Univerza v Ljubljani
Fakulteta
za računalništvo
in informatiko

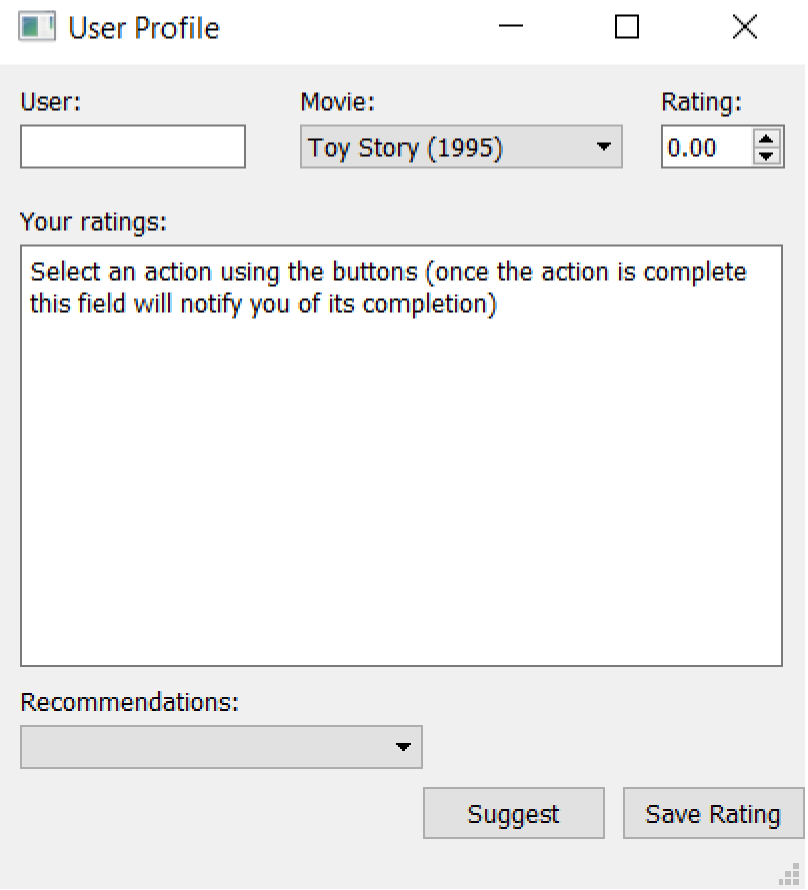
Recommender System

Fourth homework assignment for Datamining

Ardit Nela (63190375)

User Interface

I created a GUI using the pyqt library (the template file is stored as a .ui file and I load it in my main python script).



User:

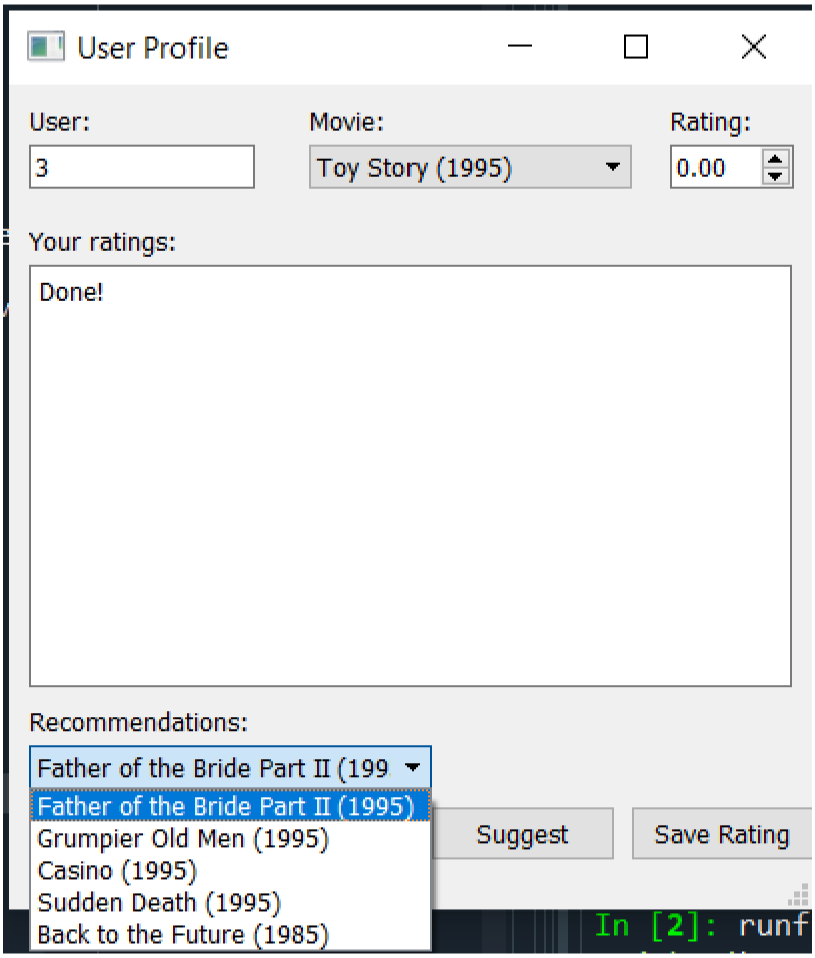
Movie: Toy Story (1995)

Rating: 0.00

Your ratings:

Select an action using the buttons (once the action is complete this field will notify you of its completion)

Default User Profile



User: 3

Movie: Toy Story (1995)

Rating: 0.00

Your ratings:

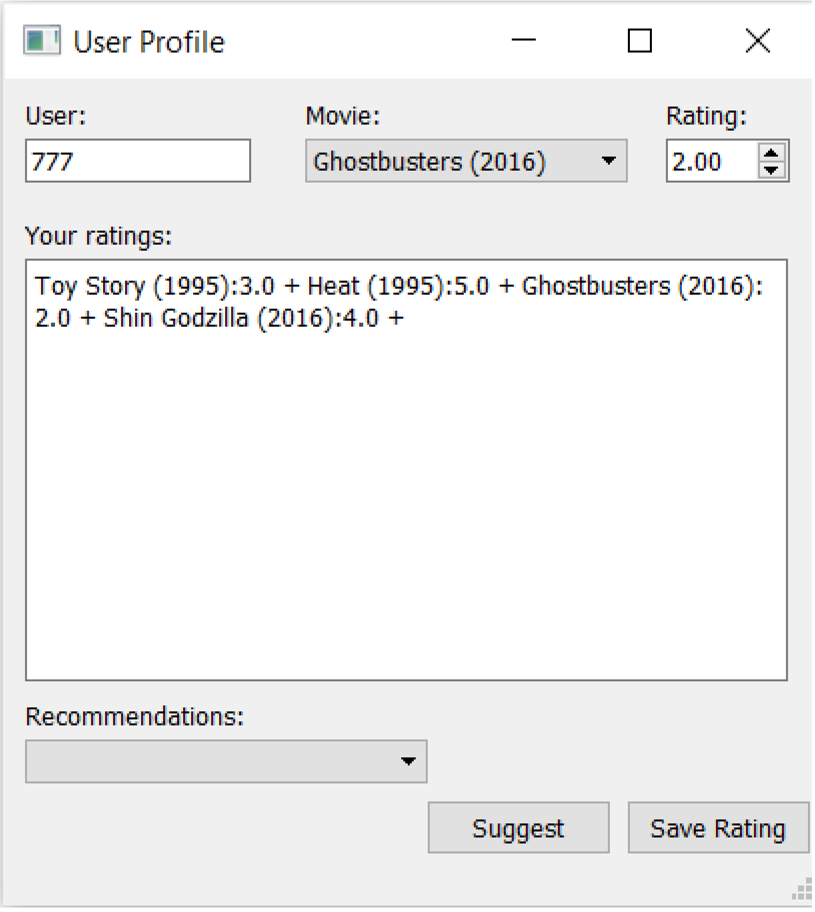
Done!

Recommendations: Father of the Bride Part II (199

- Father of the Bride Part II (1995)
- Grumpier Old Men (1995)
- Casino (1995)
- Sudden Death (1995)
- Back to the Future (1985)

```
In [2]: runf
```

After a user ID is entered and the “Suggest” button is pressed, your movie recommendations will appear in a combo box.



User: 777

Movie: Ghostbusters (2016)

Rating: 2.00

Your ratings:

Toy Story (1995):3.0 + Heat (1995):5.0 + Ghostbusters (2016):2.0 + Shin Godzilla (2016):4.0 +

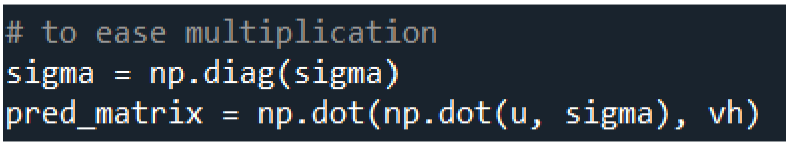
Recommendations:

You may also add a new rating for the given user.

NOTE: Suggestions only work for already existing users.

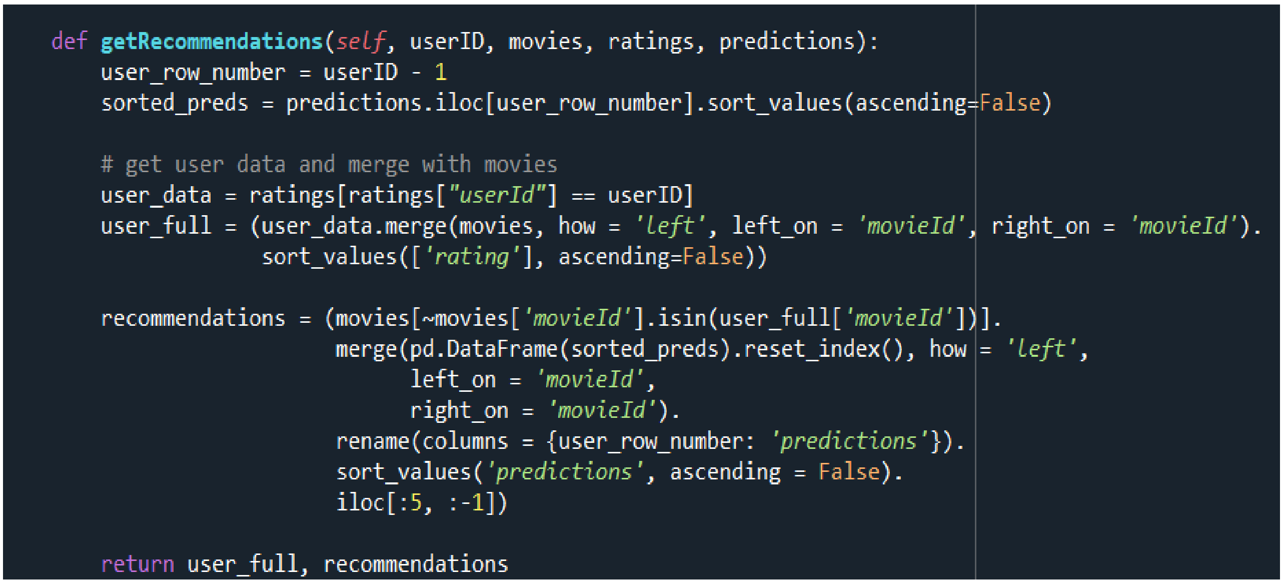
Recommender

I decided to use matrix factorization (more specifically, singular value decomposition) to build my recommender system (I was inspired by the early days of Netflix). I used the built-in numpy.linalg.svg function to get u, sigma and vh. Afterwards, I calculated the recommendation matrix by using their dot product:



```
# to ease multiplication
sigma = np.diag(sigma)
pred_matrix = np.dot(np.dot(u, sigma), vh)
```

Finally, I wrote a function that would give 5 suggestions for the given user based on the prediction matrix:



```
    def getRecommendations(self, userID, movies, ratings, predictions):
        user_row_number = userID - 1
        sorted_preds = predictions.iloc[user_row_number].sort_values(ascending=False)

        # get user data and merge with movies
        user_data = ratings[ratings["userId"] == userID]
        user_full = (user_data.merge(movies, how = 'left', left_on = 'movieId', right_on = 'movieId').
                     sort_values(['rating'], ascending=False))

        recommendations = (movies[~movies['movieId'].isin(user_full['movieId'])].
                           merge(pd.DataFrame(sorted_preds).reset_index(), how = 'left',
                                 left_on = 'movieId',
                                 right_on = 'movieId').
                           rename(columns = {user_row_number: 'predictions'}).
                           sort_values('predictions', ascending = False).
                           iloc[:5, :-1])

        return user_full, recommendations
```