Based on paper “Evaluating Recommendation Systems” by Guy Shani and Asela Gunawardana, in order to evaluate content based recommender algorithms ofﬂine, it is necessary to simulate the behavior of users that interact with a recommendation system. Offline experimentation is a type of experimentation that uses historical data to simulate how users would interact with a recommendation system. This type of experimentation is attractive because it is quick, cheap, and can be used to test a wide range of algorithms. However, offline experiments can only answer a limited set of questions. For example, they can be used to compare the prediction power of different algorithms, but they cannot be used to directly measure the impact of a recommendation system on user behavior.

It is important to use relevant and unbiased data for offline evaluation of recommender systems, in order to effectively compare algorithms before deployment. Pre-filtering data by excluding users or items with low counts would introduce bias. Randomly sampling users and items can be a better method for reducing data, and correcting biases in the data may require techniques such as reweighing data.

In this study, two evaluation functions have been added to the recommender.py:

Function get\_evaluation\_title, takes in a query movie title and a list of movie title suggestions recommended by get\_recommendations\_by\_title for an input query movie title and calculates precision score for recommending these movies to a subset of users based on their past movie ratings. The function first classifies users into groups based on the number of movies they have rated and assigns higher weights to smaller groups by adjusting the weights based on the number of users in each group. It then randomly samples a subset (2000) of these users and filters users and their movie list to only include the selected subset. It then calculates the precision score for the selected subset of users based on whether they have rated the selected movie and the suggested movies, and whether they have given high ratings to the selected movie. The precision score is calculated as the number of correctly recommended movies divided by the total number of recommended movies.

The function returns the precision score is calculated as the average precision across all selected subset of users in the dictionary.

The function get\_evaluation\_keyword is doing the same as function get\_evaluation\_title but takes in just a list of movie title suggestions recommended by get\_recommendations\_by\_keyword\_bert\_tfidf for an input keyword and calculates precision score for recommending these movies to a subset of users based on their past movie ratings.