旅游推荐系统SDD

32320222202743 黄晶晶

**1. 引言**

**1.1 项目背景**

北京市旅游路线推荐系统旨在解决游客选择景点和规划旅游路线的难题，通过大数据和机器学习技术，提供个性化旅游推荐服务。本项目结合 Spark、Hadoop 和 Django 框架开发，利用景点评论数据和用户兴趣生成推荐结果。

**1.2 目标**

设计一个高效、可扩展且用户友好的旅游推荐系统，提供以下核心功能：

1. 提取景点评论的主要主题。
2. 根据用户输入的兴趣点匹配相关景点。
3. 生成最优旅游路线并可视化展示。
4. 高效存储和处理大规模数据。

**1.3 范围**

本 SDD 涵盖系统的总体结构设计、模块划分、数据流以及技术选型。

**2. 系统概述**

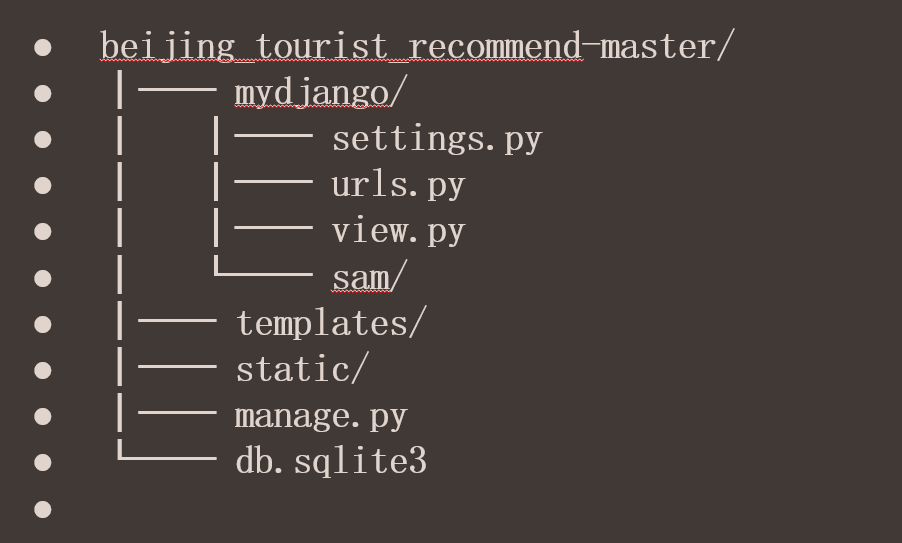
**2.1 系统架构**

系统采用分层架构设计，包含以下主要组件：

* **前端展示层**：负责用户交互和推荐结果的可视化展示。
* **后端业务逻辑层**：处理用户请求、调用算法模块并返回推荐结果。
* **数据存储与计算层**：用于数据存储、预处理以及模型训练。

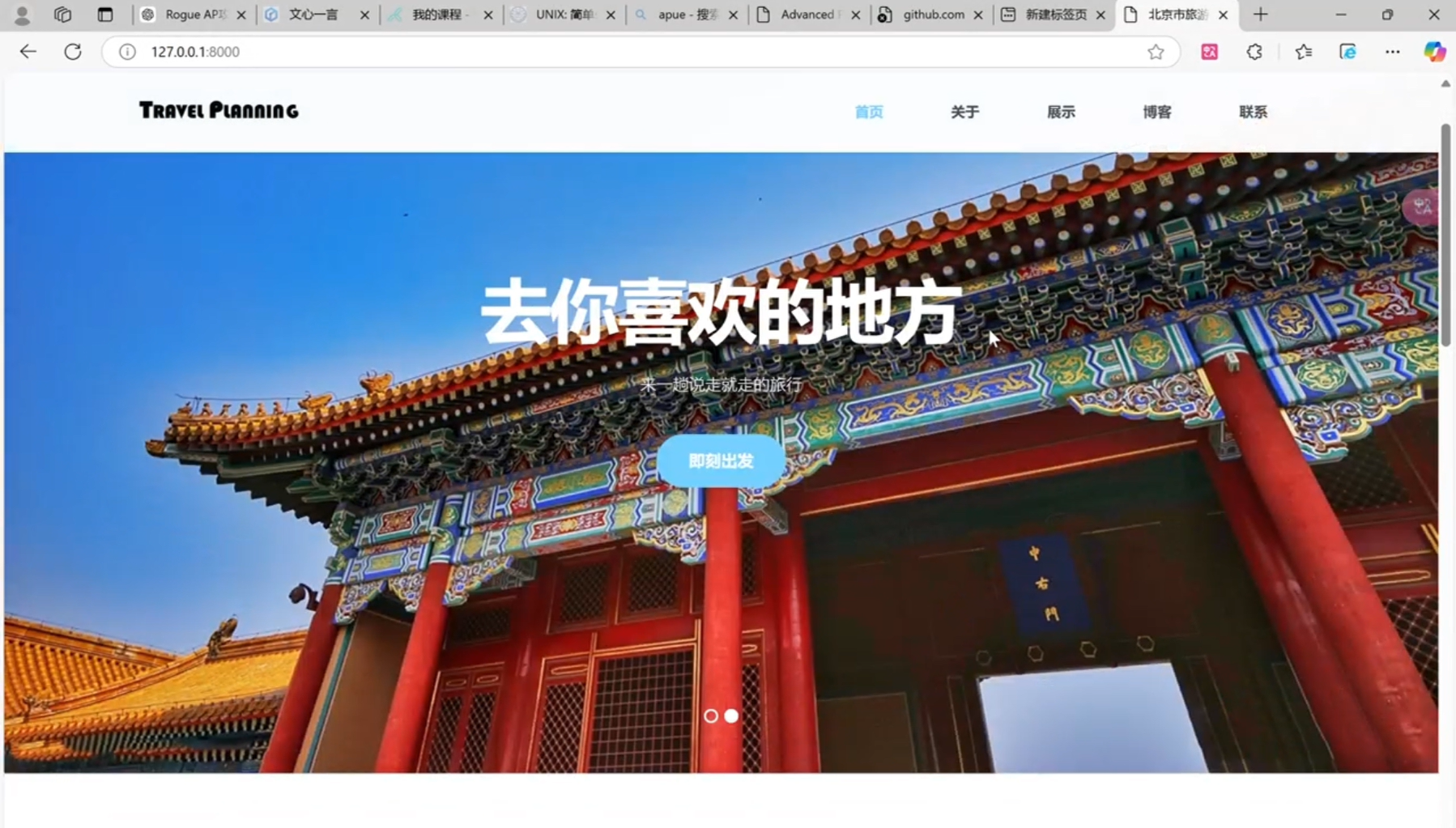
架构设计包括以下主要功能模块：

1. 用户交互模块：基于 Web 界面，提供输入接口和结果展示。
2. 数据存储模块：使用 HDFS 管理大规模评论数据，支持快速存取。
3. 数据处理模块：结合 Spark 和 Hadoop，完成数据清洗、转换和主题建模。
4. 推荐算法模块：根据用户兴趣主题生成景点和路线推荐。



**2.2 功能概述**

1. 用户输入兴趣点。
2. 基于评论数据分析用户兴趣主题。
3. 调用推荐算法生成相关景点及旅游路线。
4. 提供景点详细信息和导航功能。

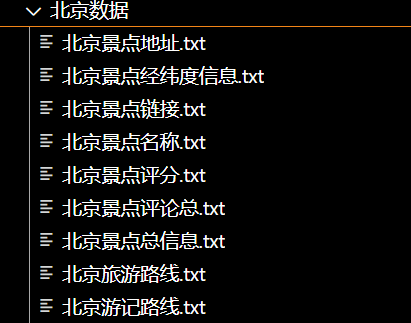


**3. 模块设计**

**3.1 数据预处理模块**

**功能描述**

1. 爬取并清洗携程景点评论数据。
2. 转化数据格式为 LDA 模型训练所需的输入。
3. 上传清洗后的数据至 HDFS。



**输入/输出**

* 输入：景点评论文本文件。
* 输出：清洗后的评论数据文件 (HDFS 存储)。

**技术选型**

* **语言**：Python。
* **依赖包**：pandas、nltk。
* **存储**：Hadoop HDFS。

**具体设计**

1. 数据清洗步骤：
   * 去除特殊字符、HTML 标签和停用词。
   * 分词处理，确保模型输入的语料库标准化。
2. 数据转换：
   * 转换为 JSON 格式，包含评论、景点 ID 和时间戳等信息。
3. 数据上传：
   * 使用 HDFS API 上传至分布式存储环境。

**3.2 LDA 模型分析模块**

**功能描述**

1. 基于评论数据进行主题建模。
2. 输出每个景点的主题分布及关键字。

**输入/输出**

* 输入：清洗后的评论数据。
* 输出：景点主题分布 (JSON 文件)。

**技术选型**

* **平台**：Spark。
* **算法**：LDA (Latent Dirichlet Allocation)。
* **依赖包**：pyspark、numpy、scikit-learn。

**具体设计**

1. 模型训练：
   * 使用 Spark 的 MLlib 模块进行并行化训练。
   * 设置主题数量、最大迭代次数和稀疏性参数。
2. 主题提取：
   * 提取每个主题的关键字及权重。
3. 结果存储：
   * 将训练结果存储为 JSON 文件，供推荐模块使用。

**3.3 推荐模块**

**功能描述**

1. 根据用户兴趣向量匹配景点主题分布。
2. 生成推荐景点列表和最优旅游路线。

**输入/输出**

* 输入：用户兴趣向量、景点主题分布。
* 输出：推荐景点列表及路线。

**技术选型**

* **语言**：Python。
* **推荐算法**：内容推荐、路径优化算法。

**具体设计**

1. 用户兴趣分析：
   * 使用向量化表示用户输入的兴趣点。
   * 通过余弦相似度计算与景点主题的匹配度。
2. 路线优化：
   * 使用 Dijkstra 算法计算最短路径。
   * 支持用户选择驾车、步行或公共交通策略。
3. 输出推荐结果：
   * 包含景点名称、主题关键词、评分和路线距离。

**3.4 Web 服务模块**

**功能描述**

1. 提供用户交互界面。
2. 接收用户请求并返回推荐结果。
3. 可视化展示推荐景点及路线。

**输入/输出**

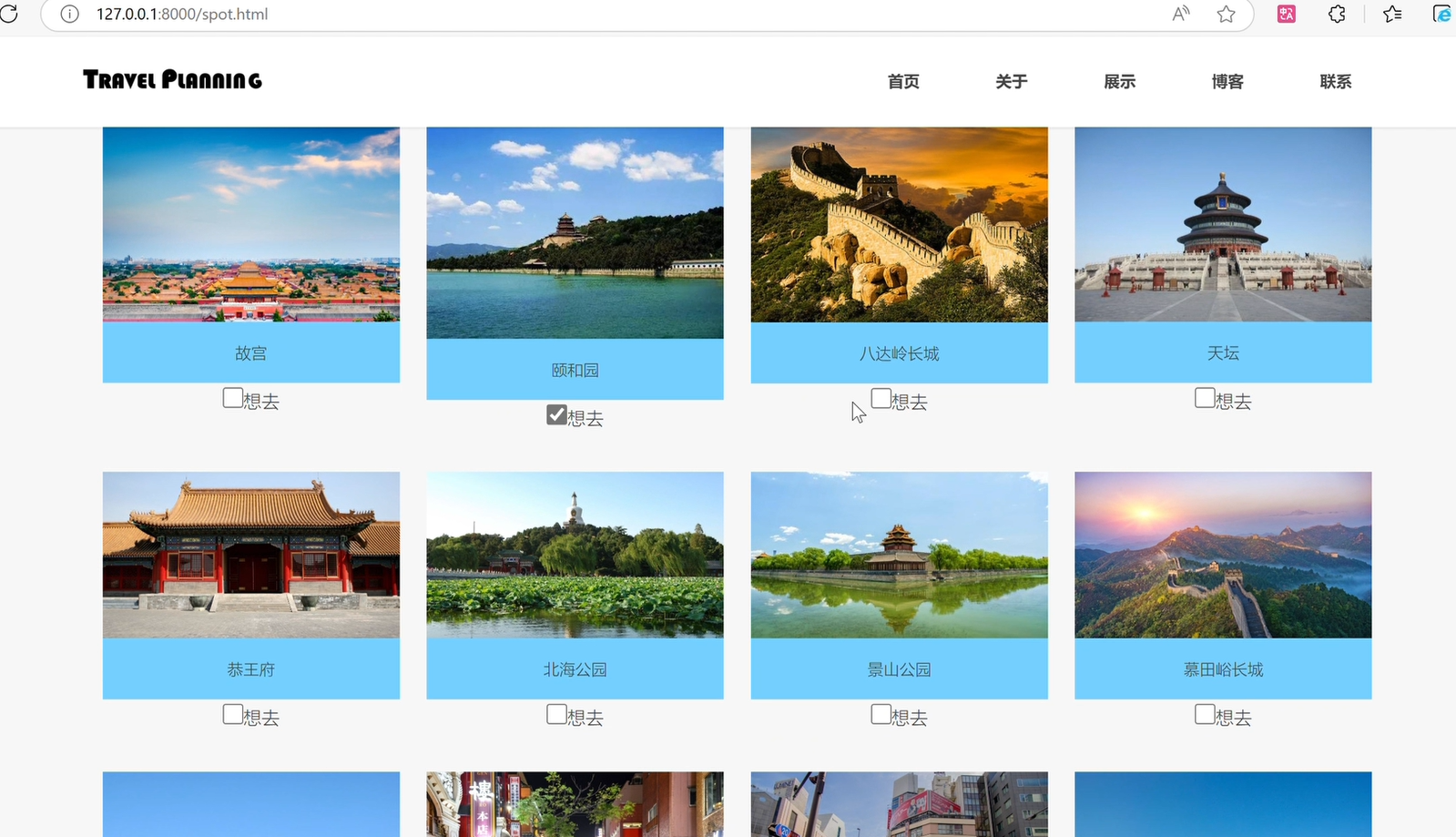
* 输入：用户兴趣点。
* 输出：推荐景点和路线的可视化结果。

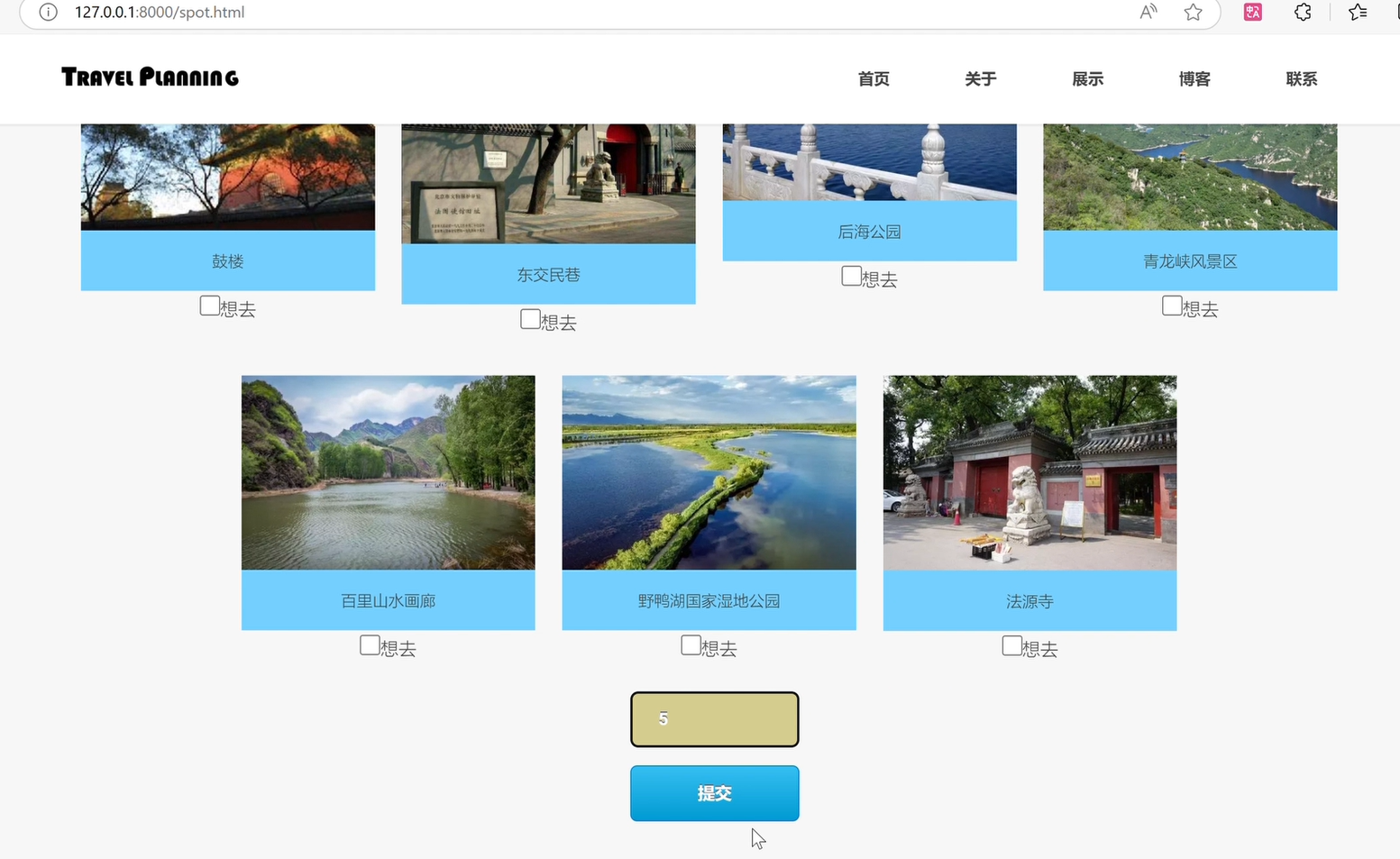
**技术选型**

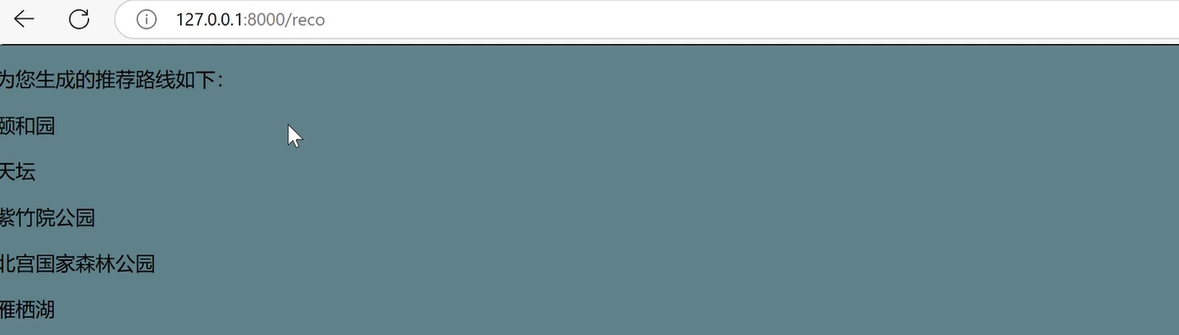
* **框架**：Django。
* **前端技术**：HTML、CSS、JavaScript。
* **数据库**：SQLite。

**具体设计**

1. 前端界面设计：
   * 使用 Bootstrap 提高界面响应速度。
   * 提供地图显示推荐路线功能。
2. 后端逻辑：
   * 编写 Django View 函数处理用户请求。
   * 使用 REST API 提供推荐结果数据接口。
3. 数据存储：
   * 记录用户查询历史，支持后续分析。







**4. 数据设计**

**4.1 数据存储**

1. **景点评论数据**：HDFS 存储。
2. **景点主题分布**：JSON 文件。
3. **用户请求记录**：SQLite。

**4.2 数据模型**

**表结构设计**

1. 景点表：
   * 字段：景点 ID、名称、地址、经纬度、评分、评论数量。
2. 用户请求表：
   * 字段：请求 ID、用户 ID、兴趣点输入、时间戳、推荐结果。

**5. 技术选型**

**5.1 开发工具**

* 操作系统：Ubuntu 22.04。
* 开发语言：Python 3.8。
* 集成开发环境：PyCharm。

**5.2 依赖库**

* 前端：HTML、CSS、JavaScript。
* 后端：Django 2.1.2。
* 数据分析：pandas、numpy、scikit-learn。
* 分布式计算：Spark、Hadoop。

**6. 性能要求**

**6.1 系统性能**

1. **高效性**：推荐结果生成时间小于 2 秒。
2. **可扩展性**：支持 10 万条以上评论数据的处理。

**6.2 用户体验**

1. 界面友好，操作便捷。
2. 推荐结果与用户兴趣高度匹配。

**7. 总结**

北京市旅游路线推荐系统结合了大数据技术和机器学习算法，在性能和用户体验方面有显著优势。通过进一步优化推荐算法和界面设计，可进一步提升系统价值和用户满意度。