A Movie Recommender System - Report

Problem Statement:

The objective of this project is to build a movie recommender system. We are building a movie recommendation system to help recommend movies on interests to the customers based on how they have rated the movies they have watched in our data set.

Dataset Description:

For this project, I wanted to find a data set which provided us with plenty of observations and multiple variables. After a lot of research, we came across a data set from MovieLens. This data set fit the criteria our criteria of providing us with 100,000+ observations, and 6 variables. This data set was collected over a period of time, you can notice the movies that are rated in this data set range from the mid 1900's to early 2000's, providing us with a variety of movies.

Data Cleaning:

The data is in two csv files. The data cleaning steps were as follows:

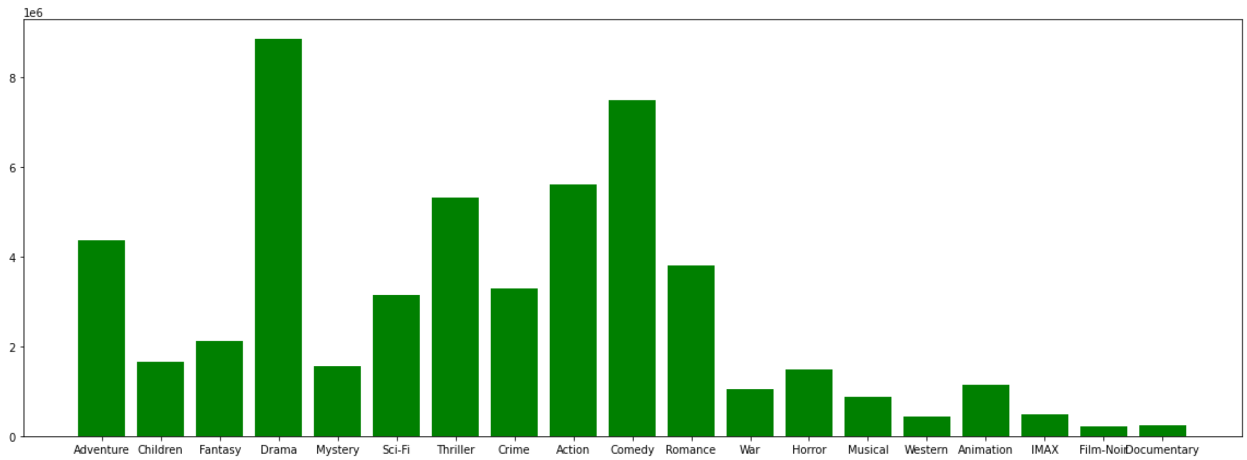
Downloaded the data to a local hard drive

I read the ratings csv to pandas then the read the movies csv to pandas next, I merged them on the 'movieId' column.

	userId	movieId	rating	timestamp	title	genres
0	1	2	3.5	1112486027	Jumanji (1995)	Adventure|Children|Fantasy
1	5	2	3.0	851527569	Jumanji (1995)	Adventure|Children|Fantasy
2	13	2	3.0	849082742	Jumanji (1995)	Adventure|Children|Fantasy
3	29	2	3.0	835562174	Jumanji (1995)	Adventure|Children|Fantasy
4	34	2	3.0	846509384	Jumanji (1995)	Adventure|Children|Fantasy
5	54	2	3.0	974918176	Jumanji (1995)	Adventure|Children|Fantasy
6	88	2	1.0	1098277938	Jumanji (1995)	Adventure|Children|Fantasy
7	91	2	3.5	1112061358	Jumanji (1995)	Adventure|Children|Fantasy
8	116	2	2.0	1132728068	Jumanji (1995)	Adventure|Children|Fantasy
9	119	2	4.0	845110667	Jumanji (1995)	Adventure|Children|Fantasy

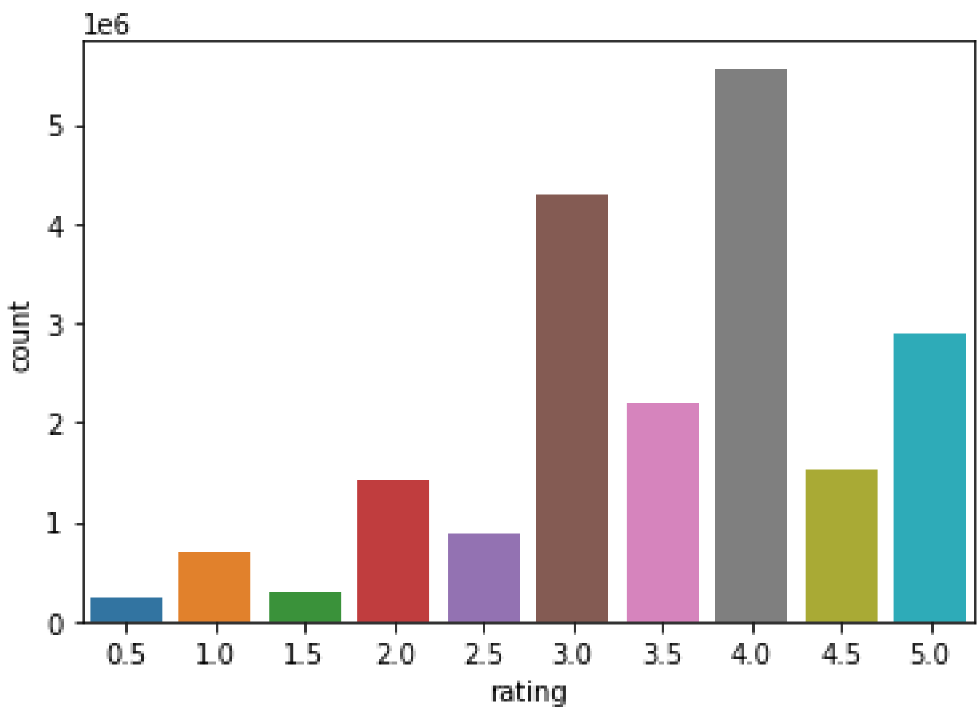
Next, I changed the data types for the columns to facilitate use in my future models.

The by genre breakdown of the movies show a clear pattern of movies that are rated:



The data was very clean and very little missing data.

I normalized the ratings.



Findings from Exploratory Data Analysis:

The data set is very large there are more than 2 million movie rating with a normalized average of 3.53 with each person reviewing on average 144 movies.

To facilitate ease of calculations in the notebook I lowered the sample size to 1200 ratings.

Building the Singular Value Decomposition:

I format of our ratings matrix to be one row per user and one column per movie using pivot ratings and made that a new variable. Then I normalized each entry by the users mean then converted that to a numpy array resulting in a sparsity level of 92.2%

I used the Scipy SVDS function because it let's me choose how many latent factors, I want to use to approximate the original ratings matrix.

I now have everything I need to make movie ratings predictions for every user. I can do it all at once by following the math and matrix multiply U, Sum, and V^T back to get the rank k = 100.

Next, I needed to add the user means back to get the actual star ratings prediction.

With the prediction matrix for every user, we can build a function to recommend movies for any user. This returns the list of movies the user has already rated.

Recommendation Function:

Now I write a function to return the movies with the highest predicted rating that the specified user has not already rated. Though I did not use any explicit movie content features, I will merge in that information to get a more complete picture of the recommendations. Here is the flow of the function get and sort the predictions using the SVD constructed matrix. Next I get the user data via UserId for which you want to predict the top rated movies, merged with the Matrix for movie data, the final return will be the top movies recommendations that have not yet been watched.

The recommendation function was used on user 810.

10 Top movies for user 810

	userId	movieId	rating	timestamp	title	genres
48	810	1639	5.0	993238226	Chasing Amy (1997)	Comedy|Drama|Romance
53	810	1799	5.0	993238295	Suicide Kings (1997)	Comedy|Crime|Drama|Mystery|
75	810	2836	5.0	993237971	Outside Providence (1999)	Comedy
32	810	1246	5.0	993238643	Dead Poets Society (1989)	Drama
55	810	1961	5.0	993238699	Rain Man (1988)	Drama
43	810	1396	5.0	993238113	Sneakers (1992)	Action|Crime|Drama|Sci-Fi
44	810	1500	5.0	993238500	Grosse Pointe Blank (1997)	Crime|Romance

87	810	4041	5.0	993238873	Officer and a Gentleman, An (1982)	Romance
96	810	4308	5.0	993239018	Moulin Rouge (2001)	Drama|Musical|Romance
46	810	1617	5.0	993238310	L.A. Confidential (1997)	Crime|Film-Noir|Thriller

Movie Recommendations for User 810

	movieId	title	genres
4797	4993	Lord of the Rings: The Fellowship of the Ring,...	Adventure|Fantasy
5753	5952	Lord of the Rings: The Two Towers, The (2002)	Adventure|Fantasy
6941	7153	Lord of the Rings: The Return of the King, The...	Action|Adventure|Drama|Fantasy
3402	3578	Gladiator (2000)	Action|Adventure|Drama
2417	2571	Matrix, The (1999)	Action|Sci-Fi|Thriller
4767	4963	Ocean's Eleven (2001)	Crime|Thriller
4799	4995	Beautiful Mind, A (2001)	Drama|Romance
345	356	Forrest Gump (1994)	Comedy|Drama|Romance|War
6329	6539	Pirates of the Caribbean: The Curse of the Bla...	Action|Adventure|Comedy|Fantasy
5221	5418	Bourne Identity, The (2002)	Action|Mystery|Thriller

Error Checking the Recommendation System:

Now that I have a function to make a movie recommendation system, I wanted to test the predicted movie rating compared to the actual movie rating rated. I will use a similarity matrix, a pairwise distance calculation using the correlation computation do determine the predicted rating for the training and test set. Then compare that predicted number to the actual number then measure the Root Mean Square Error.

Conclusion:

Overall, this recommendation system works. The results according to the error check function have a relatively high root mean square error. However, the system seems to be effective.

More Work to be done:

The process of creating a recommendation system was success this data was clean, so the process was not hard. There is still room to evaluate what movies someone would like even if they have never rated a movie. I think that recommendation system to would be the most useful for service providers.