Assignment 1: Building a recommendation system

This assignment should be carried out using Python 3, Jupyter Notebooks, NumPy, Pandas, Matplotlib/Seaborn, NLTK, Scikit-learn and (optionally) SciPy.

Assignment

Many websites give users the possibility to rate items nowadays. Companies such as Amazon, Netflix, YouTube, IMDB and Bol.com use this information to recommend similar items to their users. The MovieLens dataset is a free dataset with a collection of movie ratings.

In this assignment you will build two recommendation systems, using the following techniques: content-based and collaborative filtering.

1. Content-based

For the assignment we will use a JSON file that has been published on blackboard containing movie information for all the movies in the MovieLens dataset. It is your task to extract features for each movie by using techniques such as stopword removal, stemming, bag of words, n-grams and TF-IDF.

We have also published a Python module that can display movies in IMDB-style.

The last cells of your notebook should consist of two small programs with Jupyter widgets and the provided Python module to test your recommendation systems (please read the documentation about the interact function within the widgets library):

1. Select a movie title and show N recommendations (based on text analysis of at least the plot, the title and the writers). Please explain how you merged the recommendations of each bag-of-words/TF-IDF set.
2. Program a second strategy for recommending movies based on a "watched-list". Given your list, show the N best recommendations. Please explain the merge-strategy that you used. **It is important that you use a different strategy here!**

Make sure the N is configurable for every program!

Please note: it is sufficient to use the interact function. You don't have to spend time on building a complex user interface!

2. Collaborative filtering

User-item matrix

Set up the user-item matrix using the provided data set. You can use Pandas to convert the dataset into the user-item representation.

Similarity metrics

Implement the different similarity metrics:

- Cosine similarity
- Pearson similarity
- Adjusted cosine similarity

Provide a small example (e.g. the example in the slides) to show that you implementation is correct!

User-user recommendation

Convert the user-item matrix into a user-user similarity matrix, by applying the similarity metrics. Implement the user-user recommendation algorithm.

Give N (configurable) recommendations for a given user U (configurable) and a given similarity S (configurable). Use the 10 most similar users to base your recommendations on.

Provide a small example to show that your algorithm is working correctly!

Item-Item recommendation

Convert the user-item matrix into a item-item similarity matrix, by applying the similarity metrics. Implement the item-item recommendation algorithm.

Give N (configurable) recommendations for a given user U (configurable) based on the movies the user U rated with **at least 3.5 stars**. Explain your implementation and the strategy that you use for selecting the final recommendations!

Provide a small example to show that your algorithm is working correctly!

Validation

Evaluate the accuracy of both the user-user and the item-item collaborative filtering recommendations using the hit rate metric. Additionally you can validate your recommendation system also with the RMSE (extra points).

The last cell of your notebook should consist of a small program with Jupyter widgets and the provided Python module which should do the following: Select a recommendation type (user-user or item-item) and similarity measure (cosine, pearson, adjusted cosine) and show the N recommendations and the calculated ratings per recommendation.

Dataset

We will use a modified version of the MovieLens data set for this assignment. You can download the small data set with an additional movie information file (JSON) from blackboard.

Additional requirements

- **The resulting notebooks should read as a report. Explain your implementation, the strategy that you use for selecting/merging the best recommendations.**
- **Please provide small examples in your notebooks in which you show that the algorithms that you implement work correctly!**
- The content-based filtering data set should be constructed using stemming, stopword removal and Tf-Idf. Depending on the type of field that you are using you have to decide which filtering techniques are necessary.