Background
A vast quantity of data is produced each day through interactions between users and mobile and web services. The providers of these services retain and organize large volumes of data about their users. Moreover, companies catalogue and categorize large amounts of data about the items and services they offer, such as products or ratings and reviews. It is possible to leverage these large, rich datasets to create recommender systems that point users to content tailored to their interests.

There are various approaches that have been researched and used to create recommender systems in myriad domains. One technique is the use of frequent sequences. In this approach, common sequences of ratings are used to recommend content after certain sequences of content accesses are performed by a user. Content-based algorithms for recommender systems treat recommendation generation as a search problem for similar items in the set of possibilities. Collaborative filtering (CF) requires users to express some ratings or preferences on a subset of the available content. Variations of CF provide the backbone for the current state of the art in recommendation systems.

There are two basic classes of approaches to CF techniques: item-based approaches and user-based approaches. Item-based algorithms construct a matrix row by row by computing similarities between items, without comparing items to themselves. Some approaches only retain the greatest k similarity measures for each row. User-based algorithms operate under the assumption that each user of a service belongs to a group of users that exhibit similar behaviors and preferences. User-based CF recommenders profile users to categorize them into groups, determine which items are preferred or well-liked by users in each group and assign weights to them, and generate recommendations based on the items with greatest weight for a user’s group that have still not been consumed by that user. There are some issues in scaling user-based approaches. Cluster generation to determine groups scales linearly with the number of users and cannot be precomputed because the sets of users and items are dynamic. Thus, user-based approaches prove impractical for large databases of users and items. Some CF algorithms use subsets of items and subsets of users to implement an approach that combines item-based and user-based techniques.

Metrics to evaluate recommender systems include accuracy and scalability. Accuracy is a measure of how closely the output of a recommender system matches users’ preferences. In some cases the most accurate recommendations are not the best. For example, if a travel agency recommends locations that you have already visited, it might not be beneficial to you if you are seeking novelty. A tradeoff between prediction accuracy and scalability is often inevitable as the database the recommender operates on grows. A broader literature review of methods and parameters used to evaluate recommender systems remains to be conducted for this project.

Project Description
The project will make use of a dataset released by Yelp for academic purposes for its Yelp Dataset Challenge. The dataset includes data about businesses, anonymized reviews, anonymized user information, and check-in information for Phoenix, Las Vegas, Madison, Waterloo and Edinburgh.
stored in JSON format. I will implement two classes of CF recommender systems for businesses: one set will be item-based, focusing on similarities between businesses, and the other will be user-based, focusing on similarities between users and the types of businesses they interact with on the Yelp service. Both approaches will make use of only explicit user data because of the nature of the data provided by Yelp. The data gives no window into users’ interactions with businesses and ratings on Yelp’s mobile and web application beyond the ratings, reviews, and check-ins they made.

The exact implementation details still need to be worked out. I will likely use Python to rapidly prototype multiple recommender systems. Performance should not be an issue because while the dataset is rich, it is not too large since it is confined to a few cities. There are also many useful data cleaning, data analysis, and machine learning tools including pandas, numpy, scipy, and Matplotlib implemented as modules for Python. It is possible that textual analysis of reviews will provide better recommendations, so it may be included in the recommender approaches. The Natural Language Processing Toolkit provides tools for classification, tokenization, tagging, parsing, and semantic reasoning of texts in Python scripts.

**Objectives**

- Perform preliminary exploration and analysis of Yelp data to specify the details of how the recommender systems will interact with and use the data
- Conduct a literature review to determine appropriate metrics for evaluating recommender system performance
- Determine a statistically appropriate method to divide the Yelp dataset into a training set and a test set if applicable
- Implement and iterate on two sets of CF systems to recommend businesses: those that take an item-based approach and those that take a user-based approach
  - Use user ratings, reviews, and check-ins into businesses to recommend restaurants and businesses to users
  - Potentially compare reviews through textual analysis
- Implement testing infrastructure to evaluate performance of recommender systems employing different recommender making approaches

**References**