Report Part 1: Movie Recommendation Project

Belén García Pascual

The idea of this first part of the project is to use data (which is given clean) to create a recommender model about movies to users, according to how they previously rated other movies or their similarities to other users.

The data consists of three files:

1. The file “rangering.csv” consists of 5791 rankings, and contains one column “BrukerID” with users IDs, between 0 and 300; a second column “FilmID” with a movie ID that the corresponding user has ranked; a third column “Rangering” with the ratings of the movies that each user has done, from 1 to 5 stars (always integer numbers); and a fourth column “Tidstempel” with the time when the movie was ranked, expressed in seconds since 01.01.1970.
2. The file “bruker.csv” consists of 200 users, and contains one column “BrukerID” with the users IDs; a second colums “Kjonn” with the genders, where “F” denotes woman and “M” man; a third column “Alder” with the age gap of each user from 1 to 56, where:
    * 1: "Under 18"
    * 18: "18-24"
    * 25: "25-34"
    * 35: "35-44"
    * 45: "45-49"
    * 50: "50-55"
    * 56: "56+"

    A fourth column “Jobb” with the types of jobs that the users have, numbered from 0 to 20; and a fifth column “Postkode” with the respective postcodes.
3. The file “film.csv” with a first column “FilmID”; a second column “Tittel” with the title of each movie; and 18 more columns of different genres, where 1 denotes that the movie belongs to that genre and 0 that it does not.

I did not consider relevant so far the tidstempel or the postcodes.

The overview of the users from the data in “bruker.csv” is:

- Mean age: 30.8, median: 25.0, mode: ModeResult(mode=array([25], dtype=int64), count=array([71]), variance: 137.4699999999998.
- Number of women: 52, percentage of women: 26%.
- Number of men: 148, percentage of men: 74%.
- Mode of the jobs is: ModeResult(mode=array([0], dtype=int64), count=array([24]))

- Mode of the women's jobs is: ModeResult(mode=array([4], dtype=int64), count=array([1]))
- Mode of the men's jobs is: ModeResult(mode=array([20], dtype=int64), count=array([1])) *(I think there are some mistakes with these three last modes, they seem contradictory)*

The overview of the movies from "film.csv" is:

- Total number of action movies is: 70, adventure: 42, animation: 18, children's: 36, comedy: 130, crime: 31, documentary: 18, drama: 208, fantasy: 9, film-noir: 8, horror: 44, musical: 21, mystery: 24, romance: 65, Sci-Fi: 41, thriller: 62, war: 13, western: 8.
- The total number of movies is 494.
- Visualization/histogram of average ranking of movies by genre, using sns.barplot, with the x-axis the movie genres and the y-axis from 0 to 5 (stars).
- **(A)** Titles of the most popular movies, by taking from "rangering.csv" only the ones who got 5 stars and counting how many users gave 5 stars to each of them. Visualization of those ranked at least 25 times with 5 stars, using sns.barplot.
- Computation of the average rating per genre for each user.
- **(B)** Computation of the average rating for each movie, and computation of the number of times that each movie has been rated.
- **(B)** Display of a table with all the film IDs and their respective average rating and number of times that each of them has been rated.

Types of models:

1) A simple baseline recommender model which is not personalized can be the movies selected in **(A)**, so that every user get recommended the ones rated at least 25 times with 5 stars.

Result:
```
Autumn in New York (2000)
Vie est belle, La (Life is Rosey) (1987)
Defying Gravity (1997)
Ruthless People (1986)
Portraits Chinois (1996)
Defending Your Life (1991)
Omega Code, The (1999)
Hard 8 (a.k.a. Sydney, a.k.a. Hard Eight) (1996)
Mad City (1997)
National Lampoon's Senior Trip (1995)
```

2) Another non-personalized baseline model consist of recommending the movies that have an average rating of at least 4 stars and have been rated by at least 20 users, from the data obtained in **(B).**

Result:
```
Two Thousand Maniacs! (1964)
Wrong Trousers, The (1993)
Lethal Weapon 2 (1989)
```

```
Nineteen Eighty-Four (1984)
Hard Target (1993)
To Sir with Love (1967)
Mole People, The (1956)
Happiness Is in the Field (1995)
Maltese Falcon, The (1941)
Man Who Would Be King, The (1975)
Best Laid Plans (1999)
```

3) A content-based recommender model, which works with the data generated from a user.

The predictions are the ratings that each user would give to a movie that has not been previously rated, so that the movies with better predictions are recommended.

I used a logistic regression for this type of model. I am not sure if I am taking the data well for the variables X or Y, I had many problems applying “train_test_split” because the variables had different shape. For Y there are many missing values because many movies have not been rated, I would like to put it as 0, and get a specific prediction between 1 and 5 for such value Y after applying the machine learning model. *Can I get any hints about how to solve this?*

I am not sure about how to interpret the results either, otherwise I would compare them with the extensive analysis of the data I did at the beginning and check if things make sense and there is a story to tell.

Something different to what I have delivered on jupyter, where I used “train_test_split”, was the following, and I was very happy taking exactly a 70% of the data for training, but it said that the function “iloc” could not work, *any opinions about which approach is better?*:

```
X=film_df.drop(['Tittel'], axis=1).values

y=rangering_df['Rangering'].values

indices=np.random.choice(X.shape[0],int(X.shape[0]*0.7))

X_train=X.iloc[indices]

y_train=y.iloc[indices]
```

4) A collaborative recommender model, which works based on how similar different users are according to:
    a) How they rated movies (using cosine similarity).

        I develop a matrix with all the users as lines and all the movies as columns, and the value in the entrance (i,j) is the rating that the user i has given to the movie j, from 1 to 5. I fill the value of a movie not rated by a user by 0. Then I apply the function cosine_similarity and get a table with the correlation between the movies.

I am unsure about how to make recommendations in this case, any help please?

b) Their characteristics, like having similar age or same gender, independently of the ratings they gave to movies.

I haven't develop this system yet, all this data science programming is new for me and I go a bit slow understanding it and coding without errors. Any hints about this approach are also very welcome.

5) A combined recommender model.

I haven't coded this model yet, I think of getting the predicted value 'y' by the formula y=(y_content_based + y_collaborative)/2, where y_content_based is given by the first content-based model and y_collaborative by the collaborative model.