Introduction

Movie recommendation systems are important as they introduce and explore many key data mining concepts, including data cleaning and user/item similarity. The concepts implemented in these systems can be applied in a variety of circumstances in the world of big data. The approaches used in this particular assignment included user based, item based, and random walker. User based recommendations are built from identifying a user and aggregating recommendations based on other users who share similar interests. Similarly, item based recommendations are built on selecting a given item from a data set and collecting recommendations by identifying similar items based on user reviews. Random walker is the most unique as it relies on graph based representation and traversing neighbors, i.e. connections between items and users.

Dataset Description

The dataset used in the assignment is the movielens 100k dataset. This dataset contains roughly one hundred thousand ratings from one thousand users on around one thousand seven hundred movies. The Python library Pandas came in handy as data cleaning was necessary throughout the process, whether that be setting missing values to 0, setting missing values to an average, or even removing rows from the data set at certain points.

Methodology

Similarity in each of the user and item-based implementations was built off of cosine similarity. Particularly the cosine similarity between users themselves or the items respectively. This is a good indicator of how similar or dissimilar any given users or items are from one another. The random walker, on the other hand, traversed a graph structure. This is effective for the random walker algorithm as it sorts the data in such a way that relevant paths are always available for the algorithm to traverse, given it started at a point of interest and can only travel to neighbors.

Implementation Details

Details on implementation are included within the notebook in labeled markdown segments near the code itself.

Results and Evaluation

Example outputs are included in the notebook. While there were no outliers as far as I could tell, I always ran each method for more iterations or recommendations than were default. I was more curious than anything and wanted to see what a longer recommendation list for each given user or item looked like. I also wanted to see a more robust list constructed by the random walker.

Conclusion

The key takeaway that I got from the assignment was how many operations could be performed on a data set from such simple implementations of a system. I had a much more complicated idea of how each of these worked in my head, though seeing it laid out in a programming environment made it much easier to compartmentalise. I think each of the models

that I implemented could easily be improved by adding more robust code that refines recommendations even more. I think the most obvious real-world applications would be movie recommendation systems for something like Netflix or even a shopping app like Amazon. Pinterest is mentioned in the other report as well.