CS410 Project Proposal
Team - NLP Ninjas

Team:

Rohit Pandey (Captain) (NetID - rp23)
Prasad Gole (NetID - gole2)
Asmita Chihnara (NetID - asmitac2)
Avinash Pathak (NetID - apathak3)
Surya Sindhu Mallimanugula (NetID - ssm13)

Project Details:

Topic:

MapMingle - Enhancing User Experience through Dynamic Nearby Place Recommendations

Detailed Description:

The MapMingle project is aimed at developing a dynamic and user-friendly nearby places recommender system that provides personalized suggestions based on a user specified latitude, longitude, and user-generated queries. The system leverages various data sources and sophisticated algorithms to sort and rank points of interest (POIs) using reviews and ratings, thereby ensuring that the recommendations are of high quality and relevance.

Task:

Our primary task is to design and implement a system that effectively filters, sorts, and presents nearby locations in an intuitive and user-friendly manner. Our focus lies on enhancing the user experience and building a reliable recommendation system.

Importance:

The project holds significant importance as it directly impacts user engagement and satisfaction, particularly for tourists, travelers, and locals exploring their surroundings. By offering tailored suggestions, users can discover new and exciting places, ultimately contributing to higher user satisfaction and potentially driving foot traffic to local businesses.

Planned Approach:

- **Data Filtering:** We will implement advanced algorithms to filter locations based on user-specified criteria such as radius and user-generated queries. This ensures that users receive relevant recommendations within their preferred distance.
- **Ranking Algorithm:** We will develop a sophisticated ranking algorithm that considers various metadata, including text-based content and reviews, to sort and prioritize locations. This will ensure that the most relevant suggestions are presented to users.
- **User Interface:** Our approach includes designing an intuitive and user-friendly UI/UX that allows users to effortlessly input their geographical coordinates and user-generated queries. The user interface will be responsive and easy to navigate.

Tools, Systems, or Datasets:

- **Programming Language:** Our choice of programming languages includes Python for back-end development and JavaScript for specific front-end functionalities.
- **Frontend Technologies:** For the front-end, we will employ modern technologies like React or Angular to create a responsive and interactive user interface.
- **Acquiring Datasets:** We will utilize external data sources such as Kaggle, Google Maps and Foursquare to form datasets for locations.
- **Database:** To store user data, location information, and more, we will consider both SQL and NoSQL databases based on project requirements.

Expected Outcome:

The expected outcome of the MapMingle project is a fully functional, intuitive, and responsive recommender system. This system will empower users to discover highly rated nearby places based on their specified location and queries. It will efficiently filter and sort locations to ensure that the recommendations are not only relevant but of high quality.

Evaluation:

To evaluate the effectiveness of our work, we will compare the accuracy of the outcomes with the state of the art services like google maps. The comparison will give us insight about how accurately we are suggesting nearby places based on the rating.

List of tools and technologies:

Here is the list of tools we are using:

- Python: Python serves as the primary programming language for the project due to its versatility, ease of use, and extensive library support. It's well-suited for both back-end and front-end development, making it a convenient choice for developing the entire system.

- SQL: SQL (Structured Query Language) is used for database management. It enables efficient data storage and retrieval, making it crucial for managing user data, location information, and other related data.
- Pandas: Pandas is a Python library that provides data structures and data analysis tools. It is used for data manipulation and preprocessing, particularly when handling datasets retrieved from external sources like APIs or databases.
- NumPy: NumPy is essential for scientific and numerical computing. It is used for efficient array and matrix operations, which can be particularly useful for spatial calculations and filtering of nearby locations.
- Matplotlib and Seaborn: Matplotlib and Seaborn are Python libraries for data visualization. They are used to create interactive and informative graphs and charts for presenting location data, statistics, and user recommendations in a visually appealing manner.
- Folium: Folium is a Python library for creating interactive maps. It is used to display recommended locations on a map, providing a visual representation of nearby places to users.
- scikit-learn (sklearn): scikit-learn is a machine learning library for Python. In the context of the project, it can be used for various tasks such as clustering locations, enhancing recommendation algorithms, and providing personalized suggestions.
- NLTK (Natural Language Toolkit): NLTK is employed for natural language processing tasks, particularly when dealing with user reviews and text-based data. It helps in extracting insights from user feedback, which can influence location recommendations.

WorkLoad:

We are planning to complete this in around **110 to 120 hours**. Here’s the detail breakdown of time allotment:

1. Ideation and Data Gathering:
   - Define project scope and goals (2 hours)
   - Identify data sources (6 hours)
   - Specify project features (4 hours)
   - Estimated Time: 12 hours

2. Architecture Design:
   - Define system architecture (3 hours)
   - Develop data flow diagrams (4 hours)
   - Create a database schema (3 hours)
   - Estimated Time: 10 hours

3. Data Filtering:
   - Research and select spatial filtering algorithms (5 hours)

- Implement and optimize the filtering algorithm (20 hours)
- Estimated Time: 25 hours

4. Ranking Algorithm:
   - Research and select ranking criteria (3 hours)
   - Design and implement the ranking algorithm (15 hours)
   - Estimated Time: 18 hours

5. Backend Development:
   - Develop API integration for data retrieval (7 hours)
   - Implement user management and request handling (8 hours)
   - Estimated Time: 15 hours

6. Front-end Development:
   - Front-end component development (14 hours)
   - Integration with back-end services (6 hours)
   - Estimated Time: 20 hours

7. Database Setup:
   - Database selection and setup (4 hours)
   - Data schema design and optimization (6 hours)
   - Estimated Time: 10 hours

8. Testing and Debugging:
   - Test case development (5 hours)
   - Debugging and issue resolution (10 hours)
   - Estimated Time: 15 hours

9. Documentation and Report Writing:
   - Project documentation (8 hours)
   - Academic report writing (7 hours)
   - Estimated Time: 15 hours