Content based

November 1, 2020

```
[26]: #content-based recommender system
```

```
[27]: import numpy as np
      import pandas as pd
      import matplotlib.pyplot as plt
      from math import sqrt
      %matplotlib inline
```

```
[28]: movies_df = pd.read_csv('movies.csv')
      ratings_df = pd.read_csv('ratings.csv')
      movies_df.head()
```

```
[28]:    movieId                               title  \
      0        1                    Toy Story (1995)
      1        2                      Jumanji (1995)
      2        3             Grumpier Old Men (1995)
      3        4            Waiting to Exhale (1995)
      4        5  Father of the Bride Part II (1995)

                                              genres
      0  Adventure|Animation|Children|Comedy|Fantasy
      1                   Adventure|Children|Fantasy
      2                               Comedy|Romance
      3                         Comedy|Drama|Romance
      4                                       Comedy
```

```
[29]: #Removing year from the title and storing it in different column
```

```
[30]: movies_df['year'] = movies_df.title.str.extract('(\(\d\d\d\d\))',expand=False)
      #Removing the parentheses
      movies_df['year'] = movies_df.year.str.extract('(\d\d\d\d)',expand=False)
      #Removing the years from the 'title' column
      movies_df['title'] = movies_df.title.str.replace('(\(\d\d\d\d\))', '')
      #Applying the strip function to get rid of any ending whitespace characters␣
       ↪that may have appeared
      movies_df['title'] = movies_df['title'].apply(lambda x: x.strip())
      movies_df.head()
```

[30]:
```
   movieId                         title  \
0        1                     Toy Story
1        2                       Jumanji
2        3              Grumpier Old Men
3        4             Waiting to Exhale
4        5   Father of the Bride Part II

                                        genres  year
0  Adventure|Animation|Children|Comedy|Fantasy  1995
1                   Adventure|Children|Fantasy  1995
2                               Comedy|Romance  1995
3                         Comedy|Drama|Romance  1995
4                                       Comedy  1995
```

[31]:
```python
movies_df['genres'] = movies_df.genres.str.split('|')
movies_df.head()
```

[31]:
```
   movieId                         title  \
0        1                     Toy Story
1        2                       Jumanji
2        3              Grumpier Old Men
3        4             Waiting to Exhale
4        5   Father of the Bride Part II

                                              genres  year
0  [Adventure, Animation, Children, Comedy, Fantasy]  1995
1                     [Adventure, Children, Fantasy]  1995
2                                  [Comedy, Romance]  1995
3                           [Comedy, Drama, Romance]  1995
4                                           [Comedy]  1995
```

[32]:
```python
#Copying the movie dataframe into a new one since we won't need to use the␣
 ↪genre information in our first case.
moviesWithGenres_df = movies_df.copy()

#For every row in the dataframe, iterate through the list of genres and place a␣
 ↪1 into the corresponding column
for index, row in movies_df.iterrows():
    for genre in row['genres']:
        moviesWithGenres_df.at[index, genre] = 1
#Filling in the NaN values with 0 to show that a movie doesn't have that␣
 ↪column's genre
moviesWithGenres_df = moviesWithGenres_df.fillna(0)
moviesWithGenres_df.head()
```

[32]:
```
   movieId                         title  \
0        1                     Toy Story
```

```
1        2                         Jumanji
2        3                Grumpier Old Men
3        4               Waiting to Exhale
4        5   Father of the Bride Part II

                                         genres  year  Adventure  \
0  [Adventure, Animation, Children, Comedy, Fantasy]  1995        1.0
1                   [Adventure, Children, Fantasy]  1995        1.0
2                                [Comedy, Romance]  1995        0.0
3                         [Comedy, Drama, Romance]  1995        0.0
4                                         [Comedy]  1995        0.0

   Animation  Children  Comedy  Fantasy  Romance  …  Horror  Mystery  \
0        1.0       1.0     1.0      1.0      0.0  …     0.0      0.0
1        0.0       1.0     0.0      1.0      0.0  …     0.0      0.0
2        0.0       0.0     1.0      0.0      1.0  …     0.0      0.0
3        0.0       0.0     1.0      0.0      1.0  …     0.0      0.0
4        0.0       0.0     1.0      0.0      0.0  …     0.0      0.0

   Sci-Fi  IMAX  Documentary  War  Musical  Western  Film-Noir  \
0     0.0   0.0          0.0  0.0      0.0      0.0        0.0
1     0.0   0.0          0.0  0.0      0.0      0.0        0.0
2     0.0   0.0          0.0  0.0      0.0      0.0        0.0
3     0.0   0.0          0.0  0.0      0.0      0.0        0.0
4     0.0   0.0          0.0  0.0      0.0      0.0        0.0

   (no genres listed)
0                 0.0
1                 0.0
2                 0.0
3                 0.0
4                 0.0

[5 rows x 24 columns]
```

[33]:
```
ratings_df.head()
```

[33]:
```
   userId  movieId  rating   timestamp
0       1      169     2.5  1204927694
1       1     2471     3.0  1204927438
2       1    48516     5.0  1204927435
3       2     2571     3.5  1436165433
4       2   109487     4.0  1436165496
```

[36]:
```
ratings_df = ratings_df.drop('timestamp',axis=1)
```

[37]:
```
ratings_df.head()
```

[37]:
```
   userId  movieId  rating
0       1      169     2.5
1       1     2471     3.0
2       1    48516     5.0
3       2     2571     3.5
4       2   109487     4.0
```

[38]:
```
userInput = [
            {'title':'Breakfast Club, The', 'rating':5},
            {'title':'Toy Story', 'rating':3.5},
            {'title':'Jumanji', 'rating':2},
            {'title':"Pulp Fiction", 'rating':5},
            {'title':'Akira', 'rating':4.5}
         ]
inputMovies = pd.DataFrame(userInput)
inputMovies
```

[38]:
```
                 title  rating
0  Breakfast Club, The     5.0
1            Toy Story     3.5
2              Jumanji     2.0
3         Pulp Fiction     5.0
4                Akira     4.5
```

[39]:
```
#Filtering out the movies by title
inputId = movies_df[movies_df['title'].isin(inputMovies['title'].tolist())]
#Then merging it so we can get the movieId. It's implicitly merging it by title.
inputMovies = pd.merge(inputId, inputMovies)
#Dropping information we won't use from the input dataframe
inputMovies = inputMovies.drop('genres', 1).drop('year', 1)
#Final input dataframe
#If a movie you added in above isn't here, then it might not be in the original
#dataframe or it might spelled differently, please check capitalisation.
inputMovies
```

[39]:
```
   movieId                title  rating
0        1            Toy Story     3.5
1        2              Jumanji     2.0
2      296         Pulp Fiction     5.0
3     1274                Akira     4.5
4     1968  Breakfast Club, The     5.0
```

[40]:
```
#Filtering out the movies from the input
userMovies = moviesWithGenres_df[moviesWithGenres_df['movieId'].
 ↪isin(inputMovies['movieId'].tolist())]
userMovies
```

```
[40]:       movieId                title  \
      0           1            Toy Story
      1           2              Jumanji
      293       296          Pulp Fiction
      1246     1274                Akira
      1885     1968  Breakfast Club, The

                                                genres  year  Adventure  \
      0     [Adventure, Animation, Children, Comedy, Fantasy]  1995        1.0
      1                   [Adventure, Children, Fantasy]  1995        1.0
      293                 [Comedy, Crime, Drama, Thriller]  1994        0.0
      1246          [Action, Adventure, Animation, Sci-Fi]  1988        1.0
      1885                              [Comedy, Drama]  1985        0.0

            Animation  Children  Comedy  Fantasy  Romance  …  Horror  Mystery  \
      0           1.0       1.0     1.0      1.0      0.0  …     0.0      0.0
      1           0.0       1.0     0.0      1.0      0.0  …     0.0      0.0
      293         0.0       0.0     1.0      0.0      0.0  …     0.0      0.0
      1246        1.0       0.0     0.0      0.0      0.0  …     0.0      0.0
      1885        0.0       0.0     1.0      0.0      0.0  …     0.0      0.0

            Sci-Fi  IMAX  Documentary  War  Musical  Western  Film-Noir  \
      0        0.0   0.0          0.0  0.0      0.0      0.0        0.0
      1        0.0   0.0          0.0  0.0      0.0      0.0        0.0
      293      0.0   0.0          0.0  0.0      0.0      0.0        0.0
      1246     1.0   0.0          0.0  0.0      0.0      0.0        0.0
      1885     0.0   0.0          0.0  0.0      0.0      0.0        0.0

            (no genres listed)
      0                    0.0
      1                    0.0
      293                  0.0
      1246                 0.0
      1885                 0.0

      [5 rows x 24 columns]
```

```
[41]: #We'll only need the actual genre table,
      #so let's clean this up a bit by resetting the index and dropping the movieId,␣
       ↪title, genres and year columns.
```

```
[42]: #Resetting the index to avoid future issues
      userMovies = userMovies.reset_index(drop=True)
      #Dropping unnecessary issues due to save memory and to avoid issues
      userGenreTable = userMovies.drop('movieId', 1).drop('title', 1).drop('genres',␣
       ↪1).drop('year', 1)
      userGenreTable
```

[42]:
```
   Adventure  Animation  Children  Comedy  Fantasy  Romance  Drama  Action  \
0        1.0        1.0       1.0     1.0      1.0      0.0    0.0     0.0
1        1.0        0.0       1.0     0.0      1.0      0.0    0.0     0.0
2        0.0        0.0       0.0     1.0      0.0      0.0    1.0     0.0
3        1.0        1.0       0.0     0.0      0.0      0.0    0.0     1.0
4        0.0        0.0       0.0     1.0      0.0      0.0    1.0     0.0

   Crime  Thriller  Horror  Mystery  Sci-Fi  IMAX  Documentary  War  Musical  \
0    0.0       0.0     0.0      0.0     0.0   0.0          0.0  0.0      0.0
1    0.0       0.0     0.0      0.0     0.0   0.0          0.0  0.0      0.0
2    1.0       1.0     0.0      0.0     0.0   0.0          0.0  0.0      0.0
3    0.0       0.0     0.0      0.0     1.0   0.0          0.0  0.0      0.0
4    0.0       0.0     0.0      0.0     0.0   0.0          0.0  0.0      0.0

   Western  Film-Noir  (no genres listed)
0      0.0        0.0                 0.0
1      0.0        0.0                 0.0
2      0.0        0.0                 0.0
3      0.0        0.0                 0.0
4      0.0        0.0                 0.0
```

[43]:
```python
#Dot produt to get weights
userProfile = userGenreTable.transpose().dot(inputMovies['rating'])
#The user profile
userProfile
```

[43]:
```
Adventure             10.0
Animation              8.0
Children               5.5
Comedy                13.5
Fantasy                5.5
Romance                0.0
Drama                 10.0
Action                 4.5
Crime                  5.0
Thriller               5.0
Horror                 0.0
Mystery                0.0
Sci-Fi                 4.5
IMAX                   0.0
Documentary            0.0
War                    0.0
Musical                0.0
Western                0.0
Film-Noir              0.0
(no genres listed)     0.0
dtype: float64
```

```
[44]: #Now let's get the genres of every movie in our original dataframe
genreTable = moviesWithGenres_df.set_index(moviesWithGenres_df['movieId'])
#And drop the unnecessary information
genreTable = genreTable.drop('movieId', 1).drop('title', 1).drop('genres', 1).
 ↪drop('year', 1)
genreTable.head()
```

```
[44]:          Adventure  Animation  Children  Comedy  Fantasy  Romance  Drama  \
movieId
1              1.0        1.0       1.0     1.0      1.0      0.0    0.0
2              1.0        0.0       1.0     0.0      1.0      0.0    0.0
3              0.0        0.0       0.0     1.0      0.0      1.0    0.0
4              0.0        0.0       0.0     1.0      0.0      1.0    1.0
5              0.0        0.0       0.0     1.0      0.0      0.0    0.0

         Action  Crime  Thriller  Horror  Mystery  Sci-Fi  IMAX  Documentary  \
movieId
1           0.0    0.0       0.0     0.0      0.0     0.0   0.0          0.0
2           0.0    0.0       0.0     0.0      0.0     0.0   0.0          0.0
3           0.0    0.0       0.0     0.0      0.0     0.0   0.0          0.0
4           0.0    0.0       0.0     0.0      0.0     0.0   0.0          0.0
5           0.0    0.0       0.0     0.0      0.0     0.0   0.0          0.0

         War  Musical  Western  Film-Noir  (no genres listed)
movieId
1        0.0      0.0      0.0        0.0                 0.0
2        0.0      0.0      0.0        0.0                 0.0
3        0.0      0.0      0.0        0.0                 0.0
4        0.0      0.0      0.0        0.0                 0.0
5        0.0      0.0      0.0        0.0                 0.0
```

```
[45]: #Multiply the genres by the weights and then take the weighted average
recommendationTable_df = ((genreTable*userProfile).sum(axis=1))/(userProfile.
 ↪sum())
recommendationTable_df.head()
```

```
[45]: movieId
1    0.594406
2    0.293706
3    0.188811
4    0.328671
5    0.188811
dtype: float64
```

```
[46]: #Sort our recommendations in descending order
recommendationTable_df = recommendationTable_df.sort_values(ascending=False)
#Just a peek at the values
```

```
recommendationTable_df.head()
```

```
[46]: movieId
      5018      0.748252
      26093     0.734266
      27344     0.720280
      148775    0.685315
      6902      0.678322
      dtype: float64
```

```
[47]: #The final recommendation table
      movies_df.loc[movies_df['movieId'].isin(recommendationTable_df.head(20).keys())]
```

```
[47]:        movieId                                               title  \
      664        673                                           Space Jam
      1824      1907                                               Mulan
      2902      2987                            Who Framed Roger Rabbit?
      4923      5018                                            Motorama
      6793      6902                                      Interstate 60
      8605     26093            Wonderful World of the Brothers Grimm, The
      8783     26340  Twelve Tasks of Asterix, The (Les douze travau…
      9296     27344  Revolutionary Girl Utena: Adolescence of Utena…
      9825     32031                                              Robots
      11716    51632                              Atlantis: Milo's Return
      11751    51939                   TMNT (Teenage Mutant Ninja Turtles)
      13250    64645                                    The Wrecking Crew
      16055    81132                                               Rubber
      18312    91335                                        Gruffalo, The
      22778   108540           Ernest & Célestine (Ernest et Célestine)
      22881   108932                                       The Lego Movie
      25218   117646                        Dragonheart 2: A New Beginning
      26442   122787                                        The 39 Steps
      32854   146305                               Princes and Princesses
      33509   148775                    Wizards of Waverly Place: The Movie

                                                        genres  year
      664    [Adventure, Animation, Children, Comedy, Fanta…  1996
      1824   [Adventure, Animation, Children, Comedy, Drama…  1998
      2902   [Adventure, Animation, Children, Comedy, Crime…  1988
      4923   [Adventure, Comedy, Crime, Drama, Fantasy, Mys…  1991
      6793   [Adventure, Comedy, Drama, Fantasy, Mystery, S…  2002
      8605   [Adventure, Animation, Children, Comedy, Drama…  1962
      8783   [Action, Adventure, Animation, Children, Comed…  1976
      9296   [Action, Adventure, Animation, Comedy, Drama, …  1999
      9825   [Adventure, Animation, Children, Comedy, Fanta…  2005
      11716  [Action, Adventure, Animation, Children, Comed…  2003
      11751  [Action, Adventure, Animation, Children, Comed…  2007
```

```
13250  [Action, Adventure, Comedy, Crime, Drama, Thri…  1968
16055  [Action, Adventure, Comedy, Crime, Drama, Film…  2010
18312    [Adventure, Animation, Children, Comedy, Drama]  2009
22778  [Adventure, Animation, Children, Comedy, Drama…  2012
22881  [Action, Adventure, Animation, Children, Comed…  2014
25218  [Action, Adventure, Comedy, Drama, Fantasy, Th…  2000
26442  [Action, Adventure, Comedy, Crime, Drama, Thri…  1959
32854  [Animation, Children, Comedy, Drama, Fantasy, …  2000
33509  [Adventure, Children, Comedy, Drama, Fantasy, …  2009
```

```
[ ]:
```