Applied Data Science
Lab 2 Report

Eklavya Sarkar

November 16, 2018

Introduction

This assignment builds on the previous one, again requiring data ingestion, wrangling, cleaning, visualisation, and reporting, but also tests our Machine Learning knowledge algorithm and ability to apply this algorithm to build a recommendation system.

Code Methodology

Libraries

Like the previous coursework, this required `Pandas` for data wrangling, `Numpy` for a few specific functions such as `np.NaN`, `Matplotlib` for data visualisation, and most importantly `sklearn.neighbors`'s `NearestNeighbors` library for implementing the k-NN algorithm.

Data Ingestion and Wrangling

I read in the data into `Pandas`'s tables from the .csv files by using its `read_csv` function, and making sure the appropriate separator value for chosen for each file as necessary, such as `sep='\t'` for TSVs. For the movies dataset, an `encoding="ISO-8859-1"` parameter was required for correct ingestion. I gave each column a relevant name.

Data Analysis and Modelling

For **task 1**, I undertook a very straightforward approach, by calculating the mean rating for each movie, and then I looped through all the movies in each genre, until finding the one with the highest average rating, which would be chosen as the best rated film for that genre. I used lists to contain these movies, so if several movies in one genre had the same mean rating, then they'd all be included as the best movies of that particular genre. Finally, I used dictionaries to store the final lists of movies of each genre. The complete list can be found in the appendix under task 1.

The key to working out **task 2** was to understand the k-NN algorithm and how it could be applied to our dataset. To find similar users to a chosen target user, one has to input a data vector to `sklearn`'s `NearestNeighbors` function. Since this task only essentially requires the users, movies, and the ratings given by each user to a selection of movies, we can model this into a `UserID` by `MovieID` matrix with each cell containing the rating. This allows us to visualise and use each row as an input vector unique to that particular user. `Sklearn` then proceeds to find the nearest neighbours by essentially doing measuring the Euclidean distance between the target user's vector and all the others, and returns the chosen amount of neighbours with the highest 'similarity', i.e. the ones with the lowest Euclidean distance.

Similarly, for **task 3**, we could improve the model by including a few more parameters, such as age, gender, occupation. The key for this task was to understand how to convert categorical data into numeric data, as the additional usable information is given as Strings. `Pandas`'s `get_dummies` function was extremely handy in this case, as it converted a single column of `M` or `F` values into 2 'dummy' columns of `0` and `1` values, for male and female respectively. This allows us to very easily add these dummy columns into our input vector and improve the accuracy of the selected neighbours. I made and included dummies for the user's age, gender, occupation, and zipcode, although the correlation between a user's choice of movies and his living area might be lower than the other factors.

Data Prediction and Optimisation

I decided to split my data, and take the first 900 users as the training set, and the 43 remaining users as the test users. Furthermore, I calculated the total number of ratings each movie received, to remove the movies which had a count below a chosen threshold (of 10), as this skewed the data and the recommendations: movies could have a high ratings (above 4.5) with only 1-10 total votes.

Till now, we have only found similar neighbours, but to actually recommend a movie to our target user, we still have a do a few more steps. We can calculate the mean of all the movies seen by one or more of the `k` selected neighbours, and use those means to predict ratings of those movies for our target user. We can then descendingly sort the newly calculated means of these movies, and take the top 10 movies highest mean ratings as recommendations, as those are most likely to be enjoyed by the target user.

I went a step further by storing and changing the value of a highly rated movie (by the target user) to 0.0, to see if our algorithm would recommend this movie. One can calculate the mean square error of our algorithm, by comparing the actual rating Y_i and predicted rating Y_i':

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - Y_i')^2 \tag{1}$$

There are several ways to further optimise and improve this algorithm. Firstly, one should change more user ratings to 0.0 and see if those movies appear in the recommendation lists, as this would improve our mean square function by virtue of having a higher n. Secondly, we could find out which value of k neighbours is the best (without over-fitting), by looking at our mean square function while incrementing k from 0 to the total possible number of neighbours. A graph of this function would also be very useful for visualisation purposes. Similarly, one could also change the value of radius, to see what effect this has on the accuracy of the recommendations. Finally, one could keep on getting more and more data of each user, to provide even more matching neighbours. Eventually, however, the algorithm remains the same, will reach a threshold, as in reality, each user is unique has his own slightly different taste.

Code Development Issues

The main difficulty in this assignment was understanding the k-NN algorithm and realising one could use a matrix to have rows of input vectors for `sklearn`'s function. One has become so used to tables (as opposed to matrices), that it required some mental gymnastics to be able to visualise the the ratings as a correlation between the movies and users. Further issues were relatively easier to solve, such 'flattening' arrays returned by the k-NN algorithm into 1D straightforward lists by using loops. Learning to think in a Pythonic (and Pandas-esque) way to calculate means, and get row's indices was another challenge, as I come from a `C`-oriented background. It was also important to not get confused between `df.loc` and `df.iloc`.

Experimentation results

My algorithm return 10 recommendations for each user, however for most of them it did not contain the particular movie I had selected and converted to a rating of 0.0. This seems to be because of the high value of k neighbours. This means the target user receives plenty of new means, as the larger k is, the more likely it is that there are movies seen by other users and not by the target user. Therefore the returned list of recommended movies contain movies with high mean ratings, which are all higher than the movie highly rated by the target user. Varying the value of k and the radius has an important impact on the recommendations, as it often alternates the order of the movies. Adding the extra factors, as done in task 3, however seems to completely change the entirety of the top 10 recommendations!

Conclusion

This was a useful exercise which forced us to develop a deeper understanding of `Pandas` and how to use them in conjunction with libraries such as `sklearn`. It also gave a real-word insight on how people's tastes on movies can be quite similar if they have basic commonalities such as age, gender and occupation.

Appendix

Data Exploration and Visualisation

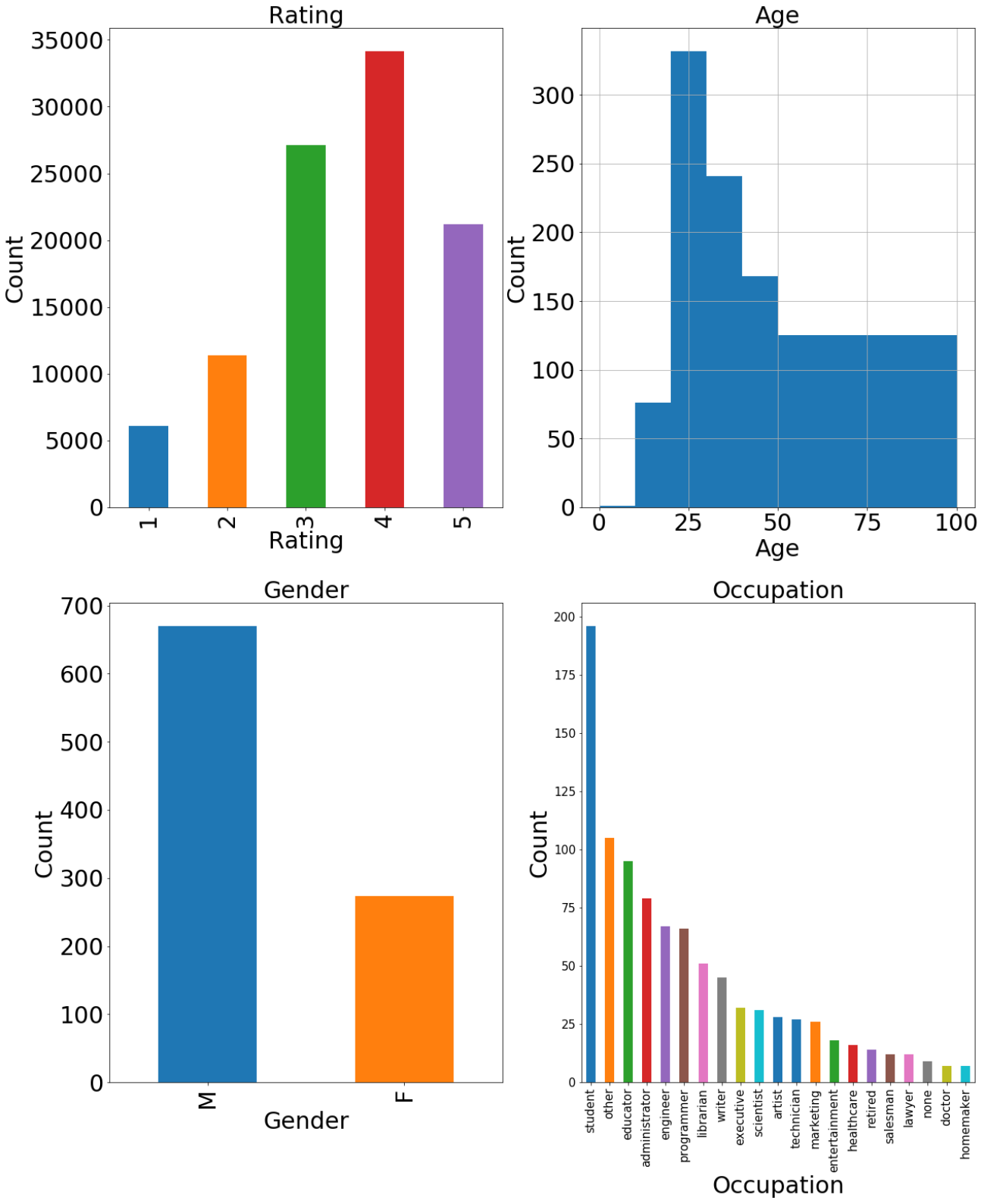

Figure 1: Data distributions

Task 1: Top rated movies of each genre

Genre	**Movie Title**
Action	Star Wars
Adventure	Star Kid
Animation	A Close Shave
Children's	Star Kid
Comedy	Santa with Muscles
Crime	They Made Me a Criminal
Documentary	Great Day in Harlem, A (1994) Marlene Dietrich: Shadow and Light (1996)
Drama	They Made Me a Criminal Prefontaine (1997) The Saint of Fort Washington Aiqing wansui Someone Else's America Entertaining Angels: The Dorothy Day Story
Fantasy	Star Kid
Film-Noir	The Manchurian Candidate
Horror	Psycho
Musical	The Wizard of Oz
Mystery	Rear Window
Romance	Casablanca
Sci-Fi	Star Kid
Thriller	A Close Shave
War	Schindler's List
Western	High Noon
uknown	unknown

Task 2: Sample movie recommendations outputs

```
For user 900 the top recommended movies are:
   Incognito (1997)
   My Man Godfrey (1936)
   Four Days in September (1997)
   Quiet Man, The (1952)
   Alphaville (1965)
   Paths of Glory (1957)
   To Live (Huozhe) (1994)
   8 1/2 (1963)
   Women, The (1939)
   Duck Soup (1933)

For user 901 the top recommended movies are:
   Stalker (1979)
   Foreign Correspondent (1940)
   Free Willy 2: The Adventure Home (1995)
   Spice World (1997)
   Meet John Doe (1941)
   Top Hat (1935)
   Wrong Trousers, The (1993)
   Casablanca (1942)
   Rear Window (1954)
   Once Upon a Time in the West (1969)

For user 902 the top recommended movies are:
   Ponette (1996)
   Incognito (1997)
   Four Days in September (1997)
   Miami Rhapsody (1995)
   Grand Day Out, A (1992)
   Wallace \& Gromit: The Best of Aardman Animation (1996)
   M (1931)
   Schindler's List (1993)
   Casablanca (1942)
   Shawshank Redemption, The (1994)

For user 903 the top recommended movies are:
   Mark of Zorro, The (1940)
   Gold Diggers: The Secret of Bear Mountain (1995)
   Shooting Fish (1997)
   Firestorm (1998)
   Body Snatcher, The (1945)
   Gang Related (1997)
   I'm Not Rappaport (1996)
   Stalker (1979)
   Incognito (1997)
   Spice World (1997)
```

Task 3: Sample improved movie recommendations outputs

```
For user 900 the top recommended movies are:
   She's the One (1996)
   Love! Valour! Compassion! (1997)
   Alien (1979)
   Don't Be a Menace to South Central While Drinking Your Juice in the Hood (1996)
   Swingers (1996)
   Until the End of the World (Bis ans Ende der Welt) (1991)
   Mars Attacks! (1996)
   Little Princess, A (1995)
   Little Women (1994)
   Frighteners, The (1996)

For user 901 the top recommended movies are:
   Vertigo (1958)
   Strange Days (1995)
   Clerks (1994)
   Sleepers (1996)
   Ben-Hur (1959)
   Bringing Up Baby (1938)
   To Catch a Thief (1955)
   Bound (1996)
   Aladdin and the King of Thieves (1996)
   Man Without a Face, The (1993)

For user 902 the top recommended movies are:
   It Happened One Night (1934)
   French Kiss (1995)
   Absolute Power (1997)
   Wedding Singer, The (1998)
   Little Women (1994)
   Much Ado About Nothing (1993)
   Big Lebowski, The (1998)
   One Flew Over the Cuckoo's Nest (1975)
   Twister (1996)
   Heat (1995)

For user 903 the top recommended movies are:
   Bad Taste (1987)
   Strange Days (1995)
   Dazed and Confused (1993)
   Dead Man (1995)
   What's Eating Gilbert Grape (1993)
   Wes Craven's New Nightmare (1994)
   Paths of Glory (1957)
   Andre (1994)
   Bullets Over Broadway (1994)
   Hudsucker Proxy, The (1994)
```