recomendação. Durante a fase de recolha de dados, descrita nos subcapítulos anteriores, cada POI é associado a uma categoria específica, que descreve detalhadamente o tipo de serviço ou atividade oferecida. No entanto, para o treino do modelo de recomendação, foi necessário adotar uma abordagem mais generalizada, baseada na categoria raiz de cada POI.

Hierarquia de Categorias no Foursquare

O Foursquare utiliza uma taxonomia proprietária composta por mais de 1.000 categorias organizadas em diferentes níveis de granularidade. Essa estrutura hierárquica permite uma classificação precisa e flexível, com dois atributos principais para cada POI:

- IDs de Categoria: Identificador único correspondente à categoria atribuída ao POI. Cada ID identifica um nível específico dentro da hierarquia.
- Nome da Categoria**:** Um rótulo descritivo, como "Jantar" ou "Bebidas", que indica a categoria de forma legível para utilizadores.

No entanto, para este projeto, foi essencial mapear as categorias específicas para as suas categorias de topo, ou categorias raiz, utilizando a hierarquia fornecida pela documentação do Foursquare. Esse mapeamento permitiu representar cada POI de forma mais agregada e consistente, facilitando a análise de preferências dos utilizadores e a recomendação de POIs semelhantes.

Lista de Categorias Raiz

A seguir apresenta-se a lista das principais categorias raiz fornecidas pela taxonomia do Foursquare [44] e utilizadas neste projeto:

- Alimentação e Bebidas
- Lazer e Entretenimento
- Compras
- Desportes e Atividades ao Ar Livre
- Viagens e Transportes
- Arte e Cultura
- Serviços Locais e Profissionais
- Educação
- Saúde e Bem-Estar

A utilização de categorias raiz apresenta vantagens significativas para sistemas de recomendação. Primeiramente, reduz a complexidade do modelo, consolidando categorias altamente específicas em grupos mais gerais. Essa abordagem melhora a capacidade do modelo de identificar padrões de preferência entre os utilizadores, mesmo quando as suas interações estão distribuídas entre diferentes categorias específicas pertencentes a uma mesma raiz. Além disso, promove uma maior generalização das recomendações, aumentando a probabilidade de relevância para um maior número de utilizadores.

5.4 Método Híbrido

O sistema de recomendação desenvolvido adota uma abordagem híbrida, combinando as técnicas de recomendação baseada em conteúdo e filtragem colaborativa, com suporte de algoritmos de aprendizagem automática para treino e avaliação. Essa abordagem foi projetada para maximizar a personalização e a precisão das recomendações, integrando múltiplas fontes de informação (textual e geográfica).

[44] https://docs.foursquare.com/data-products/docs/categories (último acesso em janeiro 2025)

O sistema inicia com a extração e transformação de atributos textuais e contextuais dos POIs em representações vetoriais, utilizando embeddings para capturar similaridades semânticas. Em seguida, padrões de comportamento dos utilizadores são analisados por meio de algoritmos colaborativos, enquanto o modelo é ajustado continuamente com técnicas de aprendizagem automática. Esse fluxo é complementado por um processo rigoroso de treino e teste, assegurando a eficácia do sistema em contextos multilíngues e variados.

A implementação do método foi inspirada no tutorial "Recommendation System in Python: *LightFM*", por (Kapadia, 2020), que serviu como ponto de partida. No entanto, foram feitas adaptações significativas para integrar dados específicos do Foursquare, como descrições textuais, categorias raiz e localizações geográficas, permitindo que o sistema se ajustasse ao contexto do projeto.

A estrutura do método híbrido é composta pelas seguintes etapas:

5.4.1 Recomendação Baseada em Conteúdo

Na recomendação baseada em conteúdo, as características dos POIs foram utilizadas para identificar semelhanças com o perfil de preferências dos utilizadores, que foi construído com base no histórico de interações, incluindo *check-ins* e avaliações anteriores. As preferências foram inferidas a partir da frequência e do tipo de locais visitados, permitindo que o sistema identificasse padrões de interesse. Esses atributos incluem:

- Nome,
- Descrições textuais,
- Categorias específicas e raiz,
- Localização (latitude e longitude),
- Rating geral,
- Status de verificação.

Processo Implementado:

- Conversão de dados textuais em embeddings: Utilizou-*se all-MiniLM-L6-v2* para capturar relações semânticas entre descrições em múltiplos idiomas. Além disso, foi utilizado. SpaCy *(en_core_web_md e pt_core_news_md)* para representar textos em vetores de 300 dimensões.
- Captura de Padrões de Similaridade entre Utilizadores e Itens: são processados no *LightFM* para capturar padrões de similaridade entre utilizadores e itens. Em vez de calcular diretamente a similaridade do cosseno

com cosine_similarity do *sklearn*[45] tal como, (Carusotto et al., 2021), o sistema aprende a relação entre utilizadores e POIs a partir das interações registadas, otimizando a personalização das recomendações. Para aprimorar ainda mais a recomendação baseada em conteúdo.

- Integração das similaridades: O perfil do utilizador foi enriquecido com dados textuais e semânticos, garantindo que as recomendações fossem alinhadas aos interesses previamente demonstrados.

Por exemplo se um utilizador frequenta regularmente POIs classificados como "Restaurantes Italianos", o sistema prioriza recomendações de outros restaurantes similares, considerando descrições textuais e categorias raiz (Kapadia, 2020).

5.4.2 Recomendação por Filtragem Colaborativa

A filtragem colaborativa utiliza o histórico de interações dos utilizadores para identificar padrões de comportamento e preferências compartilhadas. No contexto deste projeto, a biblioteca *LightFM* foi utilizada para combinar feedback explícito (rankings) e implícito (*check-ins*).

Técnicas Utilizadas:

- Matriz de Interação - Criada a partir de *check-ins* e avaliações dos utilizadores nos POIs, representada em uma matriz esparsa (CSR matrix, *Compressed Sparse Row matrix* significado do seu acrónimo em inglês), garantindo eficiência na manipulação de grandes volumes de dados.
- Factorização de Matrizes - Implementada pelo *LightFM*, permitindo decompor padrões latentes das interações entre utilizadores e POIs, facilitando previsões mesmo para novos POIs com poucas interações.
- O modelo foi treinado utilizando o algoritmo BPR (Bayesian Personalized Ranking o seu acrónimo em inglês), que otimiza a ordenação dos itens recomendados, ajustando as preferências individuais dos utilizadores para melhorar a relevância da recomendação (Rendle et al., 2012).
- Redução da dispersão - Para reduzir a dispersão dos dados e melhorar a densidade da matriz de interações, foi aplicada uma técnica de discretização por intervalos (*binning*[46]) sobre as variáveis textuais ("nome, descrição, categoria, categoria raiz, etc") reduzindo assim a granularidade dos dados e diminuindo a presença de zeros. A discretização foi realizada através da

[45] Biblioteca Python para aprendizagem computacional com algoritmos e ferramentas de modelagem https://scikit-learn.org/stable/(último acesso em janeiro 2025)

[46] Agrupamento de valores contínuos num número menor de *Bins* https://docs.tibco.com/pub/spotfire/7.0.1/doc/html/bin/bin_what_is_binning.htm#:~:text=Binning%20is%20a%20way%20to,smaller%20number%20of%20age%20intervals (último acesso em janeiro 2025).

função pd.cut() como ilustra o Apêndice G. Além disso, para avaliar a estrutura dos dados antes da transformação, foi gerado um relatório exploratório com a biblioteca Sweetviz explicado no subcapítulo 4.5. Este relatório permitiu identificar padrões, visualizar a distribuição das variáveis

Os embeddings textuais dos POIs foram incorporados ao modelo *LightFM*, permitindo que a recomendação considerasse não apenas interações passadas, mas também características textuais dos POIs. Além disso, técnicas de redução de dispersão foram aplicadas para aumentar a eficiência do sistema.

O sistema demonstrou ser eficaz na mitigação do problema do "*cold start*", aproveitando descrições textuais e categorias raiz para fornecer recomendações. Por fim, a otimização computacional permitiu que as recomendações fossem geradas de maneira eficiente.

5.4.3 Resultados do Sistema

Os resultados obtidos pelo sistema de recomendação desenvolvido demonstram a eficácia do modelo híbrido baseado no *LightFM*, que combina filtragem colaborativa e baseada em conteúdo. A componente colaborativa do modelo aprende representações latentes para utilizadores e itens, ou seja, cria vetores numéricos que capturam as preferências implícitas de cada utilizador e as características dos itens. Esses vetores são ajustados durante o treino, permitindo que o modelo faça recomendações personalizada. A principal inovação do *LightFM* está no facto de ele utilizar uma abordagem dinâmica, onde as representações latentes são aprendidas e otimizadas iterativamente usando técnicas de gradiente descendente, em vez de decompor a matriz de interações de forma fixa. Isso torna o modelo mais flexível e eficiente na previsão das interações, além disso, o modelo permite incorporar características adicionais dos itens, como categorias, descrições e classificações, etc., o que melhora ainda mais a qualidade das recomendações.

A interpretação dos resultados é facilitada pela visualização das recomendações geradas para um utilizador específico, proporcionando uma melhor compreensão de como as interações e as características dos itens influenciam as sugestões feitas pelo sistema, como se ilustra na Figura 8 proporcionando uma melhor compreensão do funcionamento do sistema.

Figura 8 - Interpretação de resultados do sistema de recomendação

Ao selecionar um utilizador do conjunto de dados, o sistema gera uma lista estruturada com as seguintes informações:

- Identificador do Utilizador
- Histórico de Locais Visitados: Lista dos POIs, visitados pelo utilizador. Para cada local, são apresentados:
  - Nome do local.
  - Categoria específica do POI.
  - Categoria raiz do POI.
  - Recomendações Personalizadas: O sistema apresenta cinco novas sugestões de POIs, ordenadas com base na relevância estimada. Para cada recomendação, são exibidos:
  - Nome do local recomendado.
  - Categoria específica do POI.
  - Categoria raiz do POI.

Os resultados comprovam que o sistema é capaz de equilibrar sugestões populares com POIs menos conhecidos, promovendo diversidade nas recomendações enquanto atende às preferências individuais do utilizador.

Para validar a qualidade das recomendações geradas, foi realizada uma tabela com uma análise com 100 utilizadores, onde para cada um foram fornecidas 5 recomendações personalizadas. Dado que a tabela completa é extensa e apresenta um elevado volume de informação, a sua inclusão direta no corpo do documento não é prática. Assim, foi disponibilizada online e pode ser consultada[47]. Cada ponto de interesse (POI) recomendado foi acompanhado da sua categoria específica e categoria raiz, permitindo avaliar a proximidade entre os locais previamente visitados e os sugeridos pelo sistema.

Neste contexto, foram calculadas métricas de interseção entre visitas e recomendações (*Intersection between Visits and Recommendation*), analisando a percentagem de coincidência entre os locais recomendados e os locais visitados em três níveis distintos:

- *POI Level*: Correspondência direta entre um local visitado e um local recomendado.
- *Category Level*: Correspondência ao nível da categoria específica do local.

[47] Tabela dos 100 utilizadores com treinamento https://github.com/Karlajien02/Trabalho-de-Projeto/blob/1f383a494dd3f5b6dd8074f2b8eb68e0062222d1/Rec_100_Utilizadores_Com_Treinamentocsv.csv (último acesso em janeiro 2025) (último acesso março 2025).

- *Root Category Level*: Correspondência ao nível da categoria raiz, permitindo uma visão mais ampla das preferências do utilizador.

A avaliação revelou os seguintes valores médios (*Average*) e desvios padrão (*Std deviation*) para cada nível de correspondência:

Apêndice A *POI Level* (Ponto de Interesse Específico)

- Média: 8,73% das recomendações coincidiram exatamente com um ponto de interesse visitado.
- Desvio padrão: 14,23%, indicando uma elevada variação entre os utilizadores.

Apêndice B *Category Level* (Categoria Específica)

- Média: 19,08% das recomendações pertencem à mesma categoria dos locais visitados.
- Desvio padrão: 24,47%, sugerindo uma diferença significativa entre os casos analisados.

Apêndice C *Root Category Level* (Categoria Raiz)

- Média: 57,77% das recomendações pertencem à mesma categoria raiz dos locais visitados, indicando que o modelo capta preferências gerais dos utilizadores.
- Desvio padrão: 30,97%, com uma distribuição mais estável em relação às demais métricas.

Os resultados indicam que o sistema de recomendação apresenta um desempenho mais consistente ao nível da categoria raiz, o que sugere que o modelo consegue identificar padrões gerais de interesse dos utilizadores e recomendar pontos de interesse que pertencem a domínios semelhantes aos que já foram visitados. No entanto, verifica-se que a precisão diminui quando se considera a correspondência direta a locais específicos, evidenciando um menor grau de personalização ao nível granular.

5.4.4 Aprendizagem Automática

O modelo foi treinado utilizando um conjunto de dados dividido entre treino e teste, garantindo a validação da sua capacidade preditiva. Durante esse processo, foram ajustados Hiper parâmetros essenciais, como a taxa de aprendizagem, o número de componentes latentes e os coeficientes de regularização, permitindo um equilíbrio entre precisão e generalização. O modelo foi treinado aplicando a otimização Bayesiana por pares, como função de perda como explicado na seção 5.4.2, uma abordagem que otimiza a ordenação dos itens recomendados ao invés de prever valores absolutos de classificação. Essa técnica tem sido amplamente utilizada em

sistemas de recomendação, pois melhora a diferenciação entre itens relevantes e irrelevantes para cada utilizador (Shi et al., 2024).

Além disso, a incorporação de embeddings textuais como características adicionais desempenhou um papel fundamental no refinamento das recomendações. Para esse fim, foram utilizados modelos de PLN para converter as descrições dos POIs em representações vetoriais semânticas. A utilização de word embeddings multilíngues (como os modelos *pt_core*, *en_core* e *all-MiniLM-L6-v2*) garantiu que as recomendações mantivessem coerência entre diferentes idiomas, melhorando a precisão das sugestões personalizadas explicado no subcapítulo 5.3.3.

5.1.1 Avaliação do Modelo

A avaliação da performance do modelo foi conduzida utilizando métricas amplamente reconhecidas na literatura, garantindo que as previsões fossem rigorosamente testadas antes da implementação final. As métricas utilizadas foram:

- *Precision*@K e *Recall*@K: utilizadas para medir a qualidade das recomendações no Top-K, conforme aplicado em (Guo et al., 2017b), (Gao et al., 2015a) e (Chang et al., 2018b).
- AUC-ROC: utilizada para avaliar a capacidade do modelo em diferenciar itens relevantes e irrelevantes (Aliannejadi & Crestani, 2018).
- MRR (Mean Reciprocal Rank): que mede a posição do primeiro item relevante dentro das recomendações, garantindo um diagnóstico detalhado da qualidade das sugestões individuais (Chang et al., 2018b).

O cálculo dessas métricas foi implementado utilizando funções da biblioteca *LightFM.evaluation*, onde a precisão foi avaliada com a função *precision*_at_k, que mede a proporção de recomendações corretas entre as *K* principais sugestões feitas pelo sistema. O *Recall*, calculado através da função *recall*_at_k, mediu a capacidade do sistema de identificar corretamente todos os itens relevantes para os utilizadores. A AUC-ROC analisou a qualidade das previsões de relevância em diferentes pontos de corte, enquanto a métrica MRR foi utilizada para avaliar a posição do primeiro item relevante recomendado.

Estudos prévios, como os de (Guo et al., 2017b) (Xiong et al., 2020), demonstraram que essas métricas são eficazes na avaliação da precisão e relevância dos sistemas de recomendação, tornando-se essenciais para a validação dos resultados obtidos no presente trabalho.

Para aprimorar ainda mais a performance do modelo, foram realizados ajustes iterativos no híper parâmetros. A taxa de aprendizagem foi ajustada progressivamente para evitar convergência prematura ou *overfitting*[48].

Os resultados dessas avaliações são detalhados no capítulo 6RESULTADOS E DISCUSSÕES, onde são analisadas as performances comparativas das diferentes configurações testadas.

6 RESULTADOS E DISCUSSÕES

Neste capítulo, apresenta-se uma análise detalhada do desempenho do sistema de recomendação implementado, com base em várias experiências e testes realizados. O objetivo principal é avaliar a eficácia do modelo em diferentes cenários e com diferentes características dos dados.

A avaliação do sistema envolveu várias etapas, incluindo análises descritivas do conjunto de dados, experiências com diferentes configurações de modelo, e a comparação de diversas métricas de desempenho. Para além disso, foram conduzidos testes específicos relacionados com a diversidade de linguagens presentes nos dados, uma vez que o conjunto incluía informações em múltiplos idiomas.

6.1 Estrutura da Avaliação

A análise experimental foi organizada em diferentes etapas para garantir uma compreensão abrangente do desempenho do sistema:

- Análises Descritivas - Esta etapa foi fundamental para explorar a distribuição das interações entre utilizadores e POIs, bem como a frequência e diversidade das categorias e outros atributos relevantes. Estatísticas descritivas permitiram identificar padrões e características importantes do conjunto de dados, como a concentração de interações em determinadas categorias ou regiões.
- Experiências com Modelos - O modelo *LightFM* foi submetido a diversos testes com diferentes configurações, incluindo a análise do impacto da segmentação linguística. Os dados foram agrupados por idioma, permitindo avaliar como a linguagem influencia o desempenho do sistema. Além disso, foram explorados os intervalos de *bins* criados para variáveis contínuas como se pode apreciar no subcapítulo 5.4.2. Um facto importante é que as *bins* são modificadas consoante ao idioma.

[48] *Overfitting* ocorre quando um modelo se ajusta demasiado aos dados de treino e não generaliza bem. https://www.ibm.com/br-pt/think/topics/overfitting (último acesso março 2025).

- Métricas de Avaliação: A eficácia foi avaliada usando métricas como MRR, Precisão,@K *Recall*@K, e AUC.

6.2 Avaliação de Desempenho com Dados Multilíngues

O conjunto de dados utilizado neste estudo contém descrições e categorias em diferentes idiomas, predominantemente português e inglês. Como a linguagem pode influenciar diretamente a qualidade das recomendações, foram conduzidas diversas experiências para avaliar o impacto da variação linguística no desempenho do sistema de recomendação.

Inicialmente, os dados foram segmentados por idioma, criando dois subconjuntos distintos: um contendo apenas POIs com descrições em português e outro com descrições em inglês. Essa separação permitiu analisar diferenças de desempenho do modelo ao operar em contextos linguísticos específicos.

Para uma avaliação mais abrangente, foram realizados testes cruzados que incluíram diferentes cenários. Primeiramente, o modelo foi treinado e testado em dados do mesmo idioma, ou seja, treino e teste em português e, separadamente, treino e teste em inglês. Em seguida, foram realizadas experiências onde o modelo foi treinado em um idioma e testado no outro, permitindo examinar a sua robustez e capacidade de generalização quando exposto a variações linguísticas.

Além disso, foi testada a seleção de diferentes características textuais para entender seu impacto na recomendação. Foram feitas experiências utilizando todas as variáveis textuais disponíveis (como nome, categoria especifica, categoria raiz e descrição) e, posteriormente, combinando algumas delas, como nome e categoria ou nome e descrição. Por fim, cada característica textual foi testada individualmente, permitindo identificar quais delas contribuíam mais significativamente para a precisão e relevância das recomendações.

Outro aspeto analisado foi o impacto da informação multilíngue na precisão e diversidade das recomendações. Especificamente, avaliou-se se a inclusão de descrições em diferentes idiomas contribuía para recomendações mais variadas e abrangentes ou, pelo contrário, introduzia ruído nas previsões do modelo. Foram observadas variações na precisão das recomendações, bem como a capacidade do modelo em capturar relações semânticas entre POIs de diferentes idiomas. Além disso, analisou-se se a segmentação linguística favorecia sugestões mais relevantes para cada perfil de utilizador ou se, ao misturar os idiomas, o modelo conseguia aprender padrões mais gerais.

Os resultados obtidos com essas avaliações são analisados no subcapítulo 6.3, onde são detalhados os impactos da segmentação linguística no desempenho do sistema de recomendação.

6.3 Resultados

Relativamente aos testes feitos no sistema de recomendação, foi realizada a escolha das características a serem avaliadas no sistema, incluindo variáveis textuais e numéricas. Este processo envolveu a identificação de atributos, como categorias, descrições, localização, avaliações gerais e verificações dos POIs, que poderiam influenciar diretamente a qualidade das recomendações geradas.

No processamento dos dados, destaca-se a existência de uma parte do código dedicada à seleção e transformação dessas variáveis, utilizando técnicas como o mapeamento de indicadores que é descrito no Apêndice I.

Para uma melhor interpretação e comparação do desempenho, foram elaboradas tabelas que detalham o impacto de diferentes configurações do modelo e o comportamento das variáveis no sistema de recomendação.

Os testes envolveram três modelos de PLN: multilingue, inglês e português. Os resultados demonstraram diferenças significativas no desempenho entre os modelos. Durante a avaliação, foi adotado o valor K=5 nas métricas *Precision*@K e *Recall*@K, significando que o desempenho do sistema foi medido considerando os cinco principais POIs recomendados para cada utilizador, tal como (Gao et al., 2015a), (Guo et al., 2017b), (Aliannejadi & Crestani, 2018), (Chang et al., 2018b).

A Figura 9 apresenta os resultados do modelo Multilingual onde foram utilizadas 11003 interações de um total de 1393 locais. Este modelo apresentou resultados equilibrados em termos de desempenho. *Precision*@k foi de 0.15 no treino e 0.13 no teste, indicando que a proporção de recomendações relevantes, especialmente no conjunto de teste, foi moderada. O *Recall*@k alcançou valores de 0.67 no treino e 0.61 no teste, demonstrando que uma quantidade razoável de itens relevantes foi recuperada, embora não de forma abrangente. O AUC Score, com 0.99 no treino e 0.98 no teste, mostrou excelente capacidade de separação entre itens relevantes e irrelevantes. No entanto, o MRR, com valores de 0.36 (treino) e 0.32 (teste), revelou que, embora o modelo possa identificar itens relevantes, eles frequentemente não estão entre as primeiras posições na lista de recomendações.

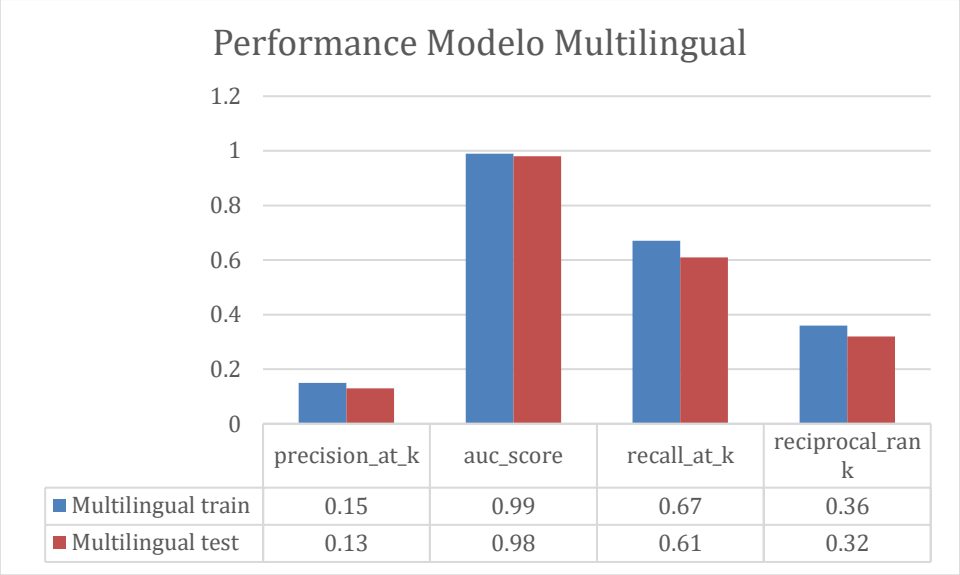


Figura 9 - Analise de métricas de avaliação Modelo Multilingual

Este desempenho sugere que o modelo multilingue é adequado para cenários gerais, mas pode não ser o mais eficiente para prioridades de posicionamento de relevância.

A Figura 10 apresenta os resultados do modelo em inglês onde foram utilizadas 923 interações de um total de 298 locais. Este foi o modelo com melhor desempenho entre os três avaliados. O *Recall*@k foi de 1.00 no treino e 0.99 no teste, demonstrando que quase todos os itens relevantes foram identificados e recomendados. O MRR, com valores de 0.92 no treino e 0.90 no teste, revelou que os itens recomendados eram frequentemente posicionados no topo das listas, indicando alta precisão na ordenação. O AUC Score, consistentemente alto em 0.99 tanto no treino quanto no teste, reforçou a robustez do modelo em distinguir itens relevantes de irrelevantes.

Figura 10 - Analise de métricas de avaliação Modelo Inglês

Apesar disso, *Precision*@k, com 0.21 no treino e 0.20 no teste, sugeriu que a proporção de itens verdadeiramente relevantes entre as recomendações ainda pode ser aprimorada. Esses resultados indicam que o modelo em inglês é eficaz e confiável, especialmente para priorizar recomendações relevantes.

A Figura 11 com o modelo em português onde foram utilizadas 5582 interações de um total de 1045 locais, apresentou o desempenho mais baixo entre os avaliados, destacando desafios significativos no processamento dos dados nessa língua. *Precision*@k e *Recall*@k próximos de 0 (0.00 no treino e no teste para Precisão; 0.01 no treino e no teste para *Recall*) indicam grande dificuldade do modelo em identificar e recomendar itens relevantes. O AUC Score, de 0.60 no treino e 0.59 no teste, revelou uma capacidade limitada de distinguir entre itens relevantes e irrelevantes. Além disso, o MRR, com valores de 0.02 no treino e no teste, mostrou que os itens relevantes raramente eram posicionados entre as primeiras posições na lista de recomendações.

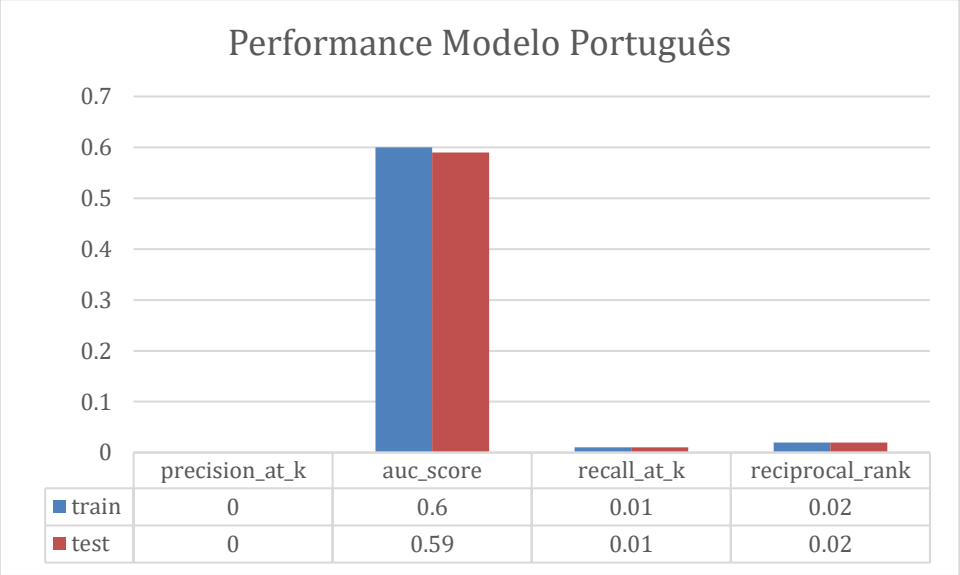


Figura 11 - Analise de métricas de avaliação Modelo Português

Esse desempenho pode ser atribuído que o modelo enfrentou dificuldades na extração e interpretação das características textuais em português.

Apesar de o modelo em português conter um número maior de dados do que o modelo em inglês, isso não significa necessariamente um melhor desempenho. O volume de dados, por si só, não garante a eficácia do modelo. Se os textos em português apresentavam muito ruído, variações linguísticas, abreviaturas, erros ortográficos ou falta de normalização, o modelo pode ter tido dificuldades em extrair padrões semânticos úteis. Além disso, a presença de dados com menor riqueza semântica (descrições curtas ou pouco informativas) pode ter impactado negativamente a capacidade do modelo de aprender representações significativas.

Adicionalmente, o português é uma língua morfologicamente mais complexa do que o inglês, apresentando maior variação gramatical, flexão de palavras e regras

sintáticas mais elaboradas, o que pode ter tornado o processamento e a modelação mais desafiantes.

Os resultados refletem a influência direta da qualidade dos dados textuais e das características linguísticas na eficácia dos modelos. O modelo em inglês destacou-se como o mais robusto, enquanto o modelo multilingue apresentou resultados equilibrados e aplicáveis em contextos mais diversificados. Por outro lado, o desempenho do modelo em português evidencia a importância de melhorias no tratamento dos dados e ajustes nos parâmetros do modelo para melhor atender às especificidades do idioma.

7 CONCLUSÕES E PERSPETIVAS FUTURAS

Neste capítulo, são apresentadas as conclusões do estudo realizado. Primeiramente, é feita uma revisão geral do tema e do contexto da pesquisa. Segue-se a revisão dos objetivos e do trabalho desenvolvido. Posteriormente, são discutidos os principais contributos do estudo e suas limitações. Finalmente, são apresentadas sugestões para trabalhos futuros.

7.1 Revisão Geral do Tema

O estudo dos sistemas de recomendação, especialmente com dados textuais, tem vindo a ganhar relevância na análise de grandes volumes de informação sobre locais e pontos de interesse. A capacidade de personalizar recomendações com base em dados sobre o comportamento e preferências dos utilizadores, aliada a um conhecimento geográfico e ao uso de informações textuais, pode promover uma experiência mais satisfatória e eficiente tanto para os utilizadores como para as organizações (como exemplo algumas plataformas no âmbito do turismo). A utilização do algoritmo *LightFM* teve um comportamento eficaz na análise e na recomendação de locais.

Contudo, a análise de dados textuais, como descrições de locais e categorias, tem um papel igualmente crucial na melhoria da qualidade das recomendações. A inclusão dessas características permitiu ao sistema capturar relações semânticas entre os POIs e as preferências dos utilizadores, resultando em recomendações mais precisas e personalizadas. Especificamente, a categoria do local mostrou-se essencial para agrupar POIs com características semelhantes, ajudando o modelo a identificar padrões de interesse dos utilizadores. Já a descrição do local foi fundamental para refinar ainda mais a recomendação, permitindo que o modelo diferenciasse locais dentro da mesma categoria com base em atributos únicos mencionados nos textos.

No entanto, também foi observado que a inclusão excessiva de texto sem um processamento adequado poderia adicionar ruído ao sistema. Para mitigar esse efeito, foram utilizadas técnicas de PLN para converter descrições e categorias em representações vetoriais, garantindo que o modelo pudesse extrair padrões relevantes e reduzir informações redundantes.

Com a crescente disponibilidade de fontes de dados gratuitas, é possível aprimorar ainda mais os sistemas de recomendação e personalização, permitindo decisões mais informadas e eficazes, tanto para as empresas como para o público em geral.

7.2 Revisão dos Objetivos e do Trabalho Realizado

O principal objetivo deste estudo foi o desenvolvimento de um sistema de recomendação híbrido para pontos de interesse em Portugal, utilizando dados do

Foursquare e aplicando técnicas como o algoritmo *LightFM* para gerar recomendações personalizadas. Além disso, foi explorada a importância da inclusão de dados textuais assim como também se incluíram dados geográficos, utilizando o algoritmo DBSCAN para agrupar coordenadas de latitude e longitude, o que permitiu identificar *clusters* e padrões espaciais relevantes para a recomendação de locais.

O trabalho realizado iniciou-se com a investigação das fontes de dados relevantes, em particular os POIs disponíveis na plataforma Foursquare. Este levantamento inicial visou compreender os tipos de dados disponíveis para a construção do sistema de recomendação. Entre os dados selecionados, destacaram-se as descrições, categorias e as coordenadas geográficas dos POIs, assim como também as interações que os utilizadores tiveram nesses locais (*Check-ins*). Foi dada especial atenção ao potencial desses dados para melhorar a personalização das recomendações, especialmente no que se refere ao uso de informações textuais sobre os locais.

Com base na coleta de dados, o passo seguinte envolveu o desenvolvimento do sistema de recomendação híbrido, integrando a biblioteca *LightFM* para a filtragem colaborativa e por conteúdo. A implementação seguiu o modelo proposto por (Kapadia, 2020), adaptando-o para incluir dados textuais, como nome, categoria e descrição, assim como também dados de localização dos POIs. Paralelamente, o algoritmo DBSCAN foi aplicado para agrupar os POIs com base em suas coordenadas geográficas, identificando padrões espaciais que pudessem contribuir para as recomendações mais contextuais.

Após a implementação, o sistema foi avaliado utilizando métricas como precisão, *recall*, AUC e MRR, a fim de medir a eficácia das recomendações geradas. A análise dos resultados mostrou que a combinação de dados textuais e geográficos resultou em recomendações mais precisas e relevantes para os utilizadores, destacando a importância de integrar diferentes fontes de informação na construção de sistemas de recomendação.

Além disso, o estudo investigou a viabilidade de aplicar o sistema a um conjunto específico de dados em Portugal, como uma prova de conceito. Embora o sistema tenha mostrado um desempenho promissor, as limitações dos dados disponíveis e a necessidade de adaptar o modelo a outros contextos geográficos foram reconhecidas. Por fim, foram discutidas possíveis melhorias, incluindo a expansão para outros países lusófonos, bem como a implementação de técnicas adicionais para refinar ainda mais as recomendações.

7.3 Contributos

Considerando as diversas fases que compõem o desenvolvimento do estudo, é possível identificar diferentes contributos teóricos e práticos que resultaram do trabalho realizado, os quais são descritos a seguir.

7.3.1 Contributos Teóricos

Este estudo contribui teoricamente ao apresentar uma abordagem híbrida para a recomendação de locais, integrando dados geográficos e textuais com o uso de algoritmos de aprendizagem automática, como o *LightFM*. Um dos principais contributos teóricos deste estudo foi a integração de diferentes fontes de dados, como informações textuais, que foi o foco de pesquisa na investigação de estudos relacionados.

De acordo com a pesquisa realizada por (Werneck et al., 2020) até 2020, foram poucos os sistemas que aproveitaram plenamente os dados textuais, o que destaca uma lacuna importante no uso desse tipo de informação em sistemas de recomendação.

Na fase do processo de recomendação, a investigação neste sentido contribui para uma compreensão mais profunda das etapas de filtragem e personalização de recomendações, sendo aplicável a uma vasta gama de sistemas de recomendação e outras áreas que dependem da análise de grandes volumes de dados.

Estes contributos proporcionam uma base sólida para futuras investigações, ampliando o entendimento sobre como as diferentes fontes de dados podem ser combinadas para melhorar a personalização e a precisão das recomendações.

7.3.2 Contributos Práticos

No âmbito prático, durante a execução deste estudo, diversas estratégias e abordagens práticas foram adotadas para alcançar os objetivos propostos. A implementação e o desenvolvimento de várias soluções tecnológicas foram fundamentais para o sucesso da pesquisa, e muitos dos avanços obtidos dependem diretamente da utilização dessas ferramentas. Entre as ações realizadas, destaca-se a utilização da API da Foursquare que facilita a aquisição do conjunto de dados dos locais, e que também por meio de (D. Yang et al., 2019) e (Di. Yang et al., 2020) que forneceram as iterações dos utilizadores juntamente com os locais que automatizaram processos e facilitaram a análise e manipulação dos dados.

Após a coleta, uma segunda filtragem foi criada para realizar a extração de dados específicos, com base em critérios definidos, como tipo de POI, localização, e categorias relevantes para o sistema de recomendação.

Além disso, foi implementada uma terceira manipulação para a análise dos *clusters* geográficos, utilizando o algoritmo DBSCAN. Essa ferramenta foi essencial para agrupar os dados de coordenadas geográficas e fornecer *insights* sobre a distribuição espacial dos pontos de interesse, elemento-chave para o aprimoramento das recomendações.

7.4 Limitações

Durante o desenvolvimento deste estudo, diversas limitações foram identificadas, as quais afetaram tanto o processo de recolha de dados quanto a análise realizada. A seguir, são apresentadas as principais limitações que surgiram ao longo da pesquisa.

7.4.1 Desafios com a Diversidade de Idiomas

Inicialmente, ao focar apenas em locais em Portugal, supôs-se que o idioma dos dados seria predominantemente o português, uma vez que a maioria dos pontos de interesse está localizada neste país. No entanto, ao utilizar um modelo PLN pré-treinado, foi constatado que os resultados eram insatisfatórios, com saídas "estranhas" para certos dados textuais. A partir dessa observação, foi realizada uma análise mais detalhada para entender a diversidade linguística dos dados textuais. Percebeu-se que, embora o português fosse o idioma predominante, havia uma quantidade significativa de textos em inglês, bem como em outros idiomas. Essa diversidade linguística é influenciada, em parte, pelo turismo, pela presença de empresas multinacionais e pela imigração em Portugal.

Como resultado, foi necessário testar modelos PLN adaptados a essa diversidade linguística, o que contribuiu para uma melhor compreensão da importância da linguagem nos modelos de recomendação. Esta variação pode ser considerada uma limitação, pois exigiu esforços adicionais para garantir que o sistema fosse capaz de lidar adequadamente com os diferentes idiomas presentes nos dados.

7.4.2 Limitações do Foursquare e Questões de Acesso aos Dados

Outra limitação relevante foi o processo de aquisição de dados através da plataforma Foursquare. Para aceder à API da Foursquare e obter dados sobre os pontos de interesse, foi necessário criar uma conta com informações pessoais, incluindo dados bancários. Além disso, a Foursquare impõe restrições no número de solicitações que podem ser feitas por hora, o que afetou a eficiência do processo de recolha de dados. A plataforma permite até 500 solicitações autenticadas por hora por *token OAuth*[49], e, com múltiplos utilizadores conectados, o número total de requisições pode ser multiplicado. No entanto, este limite de taxa, combinado com as necessidades de criação de perfis e autenticação, tornou o processo de recolha mais lento e complexo.

7.4.3 Questões de Privacidade e Uso de Dados de Redes Sociais

Uma limitação adicional foi a não utilização de dados de redes sociais, como as redes CBSNs, que poderiam fornecer informações mais detalhadas, como opiniões dos

[49] Usada para autorizar e autenticar utilizadores em APIs sem expor credenciais. https://docs.foursquare.com/developer/reference/personalization-apis-authentication (último acesso março 2025).

utilizadores sobre locais ou pedidos de recomendação. Muitos estudos como (Xiong et al., 2020), (Chang et al., 2018) e (Carusotto et al., 2021) utilizaram dados CBSNs obtendo melhores resultados, especialmente em termos de análise de sentimentos e comportamento do utilizador. Contudo, devido a preocupações com a privacidade e à crescente regulamentação sobre dados pessoais, o uso dessas fontes foi descartado.

Por exemplo, no caso de (Xiong et al., 2020), utilizaram dados heterogéneos, combinando informações do Foursquare com dados do Twitter e Facebook. No entanto, ao fazer isso, essas pesquisas enfrentaram questões de privacidade, utilizando métodos para proteger dados pessoais, como a anonimização de locais e utilizadores e a alteração das coordenadas geográficas. O uso de dados privados, de redes sociais, é altamente restrito e, portanto, optou-se por não incorporar esse tipo de informação, o que impactou a possibilidade de obter uma maior diversidade de dados textuais.

7.4.4 Limitação Geográfica

Outro desafio foi a escolha de Portugal como único país para a recolha e análise dos dados. Uma vez que Portugal é um país diminuto, a filtragem dos dados, associada à limitação geográfica, resultou num conjunto de dados relativamente pequeno. Isto limitou a capacidade de realizar comparações amplas ou de extrapolar resultados para outros contextos. Além disso, ao focar apenas num país, foi difícil analisar variações culturais ou comportamentais entre diferentes regiões, o que pode ser um fator importante para melhorar a precisão das recomendações.

Dessa forma, uma possível expansão do estudo para outros países lusófonos poderia proporcionar dados mais diversos e permitir comparações interessantes entre diferentes contextos culturais. A inclusão de dados de outros países da comunidade lusófona enriqueceria o estudo e poderia servir como um complemento importante para o trabalho aqui apresentado.

7.5 Trabalho Futuro

O estudo desenvolvido nesta investigação trouxe contribuições importantes para a compreensão e aprimoramento dos sistemas de recomendação de POIs, mas várias áreas ainda podem ser exploradas para melhorar e expandir os resultados obtidos. O futuro deste trabalho passa por integrar novas fontes de dados, expandir a análise geograficamente e explorar ferramentas e tecnologias emergentes, como inteligência artificial generativa (e.g. *Chatgpt*) e novas plataformas de dados.

Primeiramente, uma das direções mais promissoras seria a expansão geográfica da análise, considerando não apenas Portugal, mas também outros países lusófonos, como o Brasil, Angola, Moçambique e outros. Cada um desses países apresenta características culturais, turísticas e de consumo distintas, o que pode influenciar os

padrões de recomendação de maneira única. A inclusão de dados de diferentes contextos geográficos pode enriquecer significativamente o conjunto de dados e permitir a criação de um sistema de recomendação mais robusto e adaptável, capaz de operar em diferentes cenários e atender a uma maior diversidade de utilizadores.

Além disso, o uso de fontes de dados mais versáteis e atualizadas é uma área que merece atenção no futuro. Com o crescente uso de plataformas como Google Maps[50], TripAdvisor[51], entre outras, é possível acessar informações dinâmicas e constantemente atualizadas sobre os POIs. A utilização de dados provenientes dessas plataformas, que incluem avaliações de utilizadores, horários de funcionamento, imagens e recomendações personalizadas, poderia oferecer recomendações ainda mais precisas e relevantes.

Outro ponto importante para o trabalho futuro seria a integração com modelos generativos de linguagem, que pode enriquecer as recomendações de forma substancial. Em vez de apenas fornecer uma lista de locais recomendados, esses modelos poderiam explicar as razões por trás das recomendações, permitindo ao utilizador uma melhor compreensão do processo e aumentando a confiança no sistema. Além disso, a utilização de Modelos de Aprendizagem Multimodal (MML - Multimodal Machine Learning o seu acrónimo em inglês) poderia ampliar ainda mais o impacto das recomendações, combinando dados textuais, imagens e interações dos utilizadores para oferecer descrições mais detalhadas dos POIs. Isso permitiria não apenas a apresentação de informações factuais, mas também a inclusão de contextos culturais, históricos e até mesmo sugestões personalizadas, adaptadas ao perfil e interesses do utilizador. Essa capacidade de contextualizar as recomendações aumentaria a relevância e a satisfação do utilizador com o sistema. Em vez de apresentar uma lista estática de locais recomendados, seria interessante desenvolver um sistema mais dinâmico, que permitisse aos utilizadores ajustar as suas preferências em tempo real, com base em diferentes critérios como proximidade, tipo de atividade ou avaliação dos utilizadores. A adição de dados contextuais, como a hora do dia, as condições climáticas ou a época do ano, também poderia ajudar a gerar recomendações mais personalizadas, alinhadas com as necessidades e expectativas dos utilizadores. A aplicação de técnicas mais avançadas de aprendizagem computacional e análise preditiva seria fundamental para refinar ainda mais as recomendações, levando em consideração fatores como o histórico de comportamento do utilizador e variáveis externas.

Essas propostas visam enriquecer a experiência do utilizador e melhorar a precisão das recomendações. Com o contínuo avanço das tecnologias de dados, modelos generativos de linguagem e algoritmos de aprendizagem computacional, os sistemas

---

[50] Plataforma de mapas e navegação que fornece direções, informações sobre locais. https://www.google.com/maps (último acesso março 2025)

[51] Site de avaliações e recomendações de hotéis, restaurantes e atrações turísticas com base em experiências de utilizadores. https://www.tripadvisor.com (último acesso março 2025)

de recomendação de POIs tornam-se cada vez mais sofisticados, permitindo a personalização dinâmica e a adaptação contínua às preferências dos utilizadores. O uso de aprendizagem computacional já é uma parte fundamental desses sistemas, mas o futuro apresenta um grande potencial para aprimorar ainda mais a eficácia das recomendações, integrando abordagens híbridas, modelos multimodais e inteligência artificial mais avançada.

REFERÊNCIAS BIBLIOGRÁFICAS

Adomavicius, G., & Tuzhilin, A. (2005, June). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, *17*(6), 734–749. https://doi.org/10.1109/TKDE.2005.99

Aggarwal, C. C. (2016). *Recommender Systems: The Textbook* (1st ed.). Springer Publishing Company, Incorporated.

Aliannejadi, M., & Crestani, F. (2018). Personalized Context-Aware Point of Interest Recommendation. *ACM Transactions on Information Systems (TOIS)*, *36*(4). https://doi.org/10.1145/3231933

Amaral, L. (2017, July 9). *Foursquare: entenda toda a história dessa rede social*. https://rockcontent.com/br/blog/foursquare/

Bao, J., Zheng, Y., & Mokbel, M. F. (2012). Location-based and preference-aware recommendation using sparse geo-social networking data. *Undefined*, 199–208. https://doi.org/10.1145/2424321.2424348

Biega, A. J., Diaz, F., Ekstrand, M. D., & Kohlmeier, S. (2020). *Overview of the TREC 2019 Fair Ranking Track*. https://doi.org/10.48550/arxiv.2003.11650

Carusotto, V. E., Pilato, G., Persia, F., & Ge, M. (2021). User Profiling for Tourist Trip Recommendations using Social Sensing. *Proceedings - 23rd IEEE International Symposium on Multimedia, ISM 2021*, 182–185. https://doi.org/10.1109/ISM52913.2021.00036

Chang, B., Park, Y., Park, D., Kim, S., & Kang, J. (2018a). Content-aware hierarchical point-of-interest embedding model for successive POI recommendation. *IJCAI International Joint Conference on Artificial Intelligence*, *2018-July*, 3301–3307. https://doi.org/10.24963/IJCAI.2018/458

Chang, B., Park, Y., Park, D., Kim, S., & Kang, J. (2018b). Content-aware hierarchical point-of-interest embedding model for successive POI recommendation. *IJCAI International Joint Conference on Artificial Intelligence*, *2018-July*, 3301–3307. https://doi.org/10.24963/IJCAI.2018/458

Gao, H., Tang, J., Hu, X., & Liu, H. (2015a). Content-Aware Point of Interest Recommendation on Location-Based Social Networks. *Proceedings of the National Conference on Artificial Intelligence*, 1721–1727. https://doi.org/https://dl.acm.org/doi/10.5555/2886521.2886559

Gao, H., Tang, J., Hu, X., & Liu, H. (2015b). *Content-Aware Point of Interest Recommendation on Location-Based Social Networks*. 6.

Guo, Q., Sun, Z., Zhang, J., Chen, Q., & Theng, Y. L. (2017a). Aspect-aware point-of-interest recommendation with geo-social influence. *UMAP*

2017 - Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization, 17–22. https://doi.org/10.1145/3099023.3099066

Guo, Q., Sun, Z., Zhang, J., Chen, Q., & Theng, Y. L. (2017b). Aspect-aware point-of-interest recommendation with geo-social influence. *UMAP 2017 - Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*, 17–22. https://doi.org/10.1145/3099023.3099066

Honnibal, M., Montani, I., Van Landeghem, S., & Boyd, A. (2020). *spaCy: Industrial-strength Natural Language Processing in Python*. https://doi.org/10.5281/zenodo.1212303

Hu, B., & Martin, E. (2013a). Spatial topic modeling in online social media for location recommendation. *RecSys 2013 - Proceedings of the 7th ACM Conference on Recommender Systems*, 25–32. https://doi.org/10.1145/2507157.2507174

Hu, B., & Martin, E. (2013b). Spatial topic modeling in online social media for location recommendation. *RecSys 2013 - Proceedings of the 7th ACM Conference on Recommender Systems*, 25–32. https://doi.org/10.1145/2507157.2507174

Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. In *Egyptian Informatics Journal* (Vol. 16, Issue 3, pp. 261–273). Elsevier B.V. https://doi.org/10.1016/j.eij.2015.06.005

Joshi, A. (2020). *Machine Learning and Artificial Intelligence* (1st ed.). Springer. https://doi.org/10.1007/978-3-030-26622-6

Kapadia, S. (2020, February 26). *Recommendation System in Python: LightFM | by Shashank Kapadia | Towards Data Science*. https://medium.com/towards-data-science/recommendation-system-in-python-lightfm-61c85010ce17

Karatzoglou, A., Amatriain, X., Baltrunas, L., & Oliver, N. (2010). Multiverse Recommendation: N-dimensional Tensor Factorization for context-aware Collaborative Filtering. *RecSys'10 - Proceedings of the 4th ACM Conference on Recommender Systems*, 79–86. https://doi.org/10.1145/1864708.1864727

Kudo, T., & Matsumoto, Y. (2003). *Fast Methods for Kernel-Based Text Analysis*. 24–31. https://doi.org/10.3115/1075096.1075100

Lafferty, J., McCallum, A., & Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *ICML '01 Proceedings of the Eighteenth International Conference on Machine Learning*, *8*(June), 282–289. https://doi.org/10.1038/nprot.2006.61

Larry Hardesty. (2019, November 22). *The history of Amazon's recommendation algorithm.* Collaborative Filtering and Beyond.

https://www.cse.iitk.ac.in/users/nsrivast/HCC/Recommender_systems_handbook.pdf

Le, Q., & Mikolov, T. (2014). Distributed Representations of Sentences and Documents. *31st International Conference on Machine Learning, ICML 2014*, *4*, 2931–2939. https://arxiv.org/abs/1405.4053v2

Liang, H., & Wang, K. (2018). Top-k Route Search through Submodu-larity Modeling of Recurrent POI Features. *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. https://doi.org/10.1145/3209978

Lops, P; de Gemmis, M; Semeraro, G. (2011). *Content-based Recommender Systems: State of the Art and Trends*. 73–105. https://doi.org/10.1007/978-0-387-85820-3_3

Maroto Mariana. (2021, July 21). *Collaborative Filtering and some history on The Netflix Prize*. https://www.cse.iitk.ac.in/users/nsrivast/HCC/Recommender_systems_handbook.pdf

Medeiros, J. P. R. dos S. e. (2020). *Tourist Route Recommendation*. Universidade do Porto.

Mitchell, T. M. (1997). Machine Learning. In *Machine Learning* (Vol. 1, Issue Pt 1-2). Machine Learning V2. https://doi.org/10.1093/bioinformatics/btq112

Noleto C. (2022, September 9). *Google Colab: saiba o que é essa ferramenta e como usar! – Insights para te ajudar na carreira em tecnologia | Blog da Trybe*. https://blog.betrybe.com/carreira/google-colab/

Portugal, I., Alencar, P., & Cowan, D. (2018). The use of machine learning algorithms in recommender systems: A systematic review. *Expert Systems with Applications*, *97*, 205–227. https://doi.org/10.1016/J.ESWA.2017.12.020

Pullakandam Krishna. (2024, June 21). *Understanding Precision, Recall, and F-Score at K in Recommender Systems*. https://krishnapullak.medium.com/understanding-precision-recall-and-f-score-at-k-in-recommender-systems-7146a0dce68e

Qiao, S., Han, N., Zhou, J., Li, R. H., Jin, C., & Gutierrez, L. A. (2018). SocialMix: A familiarity-based and preference-aware location suggestion approach. *Engineering Applications of Artificial Intelligence*, *68*, 192–204. https://doi.org/10.1016/J.ENGAPPAI.2017.11.006

Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2012). BPR: Bayesian Personalized Ranking from Implicit Feedback. *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, UAI 2009*, 452–461. https://arxiv.org/abs/1205.2618v1

Roy, D., & Dutta, M. (2022). A systematic review and research perspective on recommender systems. *Journal of Big Data*, *9*(1), 1–36. https://doi.org/10.1186/S40537-022-00592-5/TABLES/2

Sang, J., Mei, T., Sun, J. T., Xu, C., & Li, S. (2012). Probabilistic sequential POIs recommendation via check-in data. *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, 402–405. https://doi.org/10.1145/2424321.2424375

Shi, K., Zhang, J., Fang, L., Wang, W., & Jing, B. (2024). *Enhanced Bayesian Personalized Ranking for Robust Hard Negative Sampling in Recommender Systems*. 9. https://arxiv.org/abs/2403.19276v1

Toledo, M. (2022). *Redes Neurais para Sistemas de Recomendação: uso de Redes Neurais Recorrentes para tratamento de Cold-Start Problem - Livros Acadêmicos com até 10% OFF* (Editora Di). https://loja.editoradialetica.com/ciencias-exatas-e-tecnologias/redes-neurais-para-sistemas-de-recomendacao-uso-de-redes-neurais-recorrentes-para-tratamento-de-cold-start-problem?srsltid=AfmBOor01UTkVQDWLpE4lI5CxMFIm1lC9L6bbwl56V0jEIkHepKIcIut

Velardo V. (2019, February 11). *Spotify's Discover Weekly explained — Breaking from your music bubble or, maybe not? | by Valerio Velardo | The Sound of AI | Medium*. https://medium.com/the-sound-of-ai/spotifys-discover-weekly-explained-breaking-from-your-music-bubble-or-maybe-not-b506da144123

Viniski A. (2021, January 7). *O que fazem os sistemas de recomendação?* https://www.linkedin.com/pulse/o-que-fazem-os-sistemas-de-recomendação-antonio-david-viniski

Wang, H., Fu, Y., Wang, Q., Yin, H., Du, C., & Xiong, H. (2017). A location-sentiment-aware recommender system for both home-town and out-of-town users. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *Part F129685*, 1135–1143. https://doi.org/10.1145/3097983.3098122

Werneck, H., Silva, N., Viana, M. C., Mourão, F., Pereira, A. C. M., & Rocha, L. (2020). A Survey on Point-of-Interest Recommendation in Location-based Social Networks. *ACM International Conference Proceeding Series*, 185–192. https://doi.org/10.1145/3428658.3430970

Wilson, T. and, Wiebe, J. and, & Hoffmann, P. (2005). *Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis* (R. and Mooney, C. and Brew, L.-F. and Chien, & K. Kirchhoff, Eds.). https://www.cse.iitk.ac.in/users/nsrivast/HCC/Recommender_systems_handbook.pdf

Xiong, X., Qiao, S., Han, N., Xiong, F., Bu, Z., Li, R. H., Yue, K., & Yuan, G. (2020). Where to go: An effective point-of-interest recommendation

framework for heterogeneous social networks. *Neurocomputing*, *373*, 56–69. https://doi.org/10.1016/J.NEUCOM.2019.09.060

Yang, D., Qu, B., Yang, J., Cudre-Mauroux, P., & Cudre, P. (2019). *Revisiting User Mobility and Social Relationships in LBSNs: A Hypergraph Embedding Approach*. 11. https://doi.org/10.1145/3308558.3313635

Yang, Di., Qu, B., Yang, J., & Cudre-Mauroux, P. (2020). LBSN2Vec++: Heterogeneous Hypergraph Embedding for Location-Based Social Networks. *IEEE Transactions on Knowledge and Data Engineering*, *34*(4), 1843–1855. https://doi.org/10.1109/TKDE.2020.2997869

Yuan, T., Cheng, J., Zhang, X., Liu, Q., & Lu, H. (2015). How friends affect user behaviors? An exploration of social relation analysis for recommendation. *Knowledge-Based Systems*, *88*, 70–84. https://doi.org/10.1016/J.KNOSYS.2015.08.005

Zeng, Y., Chen, X., Cao, X., Qin, S., Cavazza, M., & Xiang, Y. (2015). Optimal Route Search with the Coverage of Users' Preferences. *Undefined.*

Zhang, J. D., & Chow, C. Y. (2015). GeoSoCa: Exploiting geographical, social and categorical correlations for point-of-interest recommendations. *SIGIR 2015 - Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 443–452. https://doi.org/10.1145/2766462.2767711

ANEXOS

Apêndice A Script de Aquisição de Dados através do Foursquare

O código apresentado na Figura 12 utiliza a API do Foursquare para coletar *tips* (dicas ou avaliações) associadas a um local específico. Ele é composto por funções que fazem requisições à API, utilizando as credenciais fornecidas para autenticação.

```python
def get_tips(venue_id):
    url = f'https://api.foursquare.com/v2/venues/{venue_id}/tips'
    params = {
        'client_id': CLIENT_ID,
        'client_secret': CLIENT_SECRET,
        'v': VERSION,
        'limit': 50  # Límite de reviews a obtener por solicitud
    }
    response = requests.get(url, params=params)
    if response.status_code == 200:
        return response.json()['response']['tips']['items']
    else:
        return []

# Ejemplo de uso
venue_id = 'EjemploVenueID'
tips = get_tips(venue_id)
print(tips)

def get_tips(venue_id):
    url = f'https://api.foursquare.com/v2/venues/{venue_id}/tips'
    params = {
        'client_id': CLIENT_ID,
        'client_secret': CLIENT_SECRET,
        'v': VERSION,
        'limit': 50  # Límite de reviews a obtener por solicitud
    }
    response = requests.get(url, params=params)
    if response.status_code == 200:
        return response.json()['response']['tips']['items']
    else:
        return []
```

Figura 12 - Script Aquisição dados Foursquare

Apêndice B Script - Filtragem POIS em Portugal

O Código apresentado na Figura 13 tem como objetivo processar um conjunto de dados de POIs. Ele carrega os dados brutos a partir de um arquivo de texto (raw_POIs.txt), que contém informações como *venue_id, latitude, longitude, categoria do local e código do país*. O script filtra os POIs localizados em Portugal (código do país "PT").

```
# Cargar el dataset, asumiendo que el delimitador es una tabulación (tab-separated values)
columns = ['venue_id', 'latitude', 'longitude', 'venue_category', 'country_code']
pois = pd.read_csv(file_path, delimiter='\t', header=None, names=columns)


# Filtrar los POIs que están en Portugal (código de país 'PT')
portugal_pois_long = pois[pois['country_code'] == 'PT']

# Guardar los datos filtrados en un nuevo archivo CSV
output_path = 'C:/Users/a2019100689/.spyder-py3/TESE/Conjunto de Dados_Foursquare/pois_portugal_
portugal_pois_long.to_csv(output_path, index=False)

print(f"Datos filtrados guardados: {len(portugal_pois_long)} registros.")
```

Figura 13 - Filtragem POIs PT

Apêndice C Filtragem Categorias

Na Figura 14 o objetivo é filtrar os registos com base na categoria do local, excluindo aqueles que pertencem a categorias residenciais ou administrativas, como "*Home (private)" ou "City"*. Para cada linha que não corresponde a essas categorias, o script escreve os dados filtrados (id e categoria) em um novo arquivo, utilizando o método *employee_writer.writerow()*[52].

```
# Filtragem categorias
categorias_a_eliminar = [
    'Assisted Living', 'Home (private)', 'Housing Development',
    'Residential Building (Apartment / Condo)', 'Trailer Park',
    'City', 'County', 'Country', 'Neighborhood', 'State', 'Town',
    'Village', 'States & Municipalities'
]
```

Figura 14 - Filtragem categorias eliminadas

Apêndice D Script Local Metadata

Na Figura 15 para cada ID, o script realiza uma consulta na API, recuperando detalhes como nome, descrição, categoria, classificação, latitude, longitude e status de verificação.

[52] https://docs.python.org/3/library/csv.html (último acesso março 2025).

```
#Credenciais
client_id = 'Z3R5DVUQJT01QAVNO53LKBHBKGUUWAWGCK5QWHUWD3JSYD21'
client_secret = '5QSW3AH2QJRNGUDVL4LFGD1HTGZ3NDMTIRDUZWGBVXBM5H30'
redirect_id = 'https://foursquare.com'


client = fs.Foursquare(client_id=client_id, client_secret=client_secret)
with open('local_metadata.csv',  'w', newline='', encoding='utf8') as employee_file:
    fieldnames = ['local_id','name','rating', 'categories',  'description', 'rating', 'latitude', 'longitude', 'verified']
    employee_writer = csv.DictWriter(employee_file, delimiter=',', quotechar='"', quoting=csv.QUOTE_MINIMAL, fieldnames = fieldnam
    f = open('C:/Users/a2019100689/.spyder-py3/TESE/TEXTO/local_metadata/pois_filtrados_portugal.csv','r')

    for i in f:
        idlocal=i.strip()
        query = client.venues(idlocal)
        description=query['venue']['description']
        rating=query['venue']['rating']
        name=query['venue']['name']
        categories=query['venue']['categories'][0]
        verified=query['venue']['verified']

        employee_writer.writerow({'local_id': i,'name':name, 'categories':categories, 'description':description, 'rating':rating,
```

Figura 15 - Aquisição dos 1379 registos após filtragem

Apêndice E Script da Aquisição Conjunto de Dados Ranking

Na Figura 16 são processados dados de *check-ins* e categorias de locais para calcular *ratings* personalizados. Ele lê dois arquivos de entrada: check_user_category.csv, contendo informações sobre *check-ins* de utilizadores em locais específicos e suas categorias, e groupby_1.csv, que agrupa os utilizadores por categorias com a contagem total de *check-ins*.

Quando há correspondência, calcula um *rating* baseado na proporção de check-ins do utilizador em um local específico em relação ao total de *check-ins* na mesma categoria.

```
import pandas as pd
from collections import Counter
import io
import csv

with io.open('rating_new_2.csv', mode='w', encoding='utf8', newline='') as employee_file:
    fieldnames = ['id-user','id_local','rating']
    employee_writer = csv.DictWriter(employee_file, delimiter=',', quotechar='"', quoting=csv.QUOTE_MINIMAL,

    with open ('check_user_category.csv','r') as C:
        reader = csv.reader(C, delimiter= ',')
        for line in reader:
            x=line[0] #id user
            z=line[1] #local
            y=line[2] #countchecks
            w=line[3]#category
            with open ('groupby_1.csv','r') as D:
                reader2 = csv.reader(D, delimiter= ',')
                ##############
                for line2 in reader2:
                    i=line2[0]#iduser
                    j=line2[1] #nome da categoria
                    k=line2[2] #count
                    if(int(x==i) & int(w==j)):
                        rating=(int(y)/int(k))

                        employee_writer.writerow({'id-user': x,'id_local':z , 'rating':rating})
```

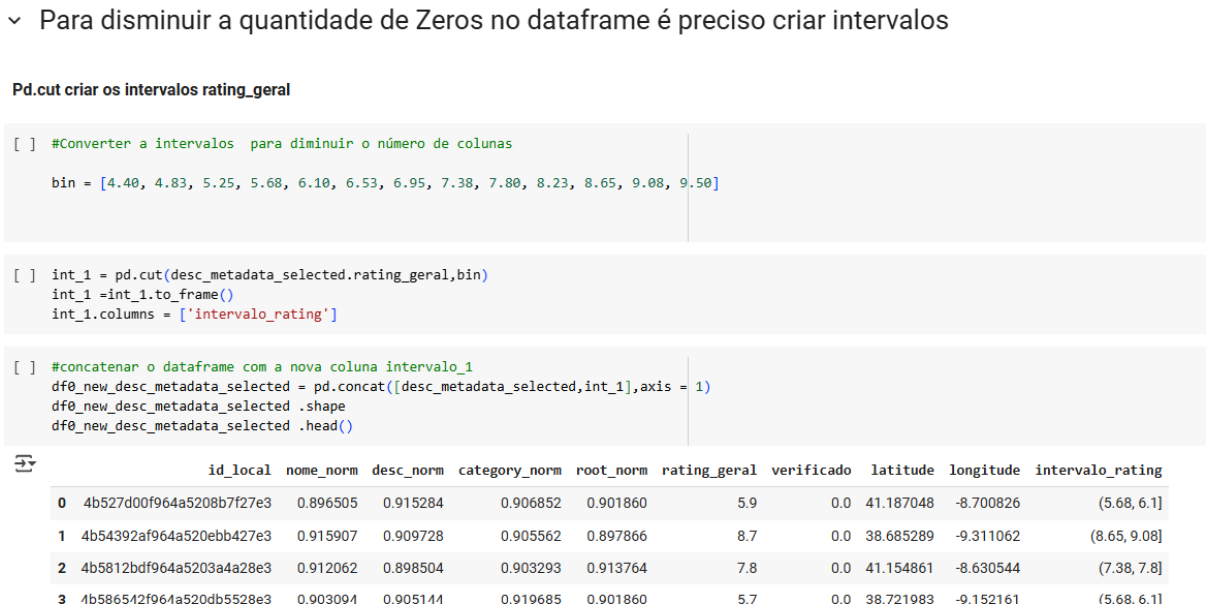
Figura 16 - Aquisição Conjunto de dados *Rating*

Apêndice F Reporte dos Dados Através da Biblioteca Sweetviz

Na Figura 17 descreve-se o processo de geração de um relatório descritivo utilizando a biblioteca **Sweetviz**. Este relatório foi fundamental para a análise exploratória de dados e para a definição dos intervalos *(bins)* no sistema de recomendação.

Figura 17 - Biblioteca Sweetviz

Apêndice G Intervalos *Bins*



	id_local	nome_norm	desc_norm	category_norm	root_norm	rating_geral	verificado	latitude	longitude	intervalo_rating
0	4b527d00f964a5208b7f27e3	0.896505	0.915284	0.906852	0.901860	5.9	0.0	41.187048	-8.700826	(5.68, 6.1]
1	4b54392af964a520ebb427e3	0.915907	0.909728	0.905562	0.897866	8.7	0.0	38.685289	-9.311062	(8.65, 9.08]
2	4b5812bdf964a5203a4a28e3	0.912062	0.898504	0.903293	0.913764	7.8	0.0	41.154861	-8.630544	(7.38, 7.8]
3	4b586542f964a520db5528e3	0.903094	0.905144	0.919685	0.901860	5.7	0.0	38.721983	-9.152161	(5.68, 6.1]

Figura 18 - Criação de intervalos com bins

Apêndice H Script DBSCAN *Clusters*

A Figura 18 apresenta a implementação do algoritmo DBSCAN para agrupar POIs com base em suas coordenadas geográficas (latitude e longitude).

O modelo é configurado com um epsilon (eps) de 0.01 e um número mínimo de 5 amostras por *cluster* (min_samples=5), parâmetros que definem a densidade necessária para formar um agrupamento. Após o ajuste do modelo, os *clusters* resultantes são atribuídos a uma nova coluna chamada '*cluster*', permitindo a identificação dos grupos formados.

DBSCAN

```
# Import the DBSCAN class from sklearn.cluster
from sklearn.cluster import DBSCAN

# Aplicar DBSCAN nas coordenadas de latitude e longitude
lat_long = df3_new_desc_metadata_selected[['latitude', 'longitude']].values

# Ajuste de eps e min_samples conforme o contexto dos dados
dbscan = DBSCAN(eps=0.01, min_samples=5)
clusters = dbscan.fit_predict(lat_long)

# Adicionar os clusters ao DataFrame
df3_new_desc_metadata_selected['cluster'] = clusters

# Verificar os clusters criados
print(df3_new_desc_metadata_selected['cluster'].value_counts())
```

```
[ ] # Remover a coluna original de 'latitude  longitude' se não for mais necessária
    df3_new_desc_metadata_selected.drop([ 'latitude', 'longitude'], axis=1, inplace=True)
```

```
[ ] print(df4_4_new_desc_metadata_selected.dtypes)
```

```
id_local               object
nome_norm             float32
desc_norm             float32
category_norm         float32
root_norm             float32
rating_geral          float64
verificado            float64
intervalo_rating     category
intervalo_nome       category
intervalo_desc       category
intervalo_category   category
intervalo_root       category
cluster                 int64
dtype: object
```

Figura 19 - Clusterização DBSCAN

Apêndice I Métricas Utilizadas para a Avaliação

Durante o processo de avaliação do desempenho do modelo de recomendação, foram utilizadas diversas métricas para analisar a qualidade das recomendações a biblioteca LightFM fornece alguns modelos de avaliação como *LightFM.evaluation.auc_score, LightFM.evaluation.precision_at_k, LightFM.evaluation.recall_at_k e LightFM.evaluation.reciprocal_rank* tal como na Figura 20.

```
[ ] #avaliar para observar a precisão
    train_precision = precision_at_k(model, train, k=5, item_features=desc_metadata_csr).mean()
    test_precision = precision_at_k(model, test, k=5, train_interactions=train, item_features=desc_metadata_csr).mean()

    # Avaliar para observar AUC
    train_auc = auc_score(model, train, item_features=desc_metadata_csr).mean()  # Correção aqui
    test_auc = auc_score(model, test, train_interactions=train, item_features=desc_metadata_csr).mean()  # Correção aqui

    # Avaliar para observar Recall
    recall_train = recall_at_k(model, train, k=5, item_features=desc_metadata_csr).mean()  # Correção aqui
    recall_test = recall_at_k(model, test, k=5, item_features=desc_metadata_csr, train_interactions=train).mean()  # Correção aqui

    # Avaliar para observar Mean Reciprocal Rank (MRR)
    mrr_train = reciprocal_rank(model, train, item_features=desc_metadata_csr).mean()  # Correção aqui
    mrr_test = reciprocal_rank(model, test, item_features=desc_metadata_csr, train_interactions=train).mean()  # Correção aqui

    print('Precision: train %.2f, test %.2f.' % (train_precision, test_precision))
    print('AUC: train %.2f, test %.2f.' % (train_auc, test_auc))
    print('Recall: train %.2f, test %.2f.' % (recall_train, recall_test))
    print('MRR: train %.2f, test %.2f.' % (mrr_train, mrr_test))

    Precision: train 0.11, test 0.08.
    AUC: train 0.98, test 0.98.
    Recall: train 0.47, test 0.40.
    MRR: train 0.25, test 0.22.
```

Figura 20 - Avaliação performance

Apêndice J Criação de um Dicionário de Itens

```
[ ] #Data Preprocessing

    item_dict ={}
    df = desc_metadata[['id_local', 'nome']].sort_values('id_local').reset_index()

    for i in range(df.shape[0]):
        item_dict[(df.loc[i,'id_local'])] = df.loc[i,'nome']

[ ] # simular características categóricas
    desc_metadata_selected_transformed = pd.get_dummies( df3_new_desc_metadata_selected, columns=['intervalo_rating','verificado', 'intervalo_category','intervalo_nome','intervalo_desc'])
    desc_metadata_selected_transformed = desc_metadata_selected_transformed.sort_values('id_local').reset_index().drop('index', axis=1)

[ ] # Remover a coluna 'id_local' e converter todas as colunas para valores numéricos
    desc_metadata_selected_transformed_numeric = desc_metadata_selected_transformed.drop('id_local', axis=1)

    # Garantir que todas as colunas tenham tipos numéricos
    desc_metadata_selected_transformed_numeric = desc_metadata_selected_transformed_numeric.astype(float)
    desc_metadata_selected_transformed.head(5)
```

Figura 21 - Mapeamento de indicadores

A Figura 21 apresenta o mapeamento representado por um dicionário, utilizado para associar os identificadores únicos dos locais (POIs) aos seus nomes correspondentes, com o objetivo de facilitar a interpretação e visualização dos resultados no sistema de recomendação.

Instituto Superior de Engenharia
Politécnico de Coimbra