

基于 Python 的旅游景点推荐系统

吴与同^{1,2}

1. 江苏师范大学机械工程学院, 江苏徐州, 221116

2. 江苏师范大学江苏理工学院—中俄学院, 江苏徐州, 221116

摘要 | 为降低旅游信息平台中用户筛选信息的成本, 并解决个性化推荐不足与行程规划效率低的问题, 本文设计并实现基于Python和Django的旅游景点推荐系统。系统利用Requests与BeautifulSoup4采集携程网景点和餐厅数据, 共获得景点8959条、餐厅2678条; 采用基于物品的协同过滤算法 (ItemCF), 融合收藏、点赞、评分、加入行程和不感兴趣等行为权重, 并以热门景点策略缓解冷启动和候选不足。同时接入DeepSeek、百度地图和OpenWeather, 实现智能问答、线路规划、地图展示与天气查询。测试表明, 系统主要功能运行稳定, AI线路规划响应时间不超过3s, 能够满足中小型旅游推荐系统的工程应用需求。

关键词 | 旅游景点推荐系统; Python语言; 协同过滤; 网络爬虫; 大语言模型

Copyright © 2026 by the author(s).

This article is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/).

<https://creativecommons.org/licenses/by-nc/4.0/>



0 引言

随着旅游信息平台持续积累景点、餐饮、评价和位置等多源数据, 用户虽然能够获得更丰富的信息, 但也面临检索结果规模扩大、选择成本上升以及个性化不足等问题^[1]。旅游景点推荐的核心任务是利用用户历史行为、景点属性及上下文信息缩小候选空间, 并在准确性、实时性、可解释性与工程成本之间取得平衡。

Werneck等对位置社交网络中的POI推荐进行了系统梳理, 指出数据稀疏、时空上下文利用和评价维度是该领域的重要问题^[2]。国内研究方面, 李志豪从深度学习角度研究个性化推荐^[3], 徐中彦讨论深度学习与知识图谱融合^[4], 姚南、王宏生关注图神经网络在推荐中的应用^[5], 于翹楚等则通过混合

协同过滤与权重优化改善传统方法^[6]。上述研究表明, 旅游推荐已由单一协同过滤逐步向多源信息融合和高阶关系建模扩展。

传统协同过滤具有结构简单、解释直观和在线计算链路短等优势。Sarwar等提出的Item-based协同过滤通过相对稳定的物品相似关系生成推荐^[7]; Koren等进一步以矩阵分解刻画潜在用户—物品因子^[8]。随着深度学习的发展, NCF以神经网络学习非线性交互函数^[9], SASRec^[10]和BERT4Rec^[11]利用自注意力建模行为序列; NGCF^[12]、LightGCN^[13-14]与自监督图学习通过用户—物品图传播高阶协同信号, KGAT等方法进一步将知识图谱中的实体关系纳入推荐建模^[15]。

在POI场景中, 近期研究更加重视位置、时间、轨迹和异构信息。HMT-GRN通过区域层级任务缓解用户—POI矩阵稀疏问题^[16], GETNext将轨迹流图

作者简介: 吴与同, 江苏师范大学机械工程学硕士研究生, 研究方向为深度学习算法和表面增强拉曼光谱。

文章引用: 吴与同. 基于 Python 的旅游景点推荐系统[J]. 计算机与智能教育技术, 2026, 1(1): 1-15.

<https://doi.org/10.68070/vksvmf70>

与Transformer结合以融合协同与时空信号^[17], Ding利用图注意力和知识图谱融合多源景点及路线信息^[18]。Wang等对2021—2023年POI推荐研究的综述进一步指出, 异构数据与上下文融合仍具有较大提升空间^[19]。

与上述复杂模型相比, 本文研究目标更偏向中小型Web系统的可落地实现。考虑到景点集合相对稳定、用户交互可持续积累以及系统对实时响应和维护成本的要求, 本文采用ItemCF作为核心推荐算法, 在行为层融合收藏、点赞、评分、加入行程和负反馈, 并以热门景点策略处理冷启动和候选不足。本文的贡献主要体现在推荐逻辑与数据采集、地图、天气、大语言模型和管理功能的工程集成,

而非提出新的推荐模型; 因此, 本文将重点讨论该方案的适用边界与后续扩展方向。

在上述研究基础与方法定位下, 系统采用Django构建业务逻辑, MySQL存储景点、餐厅、用户及行为数据, 并接入DeepSeek、百度地图和OpenWeather, 形成“数据采集—偏好建模—推荐生成—行程辅助—系统验证”的业务闭环。本文主要工作包括: 采集并清洗景点与餐厅数据, 设计基于ItemCF的个性化推荐机制, 构建面向普通用户和管理员的业务模块, 并通过功能、性能及离线模拟实验验证系统的工程可用性与推荐有效性。系统总体架构如图1所示。

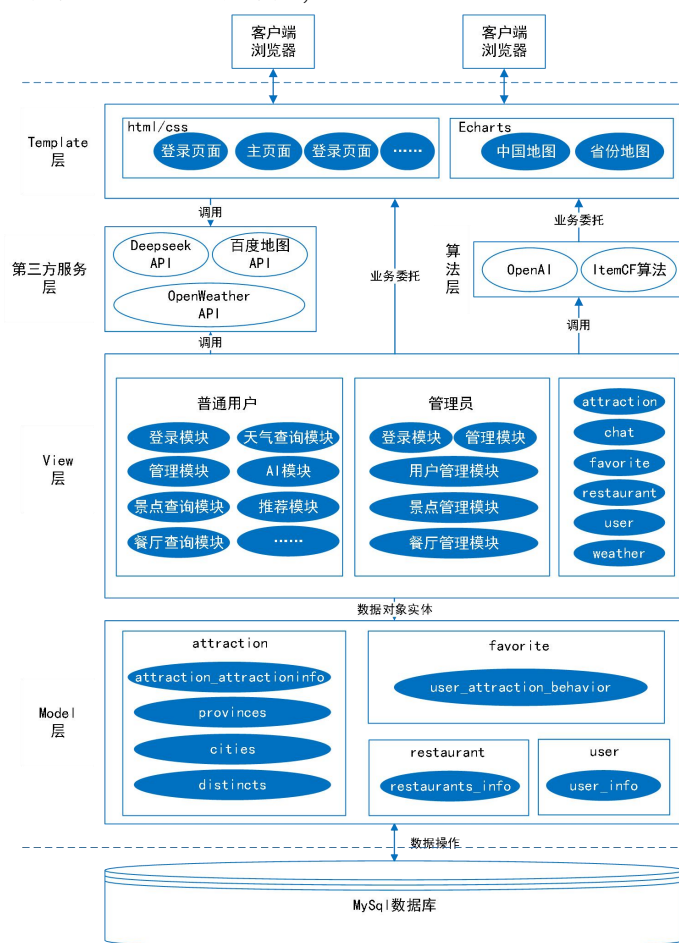


图 1 系统总体架构

Figure 1 Overall System Architecture

1 系统总体设计

1.1 系统架构

系统采用B/S架构, 核心开发语言为Python, Web 框架采用Django, 数据库采用MySQL。为降低

模块耦合度, 将整体功能划分为表示层、业务层、数据层、算法与第三方服务层。浏览器端通过Django Template、HTML、CSS、JavaScript和ECharts完成交互与数据可视化; 业务层按attraction、restaurant、favorite、user、chat、weather等模块组织。

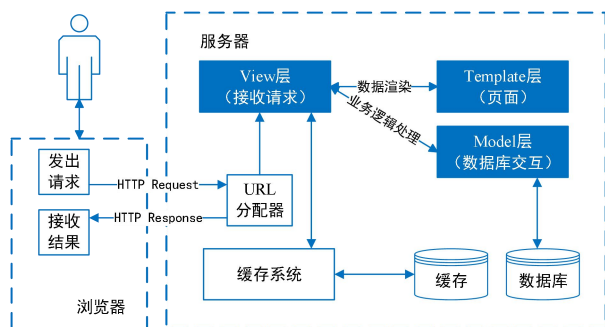


图2 Django 架构示意图

Figure 2 Schematic Diagram of the Django Architecture

1.2 MVT 分层与模块协同

Django的Model层负责对象映射和数据增删改查，View层完成登录认证、推荐计算、AI接口转发和统计聚合，Template层负责结果展示。普通用户可进行景点与餐厅检索、天气查询、AI问答及推荐交互；管理员主要完成用户、景点和餐厅数据维护。通过normal与admin角色字段区分权限，并对需授权的视图进行身份校验。

收藏、点赞、评分、加入行程和“不感兴趣”等操作统一写入行为记录，既更新页面状态，也形成推荐计算输入。地图、天气和AI功能与推荐模块保持相对独立，当第三方服务异常时，本地景点检索和ItemCF推荐仍可运行。

1.3 数据获取与预处理

相关研究从不同角度讨论了Python网页采集技术：姜庆玲、张樊分析了Requests爬取实现^[20]，鞠慧介绍了BeautifulSoup网页结构解析方法^[21]，李晨昊比较了BeautifulSoup+Requests与Selenium在网页自动化处理中的实现与性能差异^[22]。结合本系统目标页面结构相对固定、采集字段明确的特点，本文采用Requests与BeautifulSoup4构建定向爬虫，采集携程网国内景点与餐厅信息。清洗后共获得景点8959条、餐厅2678条。景点数据包括名称、区域、评分、评论数、票价、开放时间、经纬度和标签等；餐厅数据包括名称、菜系、评分、人均消费、特色美食和地址等。采集完成后对缺失值、字段类型、区域名称及重复记录进行统一处理并写入MySQL。

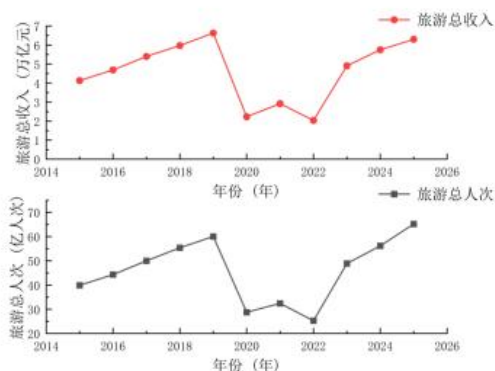


图3 2015—2025 年国内旅游总人次与总收入变化趋势

Figure 3 Domestic tourism trips and revenue, 2015–2025

注：数据来源：文化和旅游部《2025年文化和旅游发展统计公报》^[23]。

从图3可以看出，国内旅游市场在经历阶段性波动后持续恢复，2025年国内居民出游人次与出游花费均较前期明显增长^[23]。旅游市场规模的恢复意味着用户对多样化旅游信息的需求进一步提升，也增强了个性化筛选和行程辅助的现实意义。

1.4 第三方服务与数据支撑

系统通过统一接口管理外部服务：百度地图用于景点、餐厅定位及线路可视化，OpenWeather用于天气与气象图层查询，DeepSeek用于旅游问答和结构化线路规划。第三方接口返回结果在进入模板前进行异常判断和格式规范；AI线路结果经清理、表头识别和DataFrame转换后，统一经纬度及费用字段，再传递给页面与地图模块。

1.5 业务需求

普通用户需要快速获得景点、餐厅、天气和路线信息，并通过收藏、点赞、评分等交互逐步形成个性化偏好；管理员需要维护用户、景点和餐厅数据并掌握数据变化情况。系统因此将“内容检索—行为记录—个性推荐—路线规划”组织为连续业务链，历史反馈持续写入数据库，推荐时排除用户已操作景点，并根据正负反馈更新候选得分。

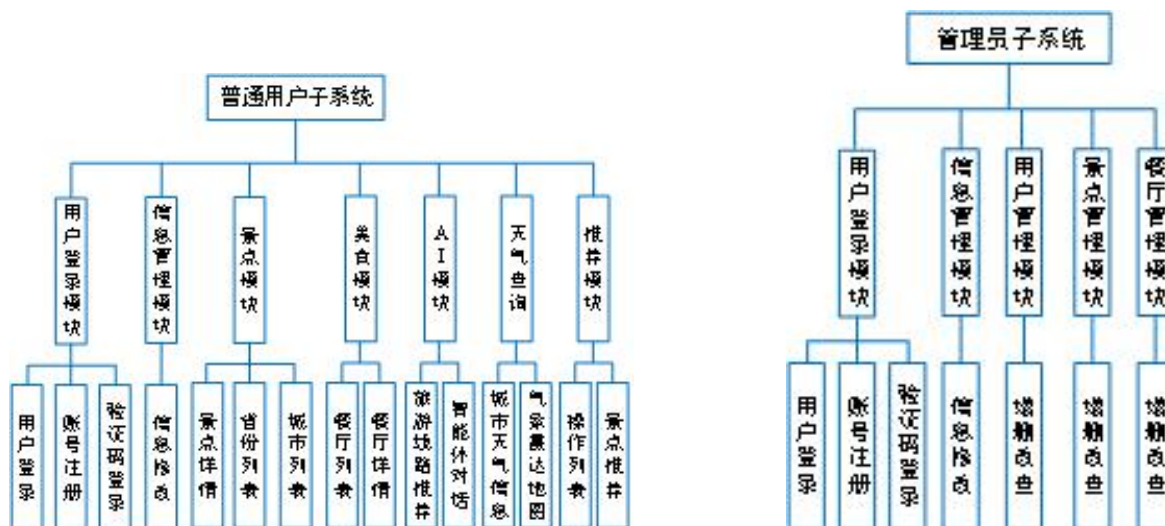


图 4 功能模块图

Figure 4 Functional Block Diagram

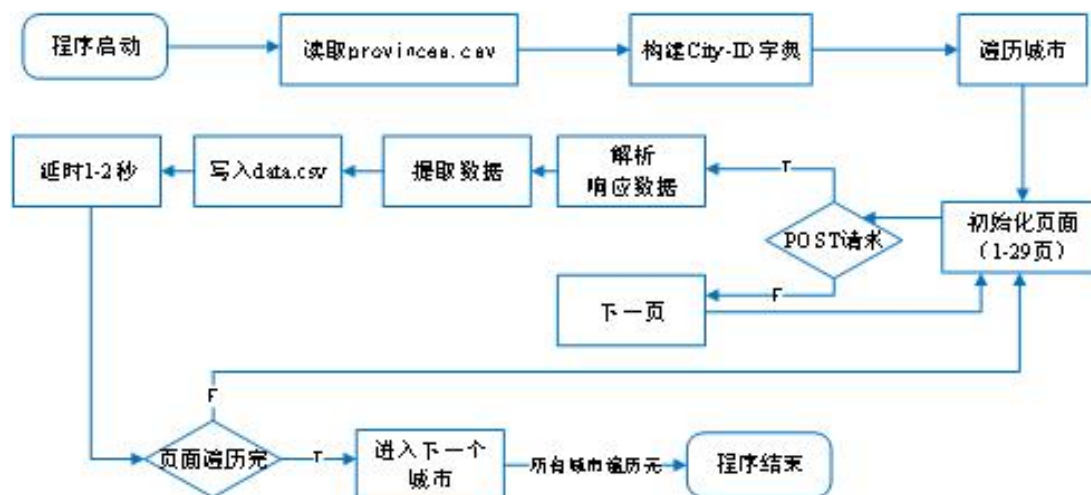


图 5 景点数据爬取逻辑

Figure 5 Logic for Scraping Attraction Data

2 推荐算法与数据模型设计

2.1 ItemCF 推荐模型

协同过滤 (Collaborative Filtering, CF) 通过用户—物品交互关系发现邻域结构, 按核心挖掘对象可分为基于用户的协同过滤 (User-Based Collaborative Filtering, UserCF) 和基于物品的协同过滤 (Item-Based Collaborative Filtering, ItemCF)。Sarwar等系统分析了Item-based协同过滤的物品相似度与推荐生成机制。UserCF需要在用户邻域中寻找兴趣相近用户, 当用户规模扩大且交互较稀疏时, 邻域维护和在线相似用户搜索成本会增加; ItemCF

则可在物品集合相对稳定的场景中预计算或缓存物品相似关系, 从而降低在线阶段的计算压力。国内已有研究将协同过滤用于金融产品推荐^[24]和广播电视节目推荐^[25]。结合本系统景点集合相对稳定、用户行为持续积累的特点, 本文采用ItemCF算法。

ItemCF算法主要在于挖掘旅游景点间的关联关系, 从而达到个性化推荐的目的。其主要计算过程分为物品相似度计算和推荐评分预测两部分, 公式如下^[26]。

(1) 物品相似度计算 (采用余弦相似):

设表示用户对景点的行为反馈权重, 为同时对景点、产生有效行为的用户集合, 物品相似度采用余弦相似度计算:

$$\text{sim}(i, j) = \frac{\sum_{u \in U_{ij}} r_{ui} \cdot r_{uj}}{\sqrt{\sum_{u \in U_i} r_{ui}^2} \cdot \sqrt{\sum_{u \in U_j} r_{uj}^2}} \quad (1)$$

其中， $\text{sim}(i, j)$ 表示景点*i*与景点*j*的相似度。由于本文的行为反馈权重同时包含正值和负值，景点向量可能含负分量，因此余弦相似度的理论取值范围为[-1,1]。 $\text{sim}(i, j) > 0$ 表示两景点在用户行为空间中的反馈方向较一致，值越接近1关联越强； $\text{sim}(i, j) = 0$ 表示二者在该空间中近似正交； $\text{sim}(i, j) < 0$ 表示反馈方向相反，值越接近-1负相关越强。

(2) 推荐评分预测：

对于目标用户*u*，候选景点的推荐得分由用户历史交互景点与候选景点之间的相似度进行加权累加得到，其计算公式为：

$$\text{score}(u, j) = \sum_{i \in I_u} (r_{ui} \cdot \text{sim}(i, j)) \quad (2)$$

其中， I_u 表示用户*u*已产生有效交互行为的景点集合。推荐得分越高，说明候选景点与用户历史偏好

的匹配程度越高。

矩阵构建时只保留有效行为记录，并按用户ID和景点ID进行透视。若同一用户对同一景点存在多次操作，则依据当前有效状态和行为类型形成综合反馈。这样能够避免取消收藏、重复点赞等历史记录对当前偏好造成干扰。最终按照推荐分数降序排列候选景点，并过滤用户已经操作过的对象。

ItemCF的优势在于物品相似关系相对稳定，适合景点集合变化频率低于用户访问频率的场景。相似度矩阵可按一定周期重新生成或缓存，从而把在线请求阶段的主要计算简化为用户行为向量与相似景点集合的加权累积，降低每次推荐的响应压力。

2.2 用户行为权重与冷启动

系统将不同交互映射为正向或负向权重。收藏、加入行程和点赞分别表示不同强度的兴趣，评分根据1—5分形成正负反馈，取消收藏和“不感兴趣”用于降低相似景点的推荐概率。具体权重见表1。

表1 用户行为及权重

Table 1 User Behavior and Weights

行为	标识	权重
收藏	1	6.0
加入行程	5	5.0
点赞	2	4.0
评分 5/4	3	2.0 / 1.0
评分 1/2/3	3	-3.0 / -2.0 / -1.0
取消收藏	6	-4.0
不感兴趣	7	-10.0

注：正值表示正向偏好，负值表示负向反馈；行为权重为结合系统交互语义设定的工程参数，用于统一不同反馈类型的相对强度，不代表通过数据驱动方法学习得到的最优参数。

推荐生成时先排除当前用户已有有效行为的景点，只保留可推荐候选。当用户行为不足、目标用户未出现在评分矩阵中或个性化候选过少时，系统使用热门景点补齐；补齐过程仍排除已操作对象，最终返回前12条结果。

2.3 数据库结构设计

数据库围绕用户、景点、餐厅、行政区划和行为记录建立实体关系，如图6所示。本系统共设计 user_info 、 attraction_attractioninfo 、 restaurants_info 、 provinces 、 cities 、 districts 与 user_attraction_behavior等7张表，如图7所示，其中用户实体ID号为主码，景点实体ID号为主码，省份名称、城市名称为外码，关联省份实体和城市实体，餐厅实体ID号为主码，所在城市、所在区域为外码，关联城市实体和省份实体，省份实体省份名称为主码，城市实体城市名称为主码，区县实体省

份名称、城市名称、区县名称为联合主码，省份名称、城市名称为外码，关联省份实体和城市实体，行为实体ID号为主码，景点ID号、用户ID号为外码，关联景点实体和用户实体。详细表结构设计见表2—表8。

2.4 数据一致性与行为记录

用户对景点执行收藏、点赞、评分、加入行程、取消收藏和不感兴趣等操作时，系统不仅更新页面状态，还写入统一行为表。推荐算法只读取有效记录，由 behavior_type与behavior_value转换为权重，因此数据库结构与算法输入保持一致。

这一设计使前台展示、后台统计与推荐算法共享同一套业务数据，减少重复数据维护。

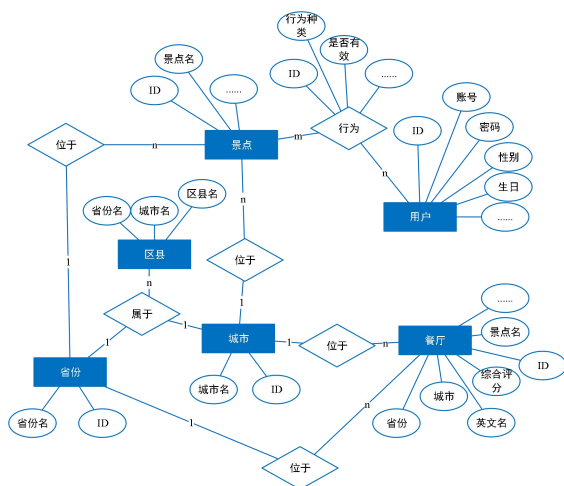


图 6 实体联系 E-R 图

Figure 6 Entity-Relationship Diagram

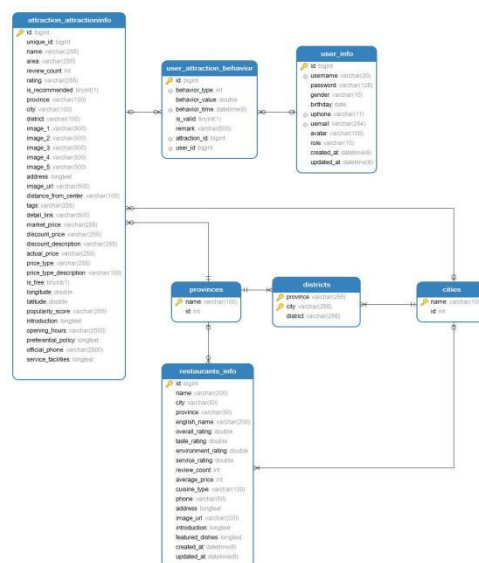


图 7 主要数据表逻辑关系

Figure 7 Logical relationships among major data tables

表 2 行为表 (user_attraction_behavior 表) 结构设计

Table 2 Structure Design of the Behavior Table (user_attraction_behavior)

字段名称	类型	是否为空	主码/外码	字段说明
id	bigint	否	主码	ID
behavior_type	int	否	—	行为类型
behavior_value	float	是	—	行为值
behavior_time	datetime	否	—	行为发生时间
is_valid	tinyint(1)	否	—	行为是否有效
remark	varchar(500)	是	—	行为备注
attraction_id	bigint	是	外码	景点 ID
user_id	bigint	是	外码	用户 ID

表 3 区县信息表 (districts 表) 结构设计

Table 3 Structural Design of the Districts Table

字段名称	类型	是否为空	主码/外码	字段说明
province	varchar(100)	不为空	主码, 外码	省份名称
city	varchar(100)	不为空	主码, 外码	城市名称
district	varchar(100)	不为空	主码	区县名称

表 4 省份信息表 (provinces 表) 结构设计

Table 4 Structural Design of the Provinces Table

字段名称	类型	是否为空	主码/外码	字段说明
name	varchar(100)	不为空	主码	省份名称
id	int	不为空	唯一键	省份 ID

表 5 城市信息表 (cities 表) 结构设计

Table 5 Structural Design of the Cities Table

字段名称	类型	是否为空	主码/外码	字段说明
name	varchar(100)	不为空	主码	城市名称
id	int	不为空	唯一键	城市 ID

表 6 用户信息表 (user_info 表) 结构设计

Table 6 Structure Design of the User Information Table (user_info)

字段名称	类型	是否为空	主码/外码	字段说明
id	bigint	不为空	主码	ID
username	varchar(20)	不为空	唯一键	用户名
password	varchar(128)	不为空	—	密码
gender	varchar(10)	不为空	—	性别
续表6				
字段名称	类型	是否为空	主码/外码	字段说明
birthday	date	可为空	—	生日
uphone	varchar(11)	不为空	唯一键	电话
uemail	varchar(254)	不为空	唯一键	电子邮箱
avatar	varchar(100)	可为空	—	头像
role	varchar(10)	不为空	—	用户角色
created_at	datetime(6)	不为空	—	创建时间
updated_at	datetime(6)	不为空	—	更新时间

表 7 景点信息表 (attraction_attractioninfo 表) 结构设计

Table 7 Structure Design of the Attraction Information Table (attraction_attractioninfo Table)

字段名称	类型	是否为空	主码/外码	字段说明
id	bigint	不为空	主码	ID
unique_id	bigint	不为空	唯一键	唯一标识
name	varchar(255)	可为空	—	景点名称
area	varchar(255)	可为空	—	所在区域
review_count	int	可为空	—	点评数量
rating	varchar(255)	可为空	—	评分
is_recommended	tinyint(1)	可为空	—	是否被推荐
province	varchar(100)	可为空	外码	省份名称
city	varchar(100)	可为空	外码	城市名称
district	varchar(100)	可为空	—	区县名称
image_1	varchar(500)	可为空	—	图片1
image_2	varchar(500)	可为空	—	图片2
image_3	varchar(500)	可为空	—	图片3
image_4	varchar(500)	可为空	—	图片4
image_5	varchar(500)	可为空	—	图片5
address	longtext	可为空	—	地址
image_url	varchar(500)	可为空	—	图片链接
distance_from_center	varchar(100)	可为空	—	与市中心距离
tags	varchar(255)	可为空	—	标签
detail_link	varchar(500)	可为空	—	详情页链接
market_price	varchar(255)	可为空	—	市场票价
discount_price	varchar(255)	可为空	—	优惠票价格
discount_description	varchar(255)	可为空	—	优惠描述
actual_price	varchar(255)	可为空	—	实际售价
price_type	varchar(255)	可为空	—	价格类型
price_type_description	varchar(100)	可为空	—	价格类型描述
is_free	tinyint(1)	可为空	—	是否免费
longitude	double	可为空	—	经度
latitude	double	可为空	—	纬度
popularity_score	varchar(255)	可为空	—	热度评分
introduction	longtext	可为空	—	介绍
opening_hours	varchar(2500)	可为空	—	开放时间
preferential_policy	longtext	可为空	—	优待政策
official_phone	varchar(2500)	可为空	—	官方电话
service_facilities	longtext	可为空	—	服务设施

表 8 餐厅信息表 (restaurants_info 表) 结构设计

Table 8 Structure Design of the Restaurant Information Table (restaurants_info)

字段名称	类型	是否为空	主码 / 外码	字段说明
id	bigint	不为空	主码	ID
name	varchar(200)	不为空	—	餐厅名称
city	varchar(50)	不为空	外码	所在城市

续表8

字段名称	类型	是否为空	主码 / 外码	字段说明
province	varchar(50)	不为空	外码	所在区域
english_name	varchar(200)	可为空	—	英文名称
overall_rating	double	不为空	—	综合评分
taste_rating	double	不为空	—	口味评分
environment_rating	double	不为空	—	环境评分
service_rating	double	可为空	—	服务评分
review_count	int	不为空	—	评论数量
average_price	int	可为空	—	人均消费
cuisine_type	varchar(100)	可为空	—	菜系
phone	varchar(50)	可为空	—	联系电话
address	longtext	可为空	—	详细地址
image_url	varchar(200)	可为空	—	图片链接
introduction	longtext	可为空	—	餐厅介绍
featured_dishes	longtext	可为空	—	特色美食
created_at	datetime(6)	不为空	—	创建时间
updated_at	datetime(6)	不为空	—	更新时间

2.5 方法适用性与局限性

从方法机理看，ItemCF以用户对景点的共同交互为基础估计景点关联，优势是关系可解释、相似度可缓存、在线推荐计算量较低，适合景点集合相对稳定且需要快速响应的中小型平台。其局限主要体现在三个方面：一是不同交互行为使用固定权重，属于工程先验，无法自动适应不同用户和场景下的反馈强度；二是热门景点补齐可缓解冷启动，但可能放大热门偏置，降低长尾景点获得曝光的机会；三是当前模型未显式建模时间、距离、预算、天气及景点语义关系，对短期兴趣变化和复杂上下文偏好的表达能力有限。

图神经网络、序列模型和知识图谱推荐能够通过高阶连接、时序依赖与语义关系增强表示能力，但通常需要更充分的交互数据、更高的训练成本和更复杂的在线服务链路。基于本文以工程可用性为主要目标的定位，现阶段保留ItemCF作为核心方案，并将上下文融合、长尾公平性和可学习行为权重作为后续优化方向。

3 软件设计与功能实现

3.1 开发环境与项目组织

系统采用 Python 3.10.19、Django 4.2.16 和 MySQL 8.2 开发。项目按景点、餐厅、用户、收藏、AI、天气和推荐等业务模块组织，其中AI模块封装DeepSeek接口，CF模块实现ItemCF算法，数据访问统一通过Django ORM完成。

3.2 AI 旅游线路规划

AI旅游线路规划功能主要根据用户填写的出发地、目的地、旅游季节、行程天数和预算等信息生成旅游方案。系统调用DeepSeek接口时，在提示词

中对返回内容进行了格式限制，要求模型按照指定字段生成结果，便于后续程序进行读取和展示。系统提示词中规定返回内容采用Markdown表格形式，表格字段包括景点名称、游玩时间、景点特点、经度、纬度、预计花费、就餐地点、就餐花费、交通方式、交通费用、酒店名称、酒店地址和酒店价格等。同时规定经纬度只返回数值，游玩时间采用“第X天-时间段”的格式，费用填写具体金额或“免费”，交通方式、就餐地点和酒店信息需尽量具体。

用户提示词根据页面提交的信息动态生成，程序中实际使用的模板为：“我从{departure}出发，想去{city}旅游，出行季节是{season}，行程{days}天。{budget_desc}请为我制定一个{days}天的旅游路线，包含三餐美食/就餐地点及开销、交通方式及费用、酒店信息。”其中，`budget\_desc`用于记录用户填写的预算信息，包括总预算、路程预算、吃喝预算和酒店预算。当其中某项预算为0或未填写时，程序将该项设置为“不限资金”，再与出发地、目的地、季节和行程天数等信息一起发送给模型。

模型返回结果后，程序首先对文本进行简单清理，并查找包含“景点名称”等字段的Markdown表头，再按照“|”对各行数据进行分割并转换为DataFrame。当某一行字段数量多于表头时保留与表头对应的部分，字段不足时使用空值补齐；如果未找到有效表头或数据行，则返回异常信息。完成表格解析后，程序进一步统一字段名称，并对经纬度等字段进行数值转换，转换失败的数据置为空值，其余字段统一转换为字符串。经过上述处理后，生成的结构化行程可继续用于前端页面展示和地图路线绘制。

3.3 景点智能推荐

推荐模块首先获取当前用户全部有效行为并建

立“已操作景点”集合，再将收藏、点赞、评分、加入行程和不感兴趣等行为映射为表1中的权重。行为数据充足时构造用户—景点评分矩阵并计算余弦相似度，随后根据式(2)累计候选得分；行为不足或矩阵为空时直接转入热门景点策略。

完成相似度计算后，程序按推荐得分排序并过滤历史操作对象；个性化结果不足时用热门有效景点补齐，最终返回12条推荐结果。用户后续操作再次写入数据库，使推荐列表随行为变化更新。

3.4 用户登录、注册与权限认证

用户登录调用Django认证模块校验账号与密码，通过session维持登录状态，并根据role字段跳转普通用户或管理员页面。注册时检查用户名、电话、邮箱和密码的格式与唯一性，密码使用Django内置哈希机制存储；关键视图在执行写入操作前进行身份和角色校验，防止未授权访问。

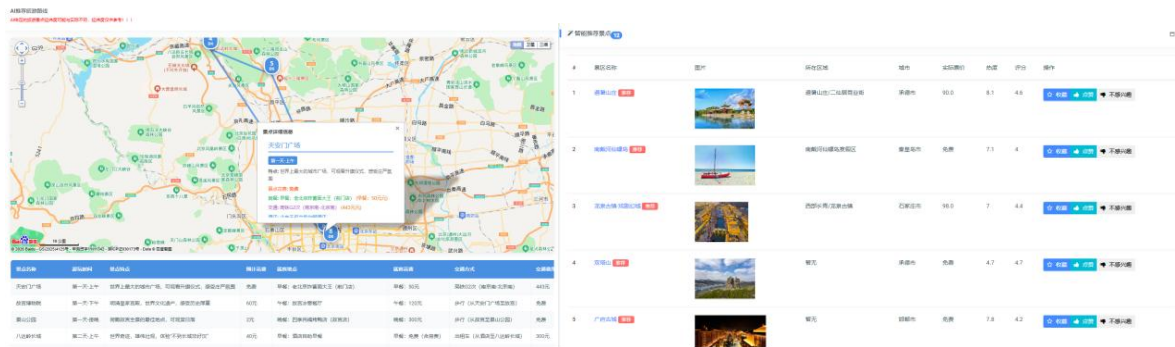


图8 AI 线路规划与个性化推荐界面

Figure 8 AI Route Planning and Personalized Recommendations Interface

4.2 景点检索与地图可视化

普通用户可按省份、城市、景点名称以及热度、评论数和评分等条件查询景点，并通过分页控制单次加载量。进入详情页后可查看评分、评论数、票价、开放时间、地址和服务设施等信息，百度地图同步显示景点位置。用户在详情页的收藏、点赞、评分、加入行程和不感兴趣等操作写入行为表，形成检索与推荐之间的反馈闭环。

4.3 餐厅、天气与AI智能问答

餐厅模块支持名称、省份、评分、评论数和人均消费筛选，并在详情页展示菜系、特色美食、地址和地图位置。天气模块根据省、市、区县或详细地址返回当前天气、未来预报及OpenWeather气象图层；AI智能体以DeepSeek为核心，为用户提供目的地介绍、景点咨询和线路建议。三类功能作为推荐结果的补充，减少用户在多个平台间重复查询。

3.5 管理端统计与可视化

管理员首页基于业务数据库生成用户新增趋势、各省景点数量及平均评分、各菜系餐厅数量及平均评分等统计结果。View层完成数据聚合并封装为JSON，前端使用ECharts绘制图表；管理端同时提供用户、景点和餐厅的查询、新增、编辑、删除和批量维护。对空数据和缺失评分进行预处理，避免前端图表组件异常。

4 系统实现与界面验证

4.1 AI 路线与推荐功能

AI线路页面根据用户输入生成按日期组织的行程，并在地图中标记推荐景点及路线；智能推荐页面根据当前用户历史行为展示12个候选景点，并过滤已收藏、点赞、评分、加入行程或标记不感兴趣的对象。两类核心界面如图8所示。

4.4 管理员子系统

管理员可维护用户、景点和餐厅信息，并查看用户创建时间、各省景点数量与平均评分、各菜系餐厅数量与平均评分等统计结果。上述统计直接来源于业务表聚合，可用于管理员掌握数据状态、发现区域数据和类别分布不均衡问题，并为后续爬虫补采、数据清洗及推荐算法样本分布分析提供依据。

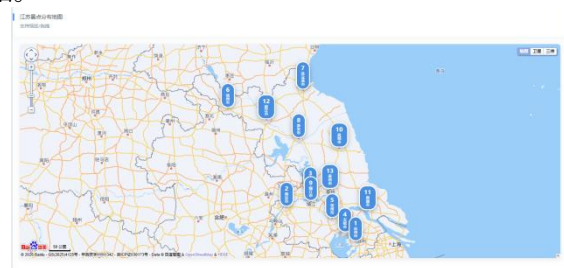


图9 省份景点地图分布

Figure 9 Provincial attraction distribution

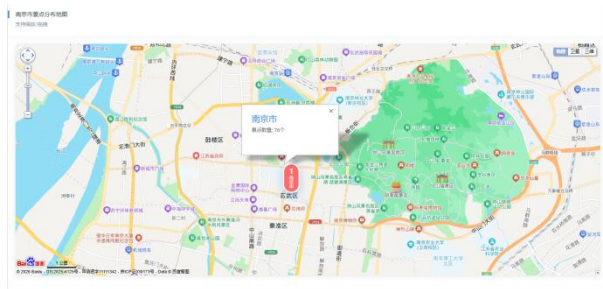


图 10 城市景点分布

Figure 10 City attraction distribution



图 11 景点详情页

Figure 11 Attraction details page

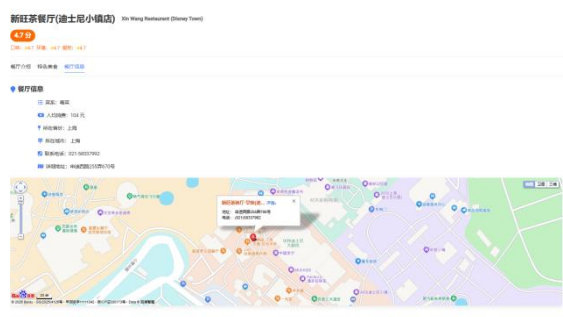


图 12 特色餐厅详情与地图

Figure 12 Restaurant details and map



图 13 天气查询界面

Figure 13 Weather query page



图 14 气象雷达地图

Figure 14 Weather radar map



图 15 AI 智能体对话
Figure 15 AI travel assistant

用户列表

按用户ID排序 按用户名排序 按创建时间排序

Q 搜索用户名/手机号... 所有角色 Y 筛选 重置

用户ID	用户名	手机号	邮箱	性别	用户角色	创建时间	操作
1	user1 普通用户	19534125340	3312118@qq.com	男	普通用户	2025-12-31 00:31	编辑 删除
2	user2 管理员	18052016535	111@qq.com	女	管理员	2026-02-27 00:20	编辑 删除
4	user3 普通用户	14512345644	11@qq.com	未知	普通用户	2026-03-04 17:26	编辑 删除
5	user4 普通用户	15612312311	2@qq.com	女	普通用户	2026-03-04 22:28	编辑 删除
6	user14 普通用户	55118510200	szeyu1207@outlook.com	女	普通用户	2002-09-06 02:06	编辑 删除
7	user87 普通用户	73301380320	mariemorris@mail.com	女	普通用户	2023-03-06 14:09	编辑 删除
8	user64 普通用户	92690670328	dennadams@mail.com	女	普通用户	2014-06-12 23:58	编辑 删除
9	user26 普通用户	09931057851	renmat@outlook.com	女	普通用户	2012-10-17 05:05	编辑 删除
10	user70 普通用户	92344456653	llnt@hotmail.com	女	普通用户	2018-06-10 13:06	编辑 删除
11	user98 普通用户	73833925675	xiuyingw@icloud.com	女	普通用户	2001-07-04 08:54	编辑 删除

1 2 下一页 末页

当前显示第 1 - 10 条, 共 14 条用户记录

图 16 用户管理界面
Figure 16 User management page

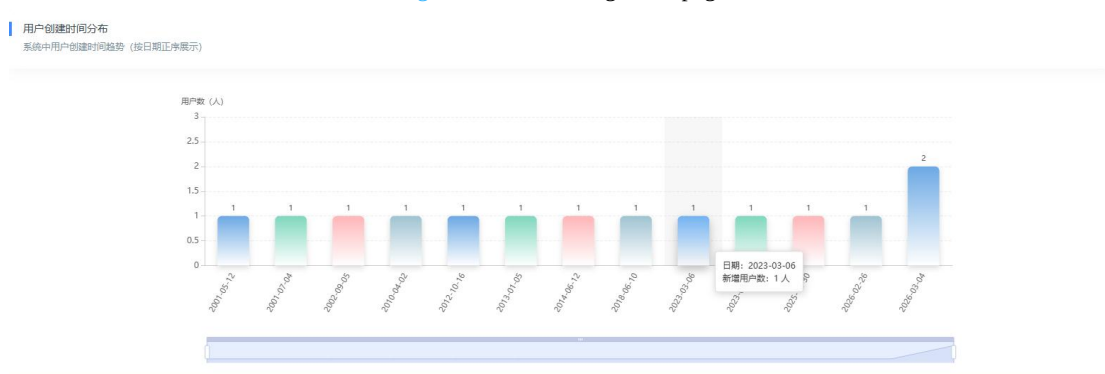


图 17 用户创建时间分布图
Figure 17 Distribution of User Creation Times

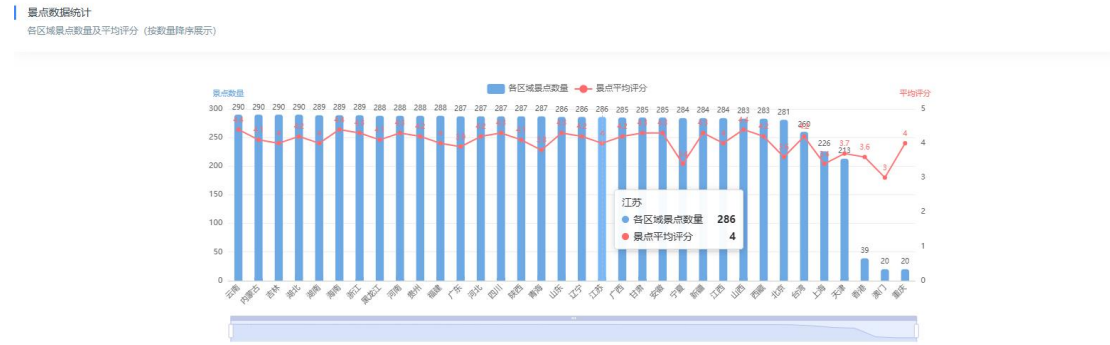


图 18 各省份景点数据统计以及平均评分图

Figure 18 Chart Showing Statistics on Tourist Attractions by Province and Average Ratings

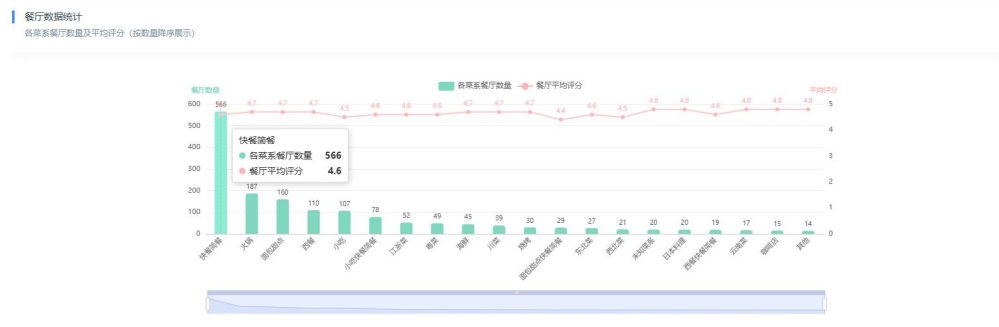


图 19 各菜系餐厅数量以及餐厅均评分图

Figure 19 Number of Restaurants by Cuisine and Average Restaurant Ratings

4.5 系统测试

系统完成后对登录、查询、地图、天气、AI、推荐、管理、并发与密码存储等功能进行测试。下表选取原系统测试中的主要用例，结果均达到预期。

测试结果表明，普通用户与管理权限能够正确区分，景点、餐厅、天气、地图、AI线路和推荐等核心功能均能返回预期结果，主要交互处于秒级

响应范围。AI线路规划响应时间不超过3 s，1000个用户并发注册可在3 min内完成，数据库未明文存储密码。

需要指出的是，当前测试主要验证工程可用性与接口响应，尚未给出Precision、Recall、NDCG等离线推荐指标，因此现有结果能够证明系统功能闭环和推荐逻辑可运行，但不能据此与其他推荐算法进行严格的精度比较。

表 9 系统主要功能与性能测试结果

Table 9 Test Results for the System's Key Functions and Performance

序号	类型	测试用例	测试方法	预期结果	实际结果	状态
1	功能	用户登录	分别输入管理员、普通用户的账户、密码	登录成功，响应时间小于等于1s	系统中的用户成功跳转对应主界面，反之提示用户不存在，响应时间小于等于0.5s	正常
2	功能	景点/餐厅查询	进入普通用户子系统，输入查询信息，进行精确和模糊查找	正确查找景点/餐厅信息，响应时间小于等于1s	成功返回查询列表，正确找到景点/餐厅信息，响应时间小于等于1s	正常
3	功能	天气查询	输入省份、城市、区县，进行天气查询	正确返回天气信息，加载气象雷达地图，响应时间小于等于1s	成功显示百度地图返回的气候信息，成功加载OpenWeather气象雷达地图，响应时间小于等于1s	正常
4	功能	智能体对话	输入景点推荐线路等，进行交互	正确返回回答，响应时间小于等于2s	输入系统存在的城市成功返回结果，不存在的城市提示系统暂无信息，响应时间小于等于1s	正常

续表9

序号	类型	测试用例	测试方法	预期结果	实际结果	状态
5	功能	AI线路推荐	输入目的地、预算等信息，进行行程规划	成功返回行程。响应时间小于等于5s	输入系统存在城市成功返回结果，不存在城市提示系统，前端和终端成功返回行程信息，在百度地图上可视化显示，响应时间小于等于3s	正常
6	功能	景点行为操作	对景点进行收藏、点赞等操作	正确写入数据库，并在收藏界面显示	成功写入数据库，响应时间小于等于1s	正常
7	功能	智能景点推荐	基于操作为用户推荐景点	正确根据用户操作推荐，避免已操作景点的推荐，响应时间小于等于3s	成功进行推荐，避免了已操作景点的推荐，同时终端显示 ItemCF 得分，响应时间小于等于2s	正常
8	性能	并发注册	模拟1000个用户同时进行注册	系统能在5min 内完成	3min 内完成，响应时间小于等于3s	正常
9	安全	密码加密存储	注册用户后，查询数据库	数据库中密码加密存储	密码加密存储	正常
10	安全	权限越界访问	普通用户访问管理员界面	无法访问	拒绝访问，并提示权限不足	正常

5 结论

本文围绕旅游信息筛选与个性化行程规划需求，完成基于Python、Django和MySQL的旅游景点推荐系统。系统以8959条景点和2678条餐厅数据为基础，采用ItemCF建立景点相似关系，并将收藏、加入行程、点赞、评分、取消收藏和不感兴趣等行为映射为不同权重；通过过滤已操作景点和热门景点兜底，使推荐模块在新用户及行为稀疏场景下仍可生成候选结果。

系统同时集成DeepSeek、百度地图和OpenWeather，实现AI线路规划、智能问答、地图定位和天气查询。功能与性能测试表明，各核心模块运行正常并满足原设计响应要求。后续可在更大规模用户行为数据上补充离线评价指标，对ItemCF、矩阵分解、深度学习和混合推荐进行统一比较，并进一步考虑时间、距离、预算和天气等上下文因素。

参考文献

- [1] 王倩. 个性化旅游景区推荐算法研究[D]. 贵阳: 贵州民族大学, 2025.
- [2] WERNECK H, SILVA N, VIANA M C, et al. A survey on point-of-interest recommendation in location-based social networks[C]//Proceedings of the Brazilian Symposium on Multimedia and the Web. São Luís, Brazil: ACM, 2020: 185-192.
- [3] 李志豪. 基于深度学习的个性推荐算法研究及应用[D]. 贵阳: 贵州大学, 2023.
- [4] 徐中彦. 融合深度学习与知识图谱的个性化推荐算法[D]. 无锡: 江南大学, 2023.
- [5] 姚南, 王宏生. 基于图神经网络的推荐算法研究[J]. 长江信息通信, 2021, 34(10): 41-43.
- [6] 于翹楚, 赵明清, 罗雨婷. 基于最优权的协同过滤混合推荐算法及应用[J]. 运筹与管理, 2024, 33(7): 79-84.
- [7] SARWAR B, KARYPIS G, KONSTAN J A, et al. Item-based collaborative filtering recommendation algorithms[C]//Proceedings of the 10th International Conference on World Wide Web. Hong Kong: ACM, 2001: 285-295.
- [8] KOREN Y, BELL R, VOLINSKY C. Matrix factorization techniques for recommender systems[J]. Computer, 2009, 42(8): 30-37.
- [9] HE X, LIAO L, ZHANG H, et al. Neural collaborative filtering[C]//Proceedings of the 26th International Conference on World Wide Web. Perth, Australia: ACM, 2017: 173-182.
- [10] KANG W C, MCAULEY J. Self-attentive sequential recommendation[C]//2018 IEEE International Conference on Data Mining. Singapore: IEEE, 2018: 197-206.
- [11] SUN F, LIU J, WU J, et al. BERT4Rec: Sequential recommendation with bidirectional encoder representations from Transformer[C]//Proceedings of the 28th ACM International Conference on Information and Knowledge Management. Beijing, China: ACM, 2019: 1441-1450.
- [12] WANG X, HE X, WANG M, et al. Neural graph collaborative filtering[C]//Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. Paris, France: ACM, 2019: 165-174.

- [13] HE X, DENG K, WANG X, et al. LightGCN: Simplifying and powering graph convolution network for recommendation[C]//Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. Virtual Event, China: ACM, 2020: 639-648.
- [14] WU J, WANG X, FENG F, et al. Self-supervised graph learning for recommendation[C]//Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. Virtual Event, Canada: ACM, 2021: 726-735.
- [15] WANG X, HE X, CAO Y, et al. KGAT: Knowledge graph attention network for recommendation[C]//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. Anchorage, AK, USA: ACM, 2019: 950-958.
- [16] LIM N, HOOI B, NG S K, et al. Hierarchical multi-task graph recurrent network for next POI recommendation[C]//Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. Madrid, Spain: ACM, 2022: 1133-1143.
- [17] YANG S, LIU J, ZHAO K. GETNext: Trajectory flow map enhanced Transformer for next POI recommendation[C]//Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. Madrid, Spain: ACM, 2022: 1144-1153.
- [18] DING T. Multi-source auxiliary information tourist attraction and route recommendation algorithm based on graph attention network[J]. Journal of Intelligent Systems, 2024, 33(1): 20240070.
- [19] WANG Z, HÖPKEN W, JANNACH D. A survey on point-of-interest recommendations leveraging heterogeneous data[J]. Information Technology & Tourism, 2025, 27(1): 29-73.
- [20] 姜庆玲, 张樊. 基于Python与Requests模块的网络图片爬虫程序设计[J]. 电脑编程技巧与维护, 2023(6): 59-61, 95.
- [21] 鞠慧. 热销图书爬取数据的BeautifulSoup库解析[J]. 福建电脑, 2021, 37(5): 133-134.
- [22] 李晨昊. 基于BeautifulSoup+Requests和Selenium爬虫网页自动化处理的实现和性能对比[J]. 现代信息科技, 2021, 5(16): 10-12, 18.
- [23] 中华人民共和国文化和旅游部. 2025年文化和旅游发展统计公报[EB/OL]. (2026-06-02)[2026-08-10].
- [24] 胡晓菁, 曲春歌, 石英, 等. 基于人工智能技术的金融产品推荐研究与应用[J]. 邮政研究, 2025, 41(4): 28-32.
- [25] 周洪萍, 郑雨菲. 基于协同过滤推荐算法的广播电视节目推荐系统实现[J]. 广播电视网络, 2024(S2): 25-30.
- [26] SARWAR B, KARYPIS G, KONSTAN J A, et al. Item-based collaborative filtering recommendation algorithms[C]//Proceedings of the 10th International Conference on World Wide Web. Hong Kong: ACM, 2001: 285-295.

A Python-Based Tourist Attraction Recommendation System

Wu Yutong^{1,2}

1. School of Intelligent Manufacturing, Jiangsu Normal University, Xuzhou 221116

2. Sino-Russian Institute, Jiangsu Normal University, Xuzhou 221116

Abstract: To reduce the cost of information filtering for users on travel information platforms and address the issues of inadequate personalized recommendations and inefficient itinerary planning, this paper designs and implements a travel attraction recommendation system based on Python and Django. The system uses Requests and BeautifulSoup4 to crawl data on attractions and restaurants from Ctrip, obtaining a total of 8,959 attractions and 2,678 restaurants. It employs an item-based collaborative filtering algorithm (ItemCF), integrating behavioral weights such as favorites, likes, ratings, adding to itineraries, and “not interested,” while using a popular attractions strategy to mitigate cold-start issues and a lack of candidates. Additionally, the system integrates DeepSeek, Baidu Maps, and OpenWeather to enable intelligent Q&A, itinerary planning, map display, and weather queries. Testing shows that the system’s core functions operate stably, with AI itinerary planning response times not exceeding 3 seconds, meeting the engineering requirements for small- to medium-sized tourism recommendation systems.

Keywords: Tourism attraction recommendation system; Python; Collaborative filtering; Web crawler; Large language model