CSC111 Project Report: Restaurant Recommendation System

Qingyi Jiang, Songxuan Wu, Rachel Yeung, Jiaqi Zhao

Winter 2025

Introduction

Moving to a new living environment can be exciting but overwhelming. A common and big challenge for newcomers is finding dining options that suit their preferences among a large number of nearby restaurants. While online platforms like Yelp and Google Reviews provide recommendations based on ratings and popularity, they often fail to consider personalized preferences. Moreover, these platforms present restaurants in long lists, making it harder to compare multiple options and select the best choice.

This issue is especially relevant in urban environments, where the increasing diversity and number of dining options create a growing demand for personalized recommendation systems. By implementing a tree-based recommendation system, we aim to develop a more user-friendly, structured, and effective method for individuals to explore restaurants in a new city. Our project applies key computer science concepts while addressing a real-world problem that affects many individuals moving to unfamiliar locations.

The goal of this project is to develop a smart restaurant recommendation system using tree-based modeling based on cuisine type, price range, and location. This will allow newcomers to efficiently find restaurants by navigating through a structured tree of options.

Motivation

Being an international student studying in Toronto, our group has personally experienced the challenge of finding good dining options in an unfamiliar city. Due to tight academic schedules, we often have limited time to search for suitable restaurants that fit our preferences and budgets. Many existing platforms do not effectively cater to newcomers who lack knowledge of local dining options, leading to frustration and inefficient decision-making.

Background Knowledge and Context

The problem of restaurant selection is closely related to hierarchical classification, which can be effectively solved using tree structures in computer science. A tree data structure efficiently represents hierarchical relationships, such as categorizing restaurants by cuisine type, price range, and location.

In our system, the root node represents all available restaurants, while intermediate nodes correspond to different classification levels, such as cuisine type or price range. The leaf nodes contain individual restaurants, each with relevant details such as name, address, rating, and customer reviews. This hierarchical organization enables efficient searching and filtering, allowing users to quickly locate restaurants that match their preferences.

Dataset

We use the Zomato Bangalore Restaurants dataset from Kaggle. This dataset provides information on various restaurants in Bangalore, India, including restaurant names, cuisines, locations, ratings, prices, and votes. It serves as an excellent resource for our project as it provides a broad range of features relevant to the filtering and ranking process.

Computational Overview

The recommendation system is built on a decision tree structure combined with graph-based ranking techniques. It allows two-step processing when restaurants are first filtered based on user preferences and then ranked based on

their similarity to other restaurants from the dataset. Once the user selects their preferences, the system presents a list of restaurants that fit the criteria.

Data Representation

Our system makes use of a decision tree to represent restaurant filtering. The tree structure allows top-down classification starting from cuisine type, followed by price range and rating. For the graph-based ranking model, we construct a similarity graph and apply the PageRank algorithm. Restaurants are nodes, and edges are created based on similarity (cuisine, rating, price). The graph serves as the foundation for PageRank, ranking restaurants based on connectivity and relevance.

Major Computations

- **Data Preprocessing**: Handle missing values, standardize categorical fields, and normalize numerical attributes (e.g., ratings, price).
- **Decision Tree Filtering**: Restaurants are filtered in a top-down manner according to user preferences.
- **Graph Construction and PageRank Ranking**: A similarity graph is constructed, and PageRank is applied to rank filtered restaurants.
- **Visualization**: The ranked list of top 10 restaurants is visualized using a horizontal bar chart generated by Plotly.

Visual Output

- **Ranked Restaurant List**: Displays the top 10 restaurants based on PageRank.
- **Bar Chart**: A simple bar chart presents the ranking of the top 10 restaurants.
- **Network Graph**: A graph visualizes the top 10 restaurants as nodes, with colors representing locations and edges indicating shared areas, revealing spatial clusters.

Python Libraries

- **networkx**: Used to construct the restaurant graph and perform graph-based computations such as PageRank. Key functions include `networkx.Graph()` and `networkx.pagerank()`.
- **numpy**: Used for efficient numerical operations and array manipulation throughout the data processing pipeline.
- **scipy**: Used for scientific computing tasks, such as distance calculations or sparse matrix handling, where applicable.
- **plotly**: Used to generate static bar chart visualizations of top-ranked restaurants. Functions include `plotly.express.bar()` and `plotly.graph_objects.Figure.update_layout()`.
- **matplotlib**: Used to create the network graph visualization, including node coloring, layout, and legend support. Key functions include `matplotlib.pyplot.figure()` and `networkx.draw()` with matplotlib as the backend.
- **python-ta**: Used for automated testing and code quality checking.

Instructions for Running the Program

- **Required Libraries**: Install all required libraries using:

```
pip install networkx numpy scipy plotly python-ta matplotlib
```

- **Dataset**: Included as `zomato_cleaned.csv` in the project folder. No external download is required.
- **Running the Program**: Run `main.py` in the terminal:

  ```
  python main.py
  ```

  Follow console prompts to enter user preferences (e.g., cuisine type, budget, rating).
- **Output**: A web browser opens to display two visualizations — a bar chart showing the top 10 restaurants by PageRank, and a network graph illustrating location-based relationships among them.

Changes from Proposal

The overall structure and methodology of the project remain consistent with the original proposal, focusing on tree-based filtering and graph-based ranking. However, the Minimum Spanning Tree (MST) optimization, initially mentioned as an optional step, was not implemented. The final implementation focuses on the decision tree and PageRank components, which effectively meet the project goals.

Discussion

Achievement of Project Goals

Our project set out to develop a personalized restaurant recommendation system that supports users—particularly those new to a city—in finding dining options suited to their preferences. By integrating decision tree filtering and graph-based ranking using the PageRank algorithm, we created a system that is both logical in structure and efficient in functionality. Based on our testing and feedback from group members, we believe that our system effectively meets the goal of offering intelligent, preference-based restaurant recommendations.

Effectiveness of Algorithms

The decision tree component proved to be a natural fit for modeling user filtering behavior. Since most users intuitively prioritize restaurant choices by considering factors such as cuisine, price range, and quality (as reflected in ratings), the tree structure allowed us to simulate this stepwise narrowing process. Each level of the decision tree represents a different dimension of filtering, helping users navigate through the dataset in a way that mirrors real-life decision-making.

Following this, we introduced graph-based ranking using PageRank, a well-known algorithm originally developed for ranking webpages. In our system, we manually constructed a graph where each node represents a restaurant and each edge represents a similarity relationship, defined by shared cuisine, close pricing, and comparable ratings. This method allowed us to capture more nuanced connections between restaurants than simple sorting by average rating.

User Experience and Output Visualization

The system outputs results through two main visualizations. First, a bar chart displays the top 10 recommended restaurants ranked by their PageRank scores, providing a clear summary of the ranking results. Second, a network graph shows the same top 10 restaurants as nodes labeled with their names and color-coded by location. Edges connect restaurants that share the same location, revealing clusters within the network. A spring layout helps distribute the nodes for clarity, and a legend maps colors to specific locations. Together, these visualizations offer a concise and informative overview of the recommendations.

Limitations and Challenges

One of the key limitations we encountered was the absence of precise geographic data. Although the dataset includes neighborhood names, it lacks latitude and longitude coordinates, which prevented us from implementing KD-Tree–based proximity filtering. Fetching external coordinates via APIs was considered but ruled out due to time constraints and project scope.

Additionally, although we had initially planned to implement a Minimum Spanning Tree (MST)–based refinement step to improve recommendation coherence, we were unable to complete this due to workload and time limitations during the semester.

Another constraint was the dataset's lack of user history, time-based data, or natural language reviews. These features are essential for more advanced forms of personalization such as collaborative filtering or sentiment analysis, and their absence limited the depth of our recommendation logic.

Future Work

Future iterations of this project could include geographic filtering based on coordinate data, allowing the system to recommend nearby restaurants in addition to those that match user preferences. Implementing the MST refinement step would also enhance the quality of the top-10 list by ensuring that the final recommendations form a coherent group.

Other possible directions include experimenting with weighted similarity scoring, learning edge weights dynamically, or integrating user-generated data to support collaborative recommendations. A full web interface using Streamlit or Flask could improve accessibility and usability, making the tool more appealing for general users.

Summary

In conclusion, our project demonstrates how fundamental concepts in computer science—such as trees, graphs, and ranking algorithms—can be used to build a meaningful and practical application. Despite a few limitations and unimplemented features, our system meets the original goal of helping users efficiently discover restaurants tailored to their stated preferences.

References

- Zomato Bangalore Restaurants Dataset – Kaggle: `https://www.kaggle.com/datasets/himanshupoddar/zomato-bangalore-restaurants`
- NetworkX Documentation: `https://networkx.org/documentation/stable/`
- Plotly Documentation: `https://plotly.com/python/`
- Matplotlib Documentation: `https://matplotlib.org/stable/contents.html`
- NumPy Documentation: `https://numpy.org/doc/stable/`