Comparative Analysis of Recommender Algorithms using API Dataset
Presenter: Sandeep Kumar Ragila
Advisor: Dr. Carol Romanowski

Introduction
- Recommender systems are an important part of today’s e-commerce ecosystem. Applications that have information about the interaction between the user and their items, user likings, user ratings, user browsing history, item text descriptions can use these recommender systems to better engage users by showing content they might more likely be interested in.
- Some of the well-known applications of recommendation systems are Amazon, Ebay, Netflix, Youtube, Spotify, etc.

Goals
- Recommend new set of APIs to users who might be interested in
- To compare the performances of different recommender algorithms

Algorithms
Collaborative Filtering is a popular technique used to build recommendations. It uses the information of users and their interaction with the items to build the models. Using this information, it will take a user-item binary matrix to calculate the similarity between different users or items and predict new items. The algorithms that are compared:

1. Random Recommendations
2. Popular Recommendations
3. Item-Based Recommendations
4. User-Based Recommendations

Below is the process to implement the above collaborative filtering techniques:

- Generating a User-Item binary matrix
- Choosing a similarity measure
- Training the Model
- Evaluating the Model

Results
The results are evaluated using the measure Recall @ M. Where M is the number of recommendations.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Figure 1</th>
<th>Figure 2</th>
<th>Figure 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Recommendations</td>
<td><img src="source1.png" alt="Figure 1" /></td>
<td><img src="source2.png" alt="Figure 2" /></td>
<td><img src="source3.png" alt="Figure 3" /></td>
</tr>
<tr>
<td>Popular Recommendations</td>
<td><img src="source1.png" alt="Figure 1" /></td>
<td><img src="source2.png" alt="Figure 2" /></td>
<td><img src="source3.png" alt="Figure 3" /></td>
</tr>
<tr>
<td>Item-Based Recommendations</td>
<td><img src="source1.png" alt="Figure 1" /></td>
<td><img src="source2.png" alt="Figure 2" /></td>
<td><img src="source3.png" alt="Figure 3" /></td>
</tr>
<tr>
<td>User-Based Recommendations</td>
<td><img src="source1.png" alt="Figure 1" /></td>
<td><img src="source2.png" alt="Figure 2" /></td>
<td><img src="source3.png" alt="Figure 3" /></td>
</tr>
</tbody>
</table>

Discussion
- Data sparsity in the user-item binary matrix is 99.6%
- Less than 1% of user-item interaction in the dataset
- Figure 1 and Figure 2 were achieved with k value 10
- Figure 3 is the data splitting technique which had 90% of dataset as training and 10% as testing
- Performed three different data partitioning techniques
  - 1. Data Splitting
  - 2. Bootstrap Sampling
  - 3. K-fold Cross Validation
- The similarity measure used here is Jaccard Index
- Jaccard Index only takes into account the positive values and not the missing values
- Number of recommendations (M value) suggested for the users ranges from 100 to 600
- The results show that User-Based recommendations outperform the other algorithms compared for this dataset
- K-fold cross validation produces the best results closely followed by bootstrap sampling
- A t-Test between K-fold and Bootstrap sampling yielded a p-value of 0.092 (t = 1.772) for TPR and 0.99 (t~0) for FPR

Future Work
- Model-based collaborative filtering techniques such as topic modeling can be used to analyze the textual information to find the similar items. We can also use Matrix factorization technique to find the hidden factors behind the user ratings

References