Homework -5

CSCE 633: Machine Learning

Manav Gurumoorthy

830000011

1. Introduction:

The goal of this Homework assignment was to build a machine learning model with relation to a real-world problem. This involved finding a suitable dataset, formulating a problem statement to be solved using a machine learning model and to apply some of the theory that was covered in class.

2. Motivation and Data:

The motivation for this work is from applications in entertainment. The idea was to use data collected about user generated movie ratings and create a system that could recommend new movies to a user based on the ratings of other users.

The data set used is called movielens. It was created by a research group called grouplens. GroupLens Research Project is a research group in the Department of Computer Science at the University of Minnesota.

The dataset consisted of three files, the first called Ratings, this contained the ratings that a user had assigned to a movie. It was in the following format, “UserID::MovieID::Rating::Timestamp”, where

1. UserID: was a number between 1 to 6040
2. MovieID: was a number between 1 and 3952
3. Ratings: Whole-star ratings on a scale of 5

Each user in this dataset has rated 20 movies at the least.

The second file was “users’ it consisted of some demographic information of the users in the following format UserID::Gender::Age::Occupation::Zip-code.

1. Gender is M or F for male or female.
2. Age is kept in the following ranges:

```
*  1:  "Under 18"
* 18:  "18-24"
* 25:  "25-34"
* 35:  "35-44"
* 45:  "45-49"
* 50:  "50-55"
* 56:  "56+”
```

3. Occupation is chosen from the following choices:

```
*  0:  "other" or not specified
*  1:  "academic/educator"
*  2:  "artist"
*  3:  "clerical/admin"
*  4:  "college/grad student"
*  5:  "customer service"
*  6:  "doctor/health care"
*  7:  "executive/managerial"
*  8:  "farmer"
*  9:  "homemaker"
* 10:  "K-12 student"
* 11:  "lawyer"
* 12:  "programmer"
* 13:  "retired"
* 14:  "sales/marketing"
* 15:  "scientist"
* 16:  "self-employed"
* 17:  "technician/engineer"
* 18:  "tradesman/craftsman"
* 19:  "unemployed"
```

```
    * 20:  "writer"
```

The third file was called “Movies”, and contains description of the movies, it is in the following format, “MovieID::Title::Genres”.

1. Titles are the movie titles from IMDB (with year of release).
2. Genres are pipe-separated and are selected from the following genres:

```
    * Action
    * Adventure
    * Animation
    * Children's
    * Comedy
    * Crime
    * Documentary
    * Drama
    * Fantasy
    * Film-Noir
    * Horror
    * Musical
    * Mystery
    * Romance
    * Sci-Fi
    * Thriller
    * War
    * Western
```

3. Problem Formulation

The idea that I had was to use this dataset to make a movie recommendation system. My plan was to make movie recommendations using the concept of Mode-Based Collaborative filtering. I made a mistake in this formulation as this is a kind of a semi-supervised learning algorithm, while the task was to use a supervised method. I did carry on with this initial idea, even though it doesn’t exactly fit the requirement of the problem. Model-based collaborative filtering is based on PCA. Where we use PCA’s matrix factorization to find latent attributes of the inputs while also reducing the dimensionality of the data.

I did try to reformulate this problem into a fully supervised task. Unfortunately, I was unable to do so successfully in the stipulated time.

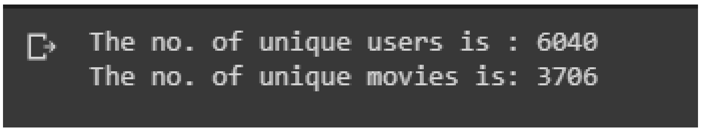
4. Data pre-processing

The data was in the ‘*.dat’ format. So, the first step was to convert the data from the 3 files into pandas data frames. Also, since the data was separated across 3 files, we merged the data into a single large dataset that contained all the information.

Also, I replaced all missing entries with ‘0’ so that the dataset was complete. I also tried to vectorise some of the features, like the genres, occupation description and the movie title using sklearn’s TfidfVectorizer to keep track of the frequency of commonly used words.

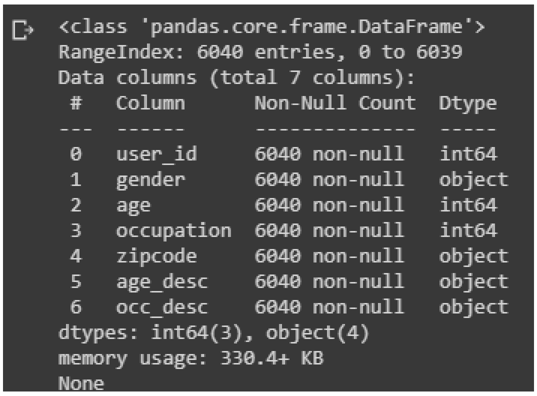
5. Data Exploration

In this part, I explored the dataset. First I we found how many unique movie and user entries were in the dataset.



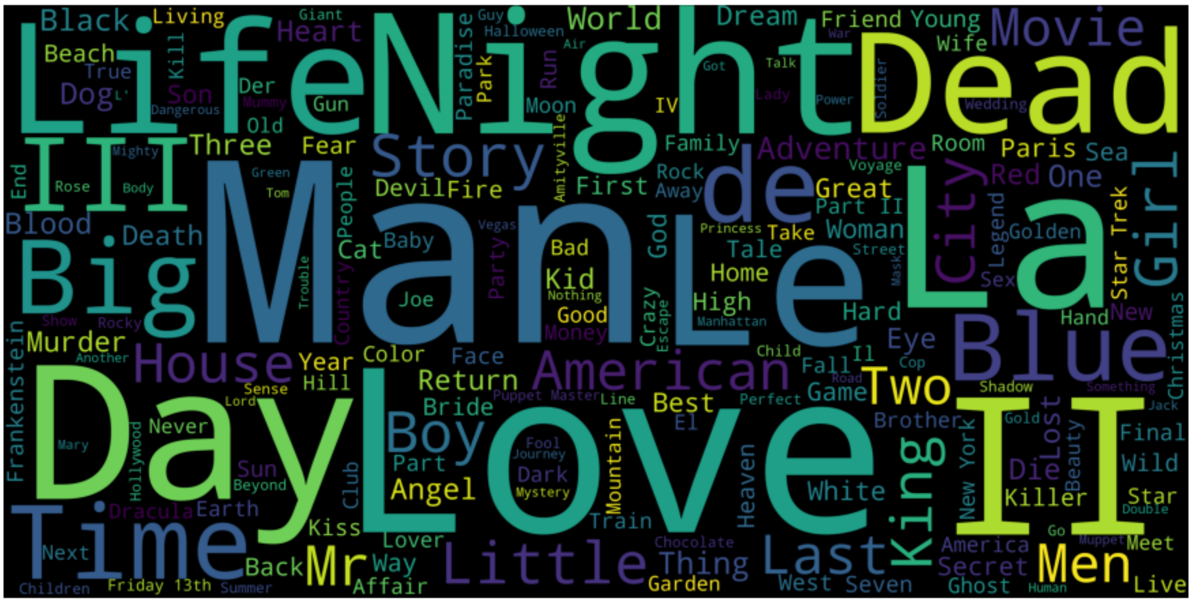
```
The no. of unique users is : 6040
The no. of unique movies is: 3706
```

Also some, statistics about the data from dataset:

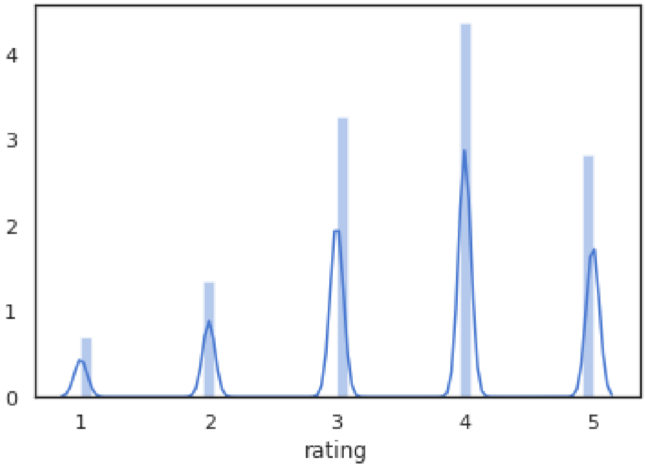


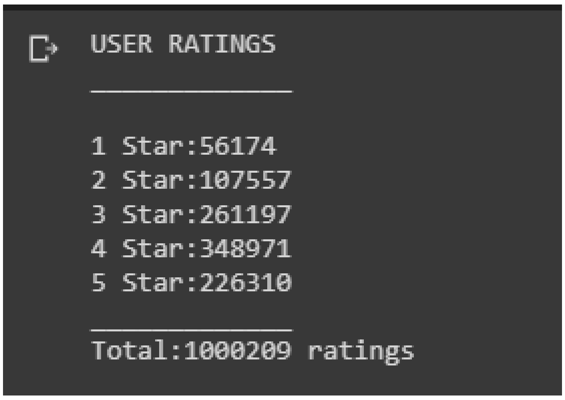
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6040 entries, 0 to 6039
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   user_id     6040 non-null   int64
 1   gender      6040 non-null   object
 2   age         6040 non-null   int64
 3   occupation  6040 non-null   int64
 4   zipcode     6040 non-null   object
 5   age_desc    6040 non-null   object
 6   occ_desc    6040 non-null   object
dtypes: int64(3), object(4)
memory usage: 330.4+ KB
None
```

To get an understanding of the frequencies of some of the terms in the dataset I used a word cloud representation. The larger the word is in the word cloud means more frequent is the word.



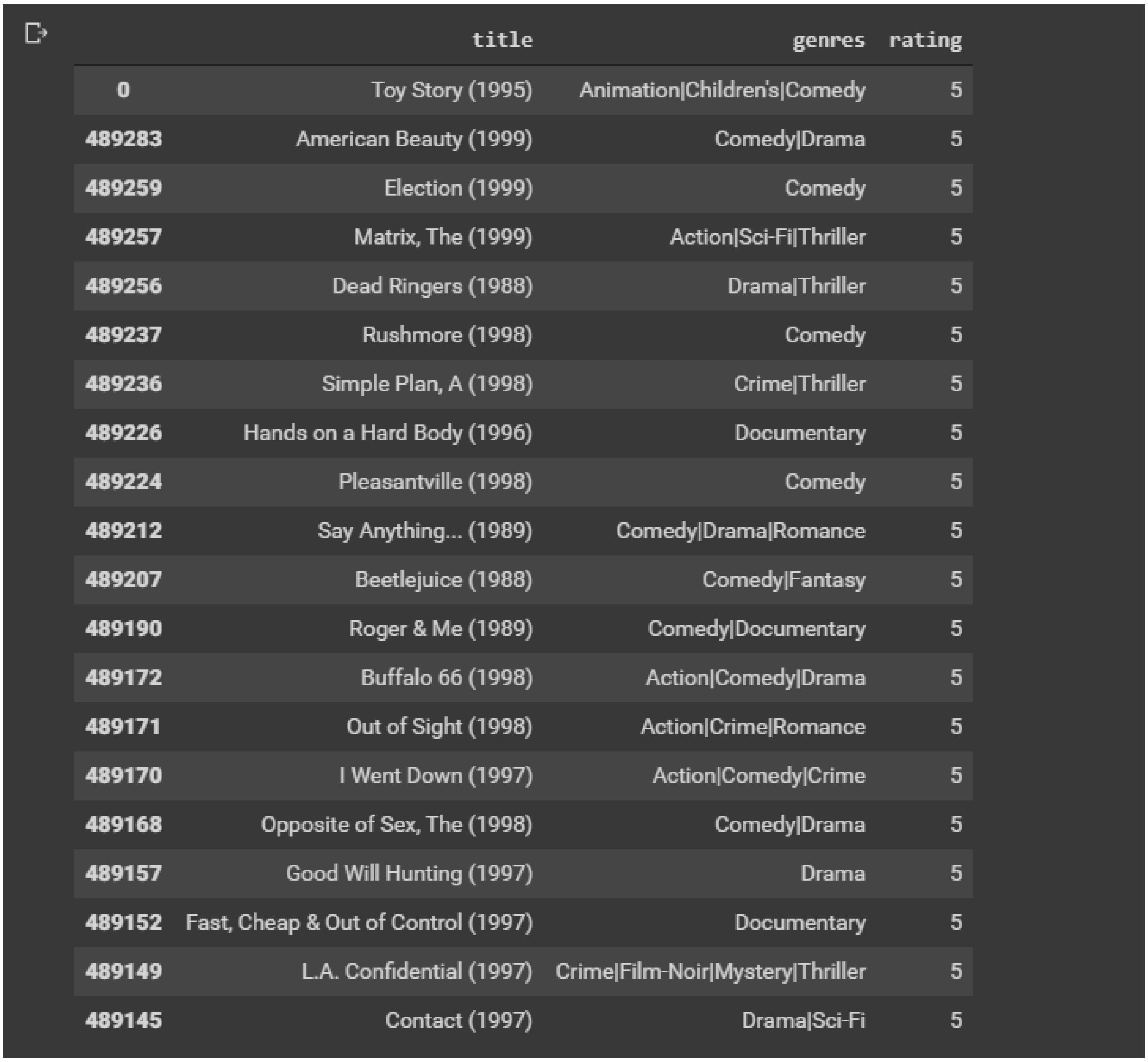
The following plot shows how the users have rated the movies. It is clear that the users are quite generous with their ratings. Because a large percentage of the movies have 4 to 5 start ratings.





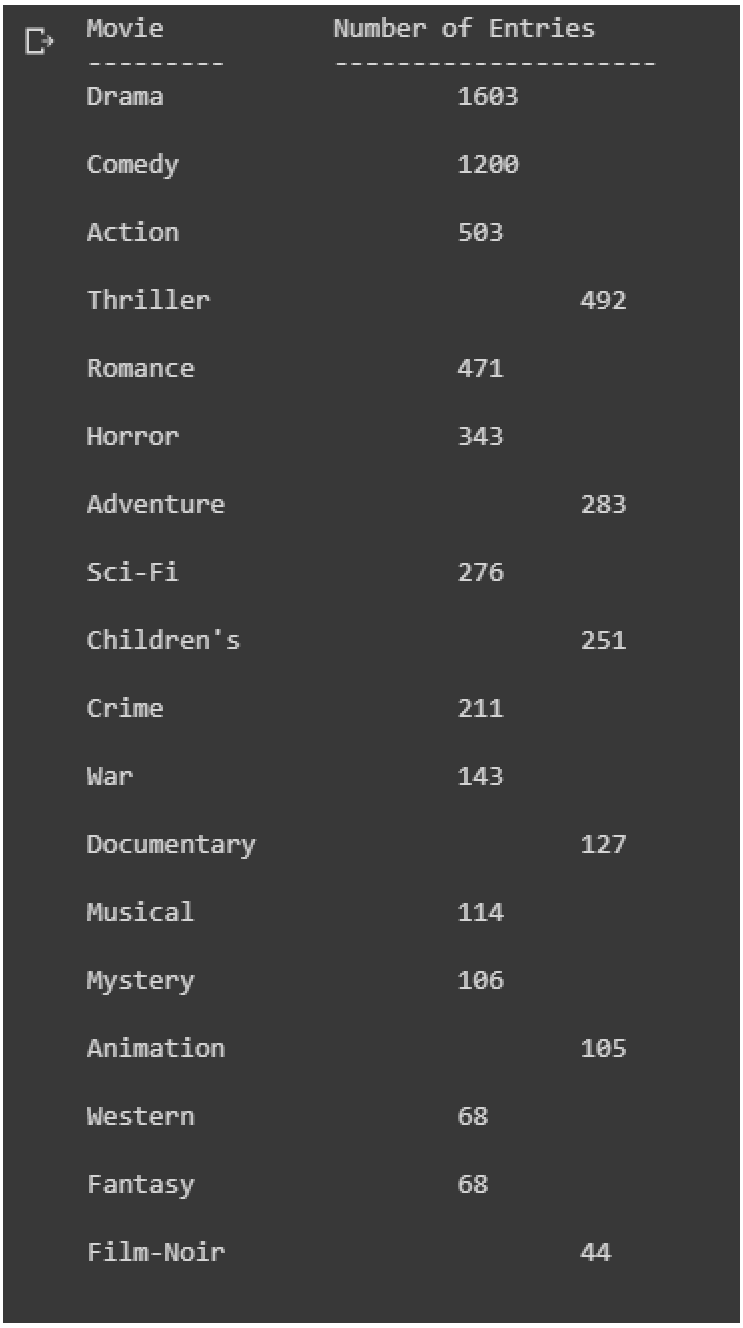
```
USER RATINGS
_____________

1 Star:56174
2 Star:107557
3 Star:261197
4 Star:348971
5 Star:226310
_____________
Total:1000209 ratings
```

I also plotted the top 20 movies that have the highest rating,



	title	genres	rating
0	Toy Story (1995)	Animation|Children's|Comedy	5
489283	American Beauty (1999)	Comedy|Drama	5
489259	Election (1999)	Comedy	5
489257	Matrix, The (1999)	Action|Sci-Fi|Thriller	5
489256	Dead Ringers (1988)	Drama|Thriller	5
489237	Rushmore (1998)	Comedy	5
489236	Simple Plan, A (1998)	Crime|Thriller	5
489226	Hands on a Hard Body (1996)	Documentary	5
489224	Pleasantville (1998)	Comedy	5
489212	Say Anything... (1989)	Comedy|Drama|Romance	5
489207	Beetlejuice (1988)	Comedy|Fantasy	5
489190	Roger & Me (1989)	Comedy|Documentary	5
489172	Buffalo 66 (1998)	Action|Comedy|Drama	5
489171	Out of Sight (1998)	Action|Crime|Romance	5
489170	I Went Down (1997)	Action|Comedy|Crime	5
489168	Opposite of Sex, The (1998)	Comedy|Drama	5
489157	Good Will Hunting (1997)	Drama	5
489152	Fast, Cheap & Out of Control (1997)	Documentary	5
489149	L.A. Confidential (1997)	Crime|Film-Noir|Mystery|Thriller	5
489145	Contact (1997)	Drama|Sci-Fi	5

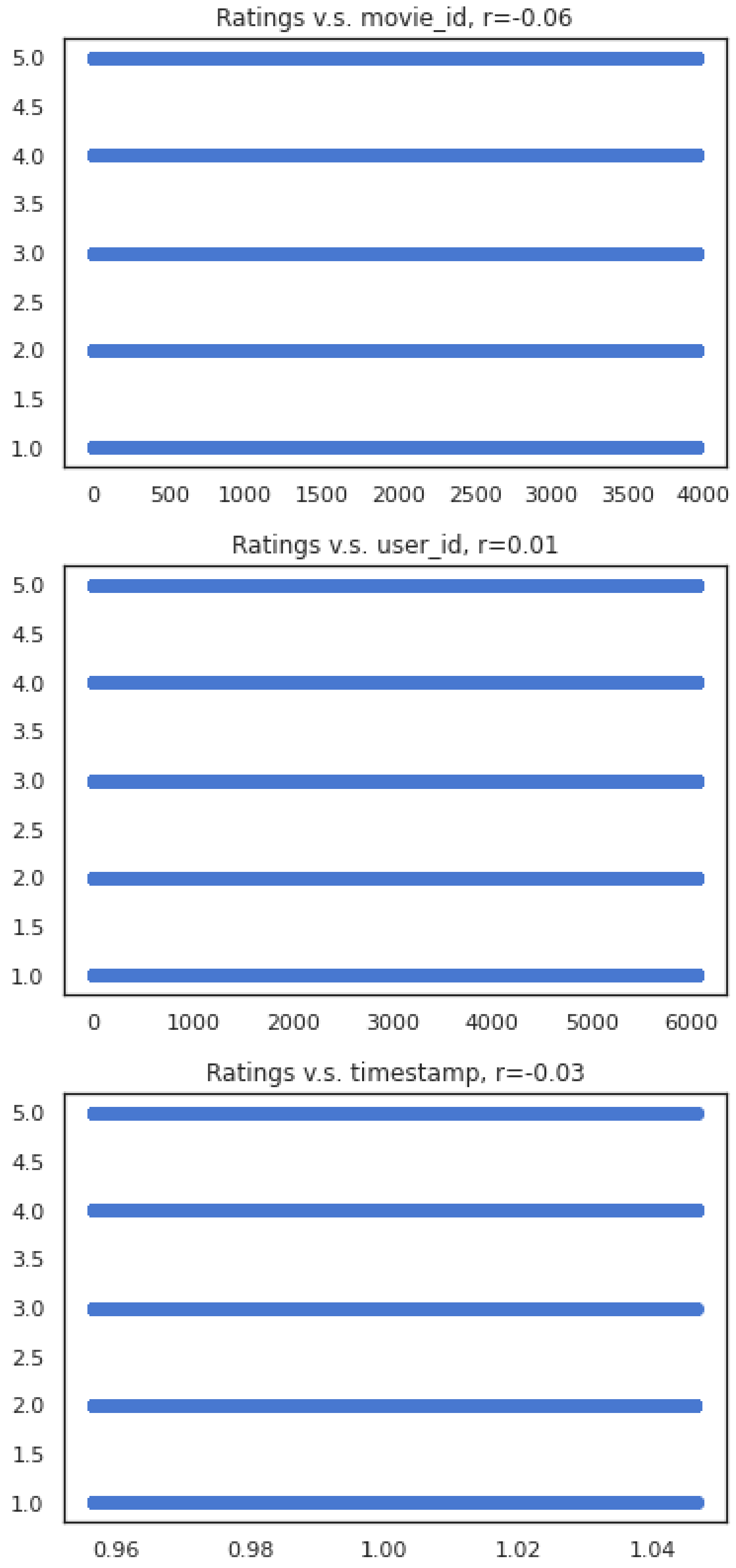
Also, this was the distribution of the movies based on the various genres.



```
Movie           Number of Entries
---------       ---------------------
Drama                   1603

Comedy                  1200

Action                  503

Thriller                        492

Romance                 471

Horror                  343

Adventure                       283

Sci-Fi                  276

Children's                      251

Crime                   211

War                     143

Documentary                     127

Musical                 114

Mystery                 106

Animation                       105

Western                 68

Fantasy                 68

Film-Noir                       44
```

6. Feature Selection

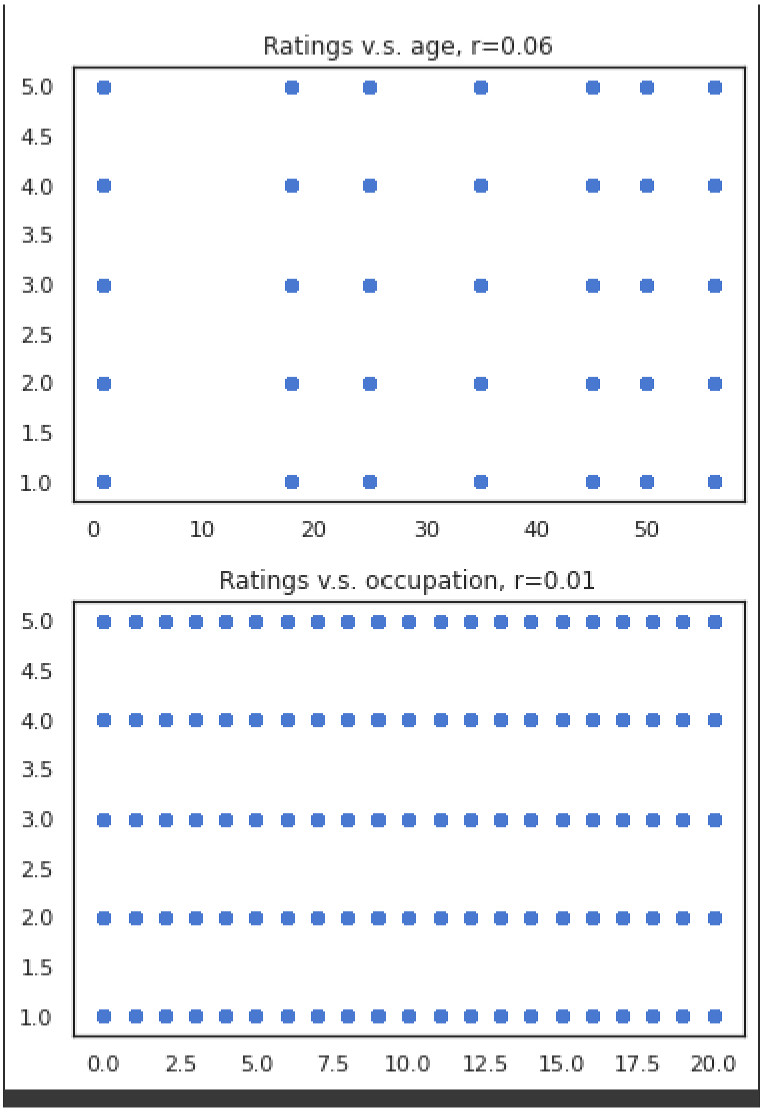
Used Pearson’s r to find the correlation between the features with the target and the correlation between the features themselves. As it is visible from below there were no real correlations between the data and the selected target. Which showed me further that this task may not be suitable for a supervised learning algorithm.



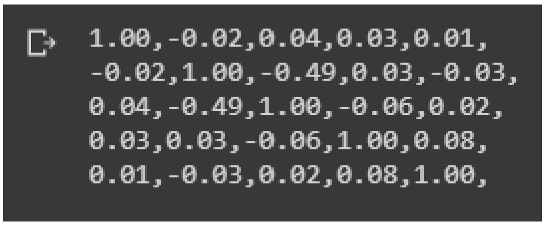
Ratings v.s. movie_id, r=-0.06

Ratings v.s. user_id, r=0.01

Ratings v.s. timestamp, r=-0.03



The correlation numbers between the features:



```
1.00,-0.02,0.04,0.03,0.01,
-0.02,1.00,-0.49,0.03,-0.03,
0.04,-0.49,1.00,-0.06,0.02,
0.03,0.03,-0.06,1.00,0.08,
0.01,-0.03,0.02,0.08,1.00,
```

7. Feature Transformation

I have used Principal Component Analysis in this section to find latent associations in the data and also to reduce the dimensionality of the data. In this portion I created a matrix between user Id and movie Id with each entry representing the rating given by that user for that movie.

Using the reconstructed matrix I make predictions for a user based on how well the movie was rated by other users.

From the dataset I find the movies that the target user has not watched and use that information to recommend the highest rated movie that the user hasn’t seen. While this model doesn’t use any other information you can see that it does make some good prediction. For example, in one case, a user had liked and seen Star Wars Episode 1 and the system recommended Episode 3 and Episode 6 to the user. Now qualitatively this shows that the system is “working” as expected.

Here is where I was faced with another challenge, for the evaluation of the model, since this was basically an unsupervised model, I was unable to come up with any powerful quantitative measures of the system performance.

Luckily, I did find a library called Surprise online that was built to test such recommendation systems, using the surprise model, I found the model to have a Root Mean Square Error to be 0.87 over a 5 fold cross validation.

The details of this can be found in Appendix A as a part of the code.

8. Conclusion

The purpose of this homework was to go through the entire process of building a machine learning model to solve a real-world problem. In this I looked at creating a movie recommendation system, very similar to the algorithms that run on YouTube and Netflix. While I may have not exactly solved all the requirements of the homework, I have learned a lot about the concepts involved in this area of work. I feel this work can be extended to make more holistic recommendation by taking the demographic information of the users into consideration. There may also be better methods in order to create such a recommendation system.

APPENDIX A
COLAB NOTEBOOK