

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

- **Filtered Restaurant List:** Displays restaurants that match the user's selected criteria.
- **Ranked Restaurant List:** Displays the top 10 restaurants based on PageRank.
- **Interactive Features:** Users can view an interactive bar chart generated using Plotly.

Python Libraries

- **networkx:** Used to construct the graph of restaurants and perform graph-based computations such as PageRank. Key functions include `networkx.Graph()`, `networkx.pagerank()`, and optionally `networkx.minimum_spanning_tree()`.
- **plotly:** Used to create interactive visualizations, such as bar charts for restaurant rankings. Relevant functions include `plotly.express.bar()` and `plotly.graph_objects.Figure.update_layout()`.

Instructions for Running the Program

- **Required Libraries:** Install `networkx` and `plotly` using:

```
pip install networkx plotly
```

- **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 an interactive bar chart showing the top 10 ranked restaurants based on PageRank.

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

The system successfully answers the research question by providing an efficient and personalized restaurant recommendation process. The decision tree allows structured filtering, while the PageRank algorithm ensures meaningful rankings based on similarity to other restaurants. Together, they form a coherent recommendation system useful for newcomers in unfamiliar environments.

Limitations

- The decision tree may exclude some potentially relevant restaurants that do not meet all the specified criteria (e.g., cuisine type, price, or rating).
- The lack of MST optimization means that recommendations are not adjusted for coherence between restaurants, though the PageRank algorithm compensates by ranking restaurants based on their relative popularity.

Exploration

- The recommendation system could be enhanced by incorporating additional factors, such as user reviews, proximity to the user, or even time-based recommendations (e.g., suggestions for breakfast, lunch, or dinner).
- Other ranking algorithms, such as collaborative filtering or content-based filtering, could be explored to further refine the recommendations.

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/>