CSC111 Project Proposal: Restaurant Recommendation System

Qingyi Jiang, Songxuan Wu, Rachel Yeung, Jiaqi Zhao

Tuesday, March 4, 2025

Problem Description and Research Question

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. As cities expand, newcomers require tools that streamline the decision-making process. 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.

1. Motivation

Being an international student studying in Canada, 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. Most of the current platforms fail to adequately serve newcomers who are unfamiliar with local dining options, resulting in frustration and poor decision-making.

2. Background Knowledge and Context

The challenge of restaurant selection is closely related to hierarchical classification, which can be 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.

Computational Plan

1. Use of Trees and Graphs

1.1: Decision Tree for Filtering

- A decision tree is used to filter restaurants based on user preferences, such as cuisine type, budget range, and minimum rating.
- The tree structure follows a hierarchical filtering mechanism:
  - **Root Node**: Represents all available restaurants.
  - **First Level**: Categorization based on cuisine type.

  - **Second Level**: Further categorization based on budget.
  - **Third Level**: Final filtering based on minimum rating.
  - **Leaf Nodes**: Restaurants that meet all criteria.

- This approach allows efficient filtering before applying ranking algorithms.

1.2: Graph-Based Ranking with PageRank

- A graph is constructed where:
  - **Nodes** represent individual restaurants.
  - **Edges** connect restaurants with similar attributes (cuisine, ratings, and pricing).
  - **Edge Weights** are computed based on similarity scores derived from restaurant attributes.

- We apply PageRank to determine the relative importance of each restaurant in the network, ensuring that more "popular" restaurants (based on customer reviews and connectivity) are ranked higher.

1.3: Minimum Spanning Tree (MST) Optimization (Optional)

- As a refinement step, we use an MST-based approach to ensure that recommended restaurants maintain a degree of coherence.
- MST computation follows these steps:
  1. Construct a graph where nodes are restaurants, and edges are weighted by similarity.
  2. Compute the Minimum Spanning Tree (MST) to cluster related restaurants.
  3. Adjust PageRank results to reflect MST connectivity.
- This step ensures that recommendations are cohesive and logically structured, rather than isolated results.

2. Types of Computation

1. **Data Preprocessing**
   - Handle missing values and standardize categorical fields.
   - Convert cuisine types into numerical encodings where needed.
   - Normalize numerical attributes (ratings, votes, pricing).
2. **Decision Tree Filtering**
   - Filter restaurants based on Cuisine Type → Rating → Average Cost for Two → details of restaurants.
3. **Graph-Based Ranking with PageRank**
   - Construct a graph representation of the filtered restaurants.
   - Compute restaurant similarity using cuisine, ratings, and cost.
   - Apply PageRank to rank restaurants based on their connectivity in the similarity graph.
4. **(Optional) MST-Based Refinement**
   - Construct an MST of the restaurant similarity graph.
   - Adjust the PageRank ranking to prioritize more connected restaurants.

3. Use of Python Libraries

3.1: pandas (Data Processing)

- **Usage**: Load, clean, and manipulate restaurant data.
- **Key Functions**: pandas.read_csv(), pandas.DataFrame.query(), pandas.DataFrame.sort_values()

3.2: networkx (Graph Construction & Ranking Computation)

- **Usage**: Construct a graph representation of restaurants.
- **Key Functions**: networkx.Graph(), networkx.pagerank(), networkx.minimum_spanning_tree()

3.3: plotly (Visualization)

- **Usage**: Display restaurant rankings using bar charts.
- **Key Functions**: plotly.express.bar(), plotly.graph_objects.Figure.update_layout()

Real-World Dataset Example

The **Zomato Bangalore Restaurants** dataset provides information on various restaurants in Bangalore, India. This dataset is sourced from Kaggle and contains details such as restaurant names, cuisines, locations, ratings, prices, and other relevant attributes.

1. Sample Data

Name	Address	Cuisines	Average Cost for Two	Aggregate Rating
Truffles	Koramangala, Bangalore	American, Continental	800	4.5
Empire Restaurant	Indiranagar, Bangalore	North Indian, Biryani	600	4.2
The Black Pearl	Marathahalli, Bangalore	BBQ, Continental, Italian	1500	4.3
Meghana Foods	Jayanagar, Bangalore	Biryani, North Indian	700	4.6

Table 1: Sample Data from Zomato Bangalore Restaurants Dataset

2. Attributes in the Dataset

- **Restaurant Name:** The name of the restaurant.
- **Address/Location:** The locality where the restaurant is situated.
- **Cuisines:** The type of food served at the restaurant.
- **Average Cost for Two:** The approximate cost for two people dining at the restaurant.
- **Aggregate Rating:** The overall rating of the restaurant based on user reviews.
- **Votes:** The number of votes the restaurant has received from users.

This dataset is useful for analyzing restaurant trends, customer preferences, pricing strategies, and geographical distribution of restaurants.

3. Plan to Use the Dataset

3.1: Building a Decision Tree for Hierarchical Filtering

- **Root Node:** All available restaurants.
- **Level 1:** Classification based on cuisine type.
- **Level 2:** Further filtering based on rating.
- **Level 3:** Final filtering based on average cost for two persons.
- **Leaf Nodes:** List of restaurants that meet all filtering criteria.

3.2: Constructing a Similarity Graph

- Each restaurant is represented as a **node** in the graph.
- An **edge** is created between two restaurants if they have a similarity score based on:
  - Same cuisine type
  - Similar price range
  - Close ratings
- The **edge weight** is calculated based on these factors to determine restaurant similarity.

3.3: Applying PageRank for Restaurant Ranking

- Compute **PageRank scores** to rank restaurants based on their connectivity in the similarity graph.
- Higher PageRank scores indicate more relevant and well-connected restaurants.

4. Expected Output and Visualization

4.1: Filtered Restaurant List (Based on User Preferences) A decision tree will first filter restaurants based on:

- Cuisine type
- Minimum rating
- Average cost for two persons

The final filtered list contains only restaurants that match the user's preferences.

4.2: Ranked Restaurant List (Using PageRank) After filtering, the system ranks restaurants based on their importance in the restaurant similarity graph. The ranking will be influenced by:

- Customer reviews
- Connectivity to similar restaurants
- Overall popularity

4.3: Visual and Interactive Output

- **Basic Table Display:** Shows the recommended restaurants in order of ranking.
- **Ranking Visualization:** Bar charts using Plotly.
- **Graph Visualization (Optional):** Interactive graph representation of restaurant connections.

References

- Zomato Bangalore Restaurants dataset: Kaggle Dataset Link
- PageRank Algorithm: Brin, S., & Page, L. (1998). "The anatomy of a large-scale hypertextual Web search engine." Computer Networks and ISDN Systems.
- Decision Trees: Breiman, L. (1984). "Classification and Regression Trees."
- Minimum Spanning Tree Algorithm: Kruskal, J. B. (1956). "On the shortest spanning subtree of a graph and the traveling salesman problem." Proceedings of the American Mathematical Society.
- Data Preprocessing Techniques: Han, J., Kamber, M., & Pei, J. (2011). "Data Mining: Concepts and Techniques."

- Python Libraries:
  - pandas: McKinney, W. (2010). "Data Structures for Statistical Computing in Python." Proceedings of the 9th Python in Science Conference.
  - networkx: Hagberg, A., Schult, D., & Swart, P. (2008). "Exploring Network Structure, Dynamics, and Function using NetworkX." Proceedings of the 7th Python in Science Conference.
  - plotly: Plotly Technologies Inc. (2015). "Collaborative Data Science." Available at Plotly Official Website.