# Assignment on Recommender Systems

Instructor: Dr. Manish Singh TA: Sailaja Rajanala

## Section A: Know Your Data

* Load and visualize IMDB movie data for content based recommender systems.
* Understand the content features/descriptors of the dataset(title, director, actors, plot, genre).
* Visualize the Movielens data used for collaborative filtering.

EX: Learn the basic utility functions used to load and visualize the data.

## Section B: Content Based Recommendations

* Filtering the data: constructing reduced dataset using content features (title, director, actors, plot, genre).
* Data preprocessing : cleaning the text, removing special characters and converting to lowercase.
* Data preprocessing : extracting keywords/key terms from movie plots.
* Constructing feature vectors for items using bag of words.
* Obtain the word count vectors using CountVectorizer.
* Learning to compute similarity between items using cosine similarity measure.
* Making recommendations using similarity.

EX1: Use TfidfVectorizer instead of CountVectorizer to obtain the word count vectors and report the change in predictions.

EX2: What is the aim of TF-IDF?

EX3: What is the input to cosine similarity function.

EX4: What is the structure of the data structure containing the representations of the movies?

EX5: Make recommendations for movieId 16.

## Section C: Collaborative Filtering

* Learning to use “surprise” library of python for collaborative filtering.
* Implementing a SVD based collaborative filtering system using “surprise” library.
* Compute the RMSE error of the recommender system.
* Learning to calculate sparsity of the dataset.
* Building a sparse dataset.
* Understanding a system cold start problem.
* Building a recommender system for sparse data (system cold start).
* Analyze the effect of sparsity on RMSE.
* Understanding the effects of various hyperparameters while making recommendations using KNN, SVD and PMF.

EX6: Fine tune hyperparameters to improve the RMSE.

EX7: How does sparsity effect RMSE?

EX8: How to compute item-item collaborative filtering?

EX9: Use different similarity measure to make recommendations.

EX10: Compare the RMSE values of the algorithms by playing around the hyperparameters such as the number of neighbours “k”, number of cross validations folds “n\_folds” the similarity measures “name” and record your observations.

## Section C: Evaluation Measures for Recommender Systems

* Understanding evaluation measures used in recommender systems

EX11: Match the description of the error function with its name.