Computational Plan for Restaurant Recommendation System

Project Group

March 2025

1 Data Representation

Our project focuses on building a restaurant recommendation system that utilizes trees and graphs to structure, process, and rank restaurant data based on user preferences. The dataset we use contains real-world restaurant attributes, including:

- **General Information**: Restaurant name, location (district-based), and type (e.g., casual dining, fast food).
- **Cuisines**
  Pricing: Types of cuisines served and approximate cost for two people.
- **Ratings**
  Popularity: Customer ratings, number of reviews (votes), and overall popularity.

This dataset provides structured information that can be effectively modeled using **trees for hierarchical filtering** and **graphs for ranking and optimization** in the recommendation process.

2 Data Modeling Using Trees and Graphs

To structure our data and process recommendations efficiently, we employ a **three-stage computational pipeline**:

2.1 Decision Tree (Filtering Based on User Preferences)

- A **decision tree** is used to filter restaurants based on user preferences, such as cuisine type, budget range, and minimum rating.
- The tree structure:
  - **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 for efficient filtering before ranking.

2.2 PageRank (Graph-Based Ranking)

- We construct a **restaurant similarity graph**:
  - **Nodes** represent restaurants.
  - **Edges** connect restaurants with similar cuisine, ratings, and price range.
  - **Edge weights** are computed based on similarity scores derived from restaurant attributes.
- PageRank is applied 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.

2.3 Minimum Spanning Tree (MST) Optimization (Optional)

- As an optional refinement step, we use an **MST-based approach** to ensure that recommended restaurants maintain a degree of coherence.
- The MST is computed as follows:
  - A graph is built using **restaurants as nodes** and **edges weighted by similarity** (based on cuisine, rating, and cost).
  - A **Minimum Spanning Tree (MST)** is computed to ensure that closely related restaurants are recommended together.
  - If necessary, adjustments to the PageRank results are made to reflect MST connectivity.
- This step ensures that the recommendation list does not contain unrelated or isolated results, making recommendations more natural.

3 Data Sources

We use a structured dataset containing restaurant information, including customer ratings, cuisine types, and price categories. A sample of the data is shown in Table 1.

Name	Cuisines	Rating	Cost	Votes	Location
A	Indian	4.5	500	150	Downtown
B	Italian	4.2	700	120	Midtown
C	Chinese	4.3	1000	200	Uptown

Table 1: Sample dataset used for recommendation system.

4 Computational Steps

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 → Budget → Minimum Rating**.
3. **Graph-Based Ranking with PageRank**
   - Construct a **graph representation** of the filtered restaurants.
   - Compute **restaurant similarity** based on 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**.

5 Visualization and User Interaction

Our program will display the recommended restaurants in a clear and interactive way:

- **Table Display (Basic Interface)**: Display the recommended restaurants in a table sorted by PageRank score.
- **Ranking Visualization (Optional)**: Use **bar charts (Plotly)** to visualize restaurant rankings based on PageRank scores.
- **Interactive Filtering (Optional)**: Allow users to adjust preferences dynamically using **Streamlit**.
- **Graph Visualization (Optional)**: Provide an interactive **graph visualization of restaurant connections**.

6 Conclusion

Our project leverages **trees (Decision Tree), graphs (PageRank), and optionally MST** to create a structured and efficient restaurant recommendation system. The system follows a **logical pipeline: hierarchical filtering → ranking → optional refinement**, ensuring that recommendations are relevant, well-ranked, and cohesive.

While MST is an optional refinement step, we include it as a **possible enhancement** depending on time constraints. This modular design allows for **gradual improvements without affecting core functionality**.