Assignment 2 (DAT340) - recommender system

January 29, 2024

1 Assignment 2 - recommender system

- Student 1 - Luca Modica
- Student 2 - Hugo Alves Henriques E Silva

---

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

sns.set_style()
%matplotlib inline
```

1.1 Reading data

[2]:
```
user_reviews = pd.read_csv('user_reviews.csv')
movie_genres = pd.read_csv('movie_genres.csv')
user_reviews.drop(columns=['Unnamed: 0'], inplace=True)
movie_genres.drop(columns=['Unnamed: 0'], inplace=True)
```

Convert `user_reviews` into an user-item matrix:

[3]:
```
user_reviews.set_index('User', inplace=True)
user_reviews.drop(user_reviews.columns[0], axis=1, inplace=True)

user_reviews.head()
```

[3]:
```
         Happily N'Ever After  Tomorrowland  American Hero  Das Boot  \
User
Vincent                   0.0           0.0            0.0       0.0
Edgar                     0.0           0.0            0.0       0.0
Addilyn                   0.0           0.0            0.0       0.0
Marlee                    0.0           0.0            0.0       0.0
Javier                    0.0           0.0            0.0       0.0
```

```
         Final Destination 3  Licence to Kill  The Hundred-Foot Journey  \
User
Vincent                  0.0              0.0                       0.0
Edgar                    0.0              0.0                       0.0
Addilyn                  0.0              0.0                       0.0
Marlee                   0.0              0.0                       0.0
Javier                   0.0              0.0                       0.0

         The Matrix  Creature  The Basket  …  The Martian  Micmacs  \
User                                       …
Vincent         0.0       0.0         0.0  …          0.0      0.0
Edgar           0.0       0.0         0.0  …          0.0      0.0
Addilyn         0.0       0.0         0.0  …          0.0      0.0
Marlee          0.0       0.0         0.0  …          0.0      0.0
Javier          0.0       0.0         0.0  …          0.0      0.0

         Solomon and Sheba  In the Company of Men  Silent House  Big Fish  \
User
Vincent                0.0                    0.0           0.0       0.0
Edgar                  0.0                    0.0           0.0       0.0
Addilyn                0.0                    0.0           0.0       0.0
Marlee                 0.0                    0.0           0.0       0.0
Javier                 0.0                    0.0           0.0       0.0

         Get Real  Trading Places  DOA: Dead or Alive  Hey Arnold! The Movie
User
Vincent       0.0             0.0                 0.0                    0.0
Edgar         0.0             0.0                 0.0                    0.0
Addilyn       0.0             0.0                 0.0                    0.0
Marlee        0.0             0.0                 0.0                    0.0
Javier        0.0             0.0                 0.0                    0.0

[5 rows x 1999 columns]
```

1.2 A first approach: using cosine similarity

Computing cosine similarity between users:

[4]:

```python
from sklearn.metrics.pairwise import cosine_similarity

# Replace 0s with NaN for calculation purposes
user_reviews_nan = user_reviews.replace(0, np.NaN)

cosine_sim = cosine_similarity(user_reviews_nan.fillna(0))
user_sim_df = pd.DataFrame(cosine_sim, index=user_reviews.index,␣
 ↪columns=user_reviews.index)
```

```
user_sim_df.head()
```

[4]:

```
User      Vincent     Edgar   Addilyn    Marlee    Javier  Marcus      Mary  \
User
Vincent  1.000000  0.016249  0.020722  0.000000  0.000000     0.0  0.040862
Edgar    0.016249  1.000000  0.000000  0.000000  0.030276     0.0  0.000000
Addilyn  0.020722  0.000000  1.000000  0.042046  0.004826     0.0  0.035055
Marlee   0.000000  0.000000  0.042046  1.000000  0.032875     0.0  0.000000
Javier   0.000000  0.030276  0.004826  0.032875  1.000000     0.0  0.000000

User     Rosalie  Giovanni   Kennedi  …     Piper  Tatum     Jonah  \
User                                  …
Vincent      0.0   0.02631  0.049025  …  0.000000    0.0  0.024604
Edgar        0.0   0.00000  0.054967  …  0.000000    0.0  0.000000
Addilyn      0.0   0.00000  0.000000  …  0.000000    0.0  0.000000
Marlee       0.0   0.00000  0.000000  …  0.021514    0.0  0.000000
Javier       0.0   0.00000  0.000000  …  0.000000    0.0  0.000000

User        Nylah     James   Mariana  Ivy     Kevin      Nora     Sarai
User
Vincent  0.043098  0.000000  0.013294  0.0  0.000000  0.000000  0.000000
Edgar    0.000000  0.000000  0.000000  0.0  0.000000  0.000000  0.026195
Addilyn  0.000000  0.000000  0.000000  0.0  0.019512  0.000000  0.055121
Marlee   0.000000  0.035800  0.000000  0.0  0.000000  0.043225  0.000000
Javier   0.020076  0.032875  0.049542  0.0  0.000000  0.074426  0.000000

[5 rows x 600 columns]
```

Then, we will implement a function `predict_ratings` to predict the first `top_n` movies to a specific users, based on the weighted average of ratings from similar ones. The similarity between users is based on the cosine similarity matrix previously computed.

[5]:

```
def get_top_n_recommendations(user_name, user_reviews, user_sim_df, n=5):
    unrated_movies = user_reviews.loc[user_name][
        user_reviews.loc[user_name].isna()]

    # Predict ratings for each unrated movie
    predictions = {}
    for movie in unrated_movies.index:
        # Similar users who have rated this movie
        similar_users = user_reviews[movie].dropna().index
        similar_users = similar_users[similar_users != user_name]

        # Calculate the weighted average rating
        ratings = user_reviews.loc[similar_users, movie]
        similarities = user_sim_df.loc[user_name, similar_users]
        weighted_ratings = ratings * similarities
        if similarities.sum() > 0:
            predicted_rating = weighted_ratings.sum() / similarities.sum()
            predictions[movie] = predicted_rating

    sorted_predictions = sorted(predictions.items(),
        key=lambda x: x[1], reverse=True)

    return sorted_predictions[:n]
```

[6]:

```
# Predict ratings for the first five users and recommend top 5
# unseen movies based on it
top_n_recommendations = {}
for user in user_reviews.index[:5]:
    recommendations = '['
    top_n_recommendations[user] = get_top_n_recommendations(user,␣
 ↪user_reviews_nan, user_sim_df)
    for movie, review in top_n_recommendations[user]:
        recommendations += movie + ', '
    recommendations += ']'
    print(f'New films recommended for {user}: {recommendations}')
```

```
New films recommended for Vincent: [Das Boot, Never Back Down 2: The Beatdown,
Fool's Gold, Jonah: A VeggieTales Movie, Eagle Eye, ]
New films recommended for Edgar: [The Contender, Maid in Manhattan, Mad City,
Get Carter, The Talented Mr. Ripley, ]
New films recommended for Addilyn: [Space: Above and Beyond, Dutch Kills,
Highlander: Endgame, The Tempest, Amadeus, ]
New films recommended for Marlee: [The Grandmaster, Big, Boiler Room, To Kill a
Mockingbird, Vicky Cristina Barcelona, ]
New films recommended for Javier: [Thirteen, Bad Company, Superman, Home,
Morning Glory, ]
```