BIL576 PROJECT PROPOSAL

Uğur Güdelek
m.gudelek@etu.edu.tr
Computer Engineering Department
TOBB ETU, Ankara

Simla Burcu Harma
s.harma@etu.edu.tr
Computer Engineering Department
TOBB ETU, Ankara

1.) Problem Definition

Movie Recommendation Systems have become popular with the increase in the popularity of movie streaming services like Netflix. Rather than checking the highest rated movies for a genre, users desire to find movies for their special taste. In this project, we will focus on implementing a user-based movie recommender system.

2.) Methodology

Collaborative filtering:
This type of filtering requires users' rating for movies and provides a recommendation of unwatched however liked by users similar to us. Considering the similarity of two users, similar likes and dislikes on the same content by these two users are evaluated. As a result, by looking at similar contents in common, the new content recommendation will easily be predicted for each user. However, as described above, content rates are essential however, not all movies are evaluated by every user in the system so that this causes a problem. Another issue is that diversity between users will be downgraded and within a certain subgroup of users, similar movies are recommended.

Content-based filtering:
This type of filtering does not count on users' ratings. Recommendations are provided for each user just by looking at their previous likes and dislikes. Same aforementioned issues arise when content-based filtering is applied however, these issues depend on the user itself. For instance, if the user's liking is diverse than recommendation will also be diverse.

Definition of similarity term:
Similarity measures between users or movies need to be calculated when applying collaborative or content-based filtering respectively and liking or disliking must be defined in a definitive way. Therefore, liking and disliking are vectorized. For example, each user has a liking vector which consists of all movies database has. And similarly, the same scenario is applied to movies. When creating the information vectors, similarity can be described as the similarity between vectors. Several approaches can be found in the literature however more general similarity metric is the cosine similarity described in Equation 1. Cosine similarity metric will be 1 if two vectors are identical and 0 if two vectors are orthogonal.

$$similarity = \cos(\theta) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|} = \frac{\sum_{i=1}^{n} \mathbf{u}_i \mathbf{v}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{u}_i)^2}\sqrt{\sum_{i=1}^{n} (\mathbf{v}_i)^2}} \quad \textbf{(1)}$$

Available Tools and Dataset:

In order to implement the movie recommendation system, Python 3 is selected because of the dominance in the area of data processing and machine learning. Python Data Analysis Library (pandas) and Numpy will be examined for data analysis and numerical computation respectively. If the machine learning approach is necessary to increase precision, Pytorch Deep Learning Framework will be used. Last but not least, *surprise*[1] - a python sci-kit for recommender systems- will be used to building and analyzing the movie recommender system.

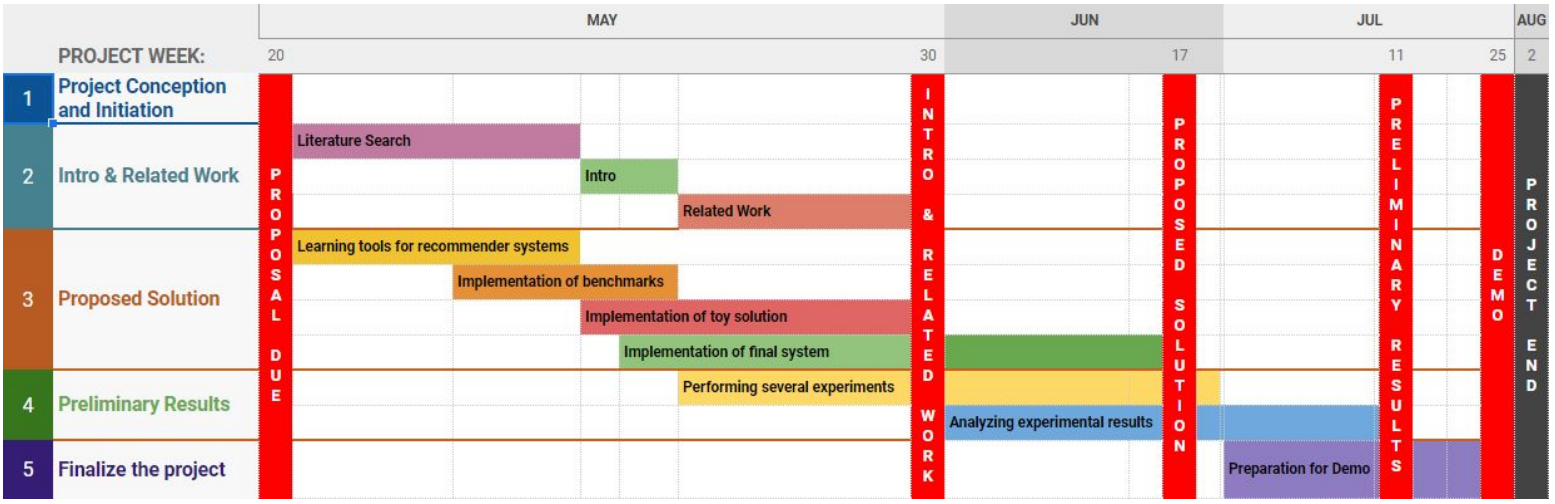
[1] http://surpriselib.com/

3.) Evaluation

For the evaluation of our work, we will use traditional metrics: Mean Absolute Error (MAE) and Root Spare Mean Error (RSME) to measure the error in the predicted ratings and Recall, Precision, Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG) to measure the quality of the top-n ranked list of items.
Experiments are examined with Netflix[2] or MovieLens[3] datasets in most of the literature, therefore, these datasets will be used in this term project.

4.) Time Plan

Milestones/Gantt Chart:



Division of Work:

While most of the work will be done together, literature search and learning part will progress in parallel. After we go in depth of the topic and decide what kind of system we will implement, we will divide the implementation part as each of us will implement a different algorithm/part in the project.

Risks:

According to our literature search so far,

- Users who do not have enough ratings may not receive relevant recommendations,
- The satisfaction of the users from different geographical locations and cultures might be challenging.

[2] https://www.kaggle.com/netflix-inc/netflix-prize-data
[3] https://grouplens.org/datasets/movielens/