****

**EC 503 Learning from Data**

**Project Summary**

**Recommendation Systems**

**Yousif Khaireddin** [**ykh@bu.edu**](mailto:ykh@bu.edu)

**Zhuofa Chen** [**zfchen@bu.edu**](mailto:zfchen@bu.edu)

**Krishna Palle** [**pallekc@bu.edu**](mailto:pallekc@bu.edu)

**Introduction**

Recommendation systems aim to narrow down choices, which plays an important role in our everyday lives and is widely used in various fields such as social media and online shopping. Almost any platform can utilize a recommendation system to study user behavior and predict responses to other products. Netflix, for instance, uses a recommendation system to suggest movies to specific users based on user data.

This project aims to use and compare multiple machine learning algorithms in building a recommendation system. The biggest challenges of a recommendation system is its data sparsity. Returning to the Netflix example, it is highly unlikely that a user has watched, let alone rated, every single movie on the platform.

One very famous recommendation system dataset is the MovieLens, found at [1]. There are multiple versions of this dataset with different ranting counts, shown below [2].

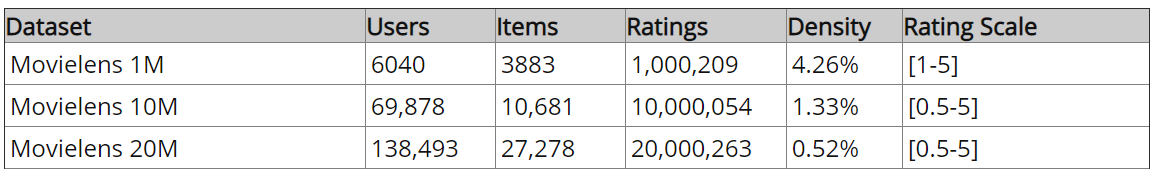


Table 1: Dataset Overview [2]

For this project, we will first build and run our algorithms on the 1M dataset. Once fine-tuned, we will then run the algorithms on one of the larger datasets.

The dataset includes six files which contain the following features:

* User ID
* Movie ID
* Rating
* Genre
* Tags
* Timestamp
* Gender
* Age
* Occupation
* Zip code

**Machine Learning Approaches:**

After some literature review, we have narrowed down our choices to the following machine learning approaches. We aim to implement and compare as many of them as possible.

**Pearson’s Correlation**

This approach finds Pearson’s correlation coefficient between movies to identify the   
users/movies that are similar to each other and make recommendation based on the correaltion

**K Nearest neighbours**

KNNs algorithm uses the K nearest neighbours to recommend movies. Here, ‘nearness’ is measured using the Euclidian distance; however, there are numerous different distance functions depending on the problem at hand.

**Singular Value Decomposition**

Looking at the density column in table 1, the sparsity problem becomes quite evident. One algorithm that has been proposed to remedy this issue is Matrix factorization. As detailed in [3], this method hopes to deconstruct the sparse ratings matrix into the dot product of 2 matrices which are learned through gradient descent with regularization. Moving forward, the users are then recommended the highest learned rating.

**Cluster & Mean Jump Algorithm**

After a brief discussion with Dr. Prakash Ishwar, we came up with the idea of clustering the users then building the dataset with the averages of movie ratings within each cluster. This process would then repeat until the movie ratings converge, if they do.

**Spotlight Framework [5]**

This is a framework which uses factorization models and sequential models to implement recommendation systems. Factorisation model uses singular value decomposition method and the sequential model builds a recommendation system as a time series model such as Long Short-Term Memory loss (LSTM) and 1-D Convolution networks.

**Distribution of Roles**

Yousif Khaireddin [ykh@bu.edu](mailto:ykh@bu.edu)

* Literature review: Pearson’s Correlation & Cluster & Mean Jump
* Implementation:
  + Pearson’s Correlation
  + SVD
  + Spotlight Framework
  + Cluster & Mean Jump

Zhuofa Chen [zfchen@bu.edu](mailto:zfchen@bu.edu)

* Literature Review: Spotlight Framework & Cluster & Mean Jump
* Implementation:
  + KNNs
  + SVD
  + Spotlight Framework
  + Cluster & Mean Jump

Krishna Palle [pallekc@bu.edu](mailto:pallekc@bu.edu)

* Literature Review: SVD & Cluster & Mean Jump
* Implementation:
  + Pearson’s Correlation
  + SVD
  + Spotlight Framework
  + Cluster & Mean Jump

**References**

[1] GroupLens. (2019). *MovieLens*. [online] Available at: https://grouplens.org/datasets/movielens/ [Accessed 26 Oct. 2019].

[2] Gude, A. (2019). *9 Must-Have Datasets for Investigating Recommender Systems - KDnuggets*. [online] KDnuggets. Available at: https://www.kdnuggets.com/2016/02/nine-datasets-investigating-recommender-systems.html [Accessed 26 Oct. 2019].

[3] Koren, Y., Bell, R. and Volinsky, C., 2009. Matrix factorization techniques for recommender systems. *Computer*, (8), pp.30-37.

[4] Schober, P., Boer, C. and Schwarte, L.A., 2018. Correlation coefficients: appropriate use and interpretation. *Anesthesia & Analgesia*, *126*(5), pp.1763-1768.

[5] GitHub. (2019). *maciejkula/spotlight*. [online] Available at: https://github.com/maciejkula/spotlight [Accessed 26 Oct. 2019].