Recommendation System

The goal of this project is to get a better understanding of recommendation system works. The algorithms and methods used to solve the business problem will be explained in detail as well as the business context.

1. What is recommendation system?

A recommendation system is a tool used to recommend items to users based on their interests. With this, is expected to engage the customer through the personalized service offered and also to improve business metrics.

2. What is the business problem?

Data is a only a tool that can help to solve a business problem, but it is a valuable asset. The focus of the business is to increase *revenue*, and therefore by offering personalized services brands increase their conversion rate, as we can see by past examples like Amazon or Netflix.
On the other hand, customer satisfaction and customer *loyalty* is something very important to the businesses. Allowing the customer to check – recommendation, his/her last products visits we are already paying attention to their behavior. By going one step above and offering them products and services that are of his/her interest we definitely are contributing to higher customer retention.
Recommendation system as it says it recommending something to someone. The first recommendation system known is word of mouth of *like-minded people*. In other words, by offering personalized services based on like-minded users we can expect a better user experience.
One other goal is discovery. Triggering different interest on customer can lead to *unexpected* behavior, hopefully the customer enjoys it at the end of the day.

1. Data

The information is gathered in three datasets: users, movies and ratings.

Dataset : users
Columns : userId, gender, age, job & zip code.
Modifications :

- userId range : 0, 1, …, 6039
- New column ageT – indicates the age range
- New column jobT – indicates job

	userId	gender	age	ageT	job	jobT	zipcode
0	0	F	1	Under 18	10	K-12 student	48067
1	1	M	56	56+	16	self-employed	70072
2	2	M	25	25-34	15	scientist	55117
3	3	M	45	45-49	7	executive/managerial	02460
4	4	M	25	25-34	20	writer	55455
...	...	...	...	...	...	...	...

Fig 1. *users* dataset

Dataset : movies
Columns : movieId, title, & genres
Modifications :

- Through hot encoding, add columns related w/ the different genres.
- New column nRatings – indicating the number of ratings each movie has.
- Sort the dataset using the column nRatings in descending order.
- New column movId – indicating the new movie Id.
  The range is 0, 1 , …, 3882 .

	movieId	title	genres
0	1	Toy Story (1995)	Animation|Children's|Comedy
1	2	Jumanji (1995)	Adventure|Children's|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama
4	5	Father of the Bride Part II (1995)	Comedy
...	...	...	...

Fig 2. *movies* dataset before changes

	movieId	movId	title	nRatings	genres	Action	Adventure	Animation	Children's	Comedy	...
0	2858	0	American Beauty (1999)	3428	Comedy|Drama	0.0	0.0	0.0	0.0	1.0	...
1	260	1	Star Wars: Episode IV - A New Hope (1977)	2991	Action|Adventure|Fantasy|Sci-Fi	1.0	1.0	0.0	0.0	0.0	...
2	1196	2	Star Wars: Episode V - The Empire Strikes Back...	2990	Action|Adventure|Drama|Sci-Fi|War	1.0	1.0	0.0	0.0	0.0	...
3	1210	3	Star Wars: Episode VI - Return of the Jedi (1983)	2883	Action|Adventure|Romance|Sci-Fi|War	1.0	1.0	0.0	0.0	0.0	...
4	480	4	Jurassic Park (1993)	2672	Action|Adventure|Sci-Fi	1.0	1.0	0.0	0.0	0.0	...
...	...	...	...	...	...	...	...	...	...	...	...

Fig3. *movies* dataset after changes

Dataset : ratings
Columns : userId, movieId, rating, timestamp
Modifications : (new dataset named ***ratings_v1***)

- New column movId – merged with movies dataset on movieId.
- Format column timestamp to yyyy/mm/dd hh:mm:ss
- Filter out movies with a few ratings using nRatings_min from movies dataset

	userId	movieId	rating	timestamp
0	1	1193	5	978300760
1	1	661	3	978302109
2	1	914	3	978301968
3	1	3408	4	978300275
4	1	2355	5	978824291
...	...	...	...	...

Fig4. *ratings* dataset

2. Exploratory Data Analysis

Num. of distinct movies: 3 883
Num. of rated movies: 3 706 (~ 95.44 %)

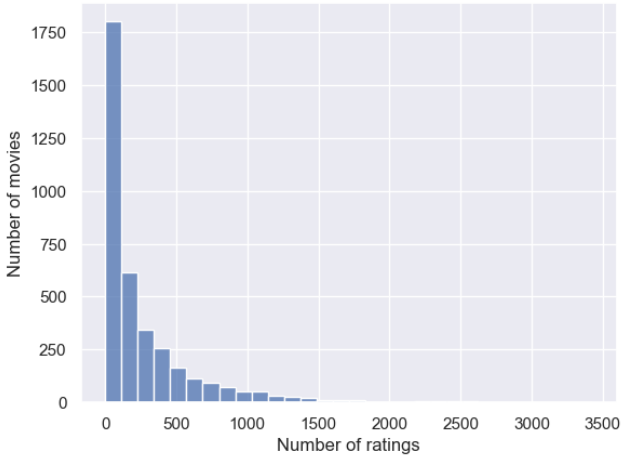
Num. of distinct users: 6 040
Num. of users that rated: 6 040 (100 %)

** Every user rated at least 20 movies, but not all movies were rated.

There are 1 000 209 ratings across 6 040 users and 3 706 movies. Each user did not rate the same movie multiple times. The possible ratings values are 1, 2, 3, 4 & 5.

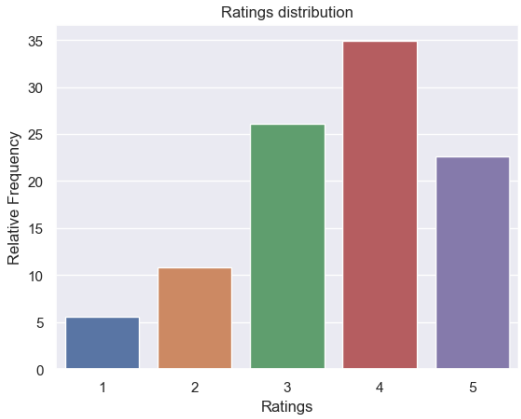
On average, each user rated 166 distinct movies , but at least 50% of the users rated no more than 96 movies. This is due to the fact that there are users that rated more than 2 300 movies.

On average, each movie was rated 270 times , but at least 50% of the movies were rated no more than 124 times. This is due to the fact that there are movies with more than 3 400 ratings.



The ratings distribution is left skewed. This suggests that the cold start problem might be present here, meaning that there is a big enough group of movies with only a few ratings and the collaborative might not be able to recommend some of these movies.

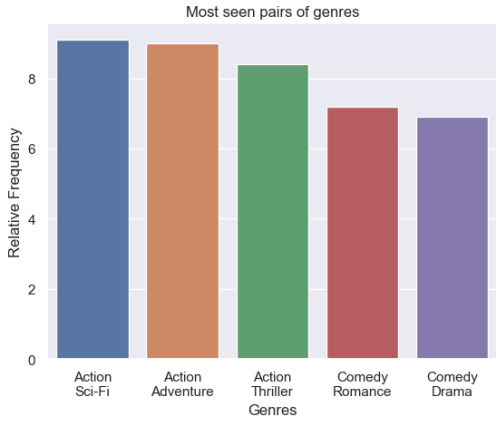
The majority of the ratings, 57.52%, are evaluated as 4 or 5 with only 5.62% of ratings being evaluated as lowest possible rating, 1.



Considering the fact that a movie can be classified under multiple genres, some of the most and least seen categories are

- Comedy : 35.7 %
- Drama : 35.4 %
- Action : 25.7 %
- …
- …
- …
- Western : 2.1%
- Film Noir : 1.8%
- Documentary : 0.8%

And top 5 combinations of genres most seen by the users are



Most popular movies are (*Number of ratings* & ***average rating***)

-	American Beauty (1999)	3 428	**4.3**
-	Star Wars Ep. 4 – A New Hope (1977)	2 991	**4.5**
-	Star Wars Ep. 5 – The Empire Strikes Back (1980)	2 990	**4.3**
-	Star Wars Ep. 6 – Return of the Jedi (1983)	2 883	**4.0**
-	Jurassic Park (1993)	2 672	**3.8**

As a small exercise to implement user-based collaborative filtering, it was regenerated association rules with the goal of identifying the most correlated movies. Only movies with ratings 4 or higher were selected, FPGrowth filter out item-sets of movies with support of 10% or less and rules presented have a confidence of at least 70%.

Movies

M1: L.A. Confidential (1997)

M2: The Godfather (1972)

M3: American Beauty (1999)

M4: Fargo (1996)

M5: Pulp Fiction (1994)

M5: Raiders of the Lost Ark (1981)

M7: Indiana Jones and the Last Crusade (1989)

M8: The Godfather: Part II (1974)

M9: Star Wars: Episode IV – A New Hope (1977)

M10: Star Wars: Episode V – The Empire Strikes Back (1980)

Antec.	Conseq.	Antec. Supp.	Conseq. Supp	Supp.	Conf.	Lift	Leverage	Conviction
M1, M2	**M3**	0.16	0.47	0.12	0.71	1.50	0.04	1.82
M1, M4	**M5**	0.19	0.29	0.14	0.72	2.44	0.08	2.49
M7	**M5**	0.21	0.37	0.19	0.88	2.36	0.11	5.41
M8, M5	**M9, M10**	0.14	0.32	0.11	0.74	2.28	0.06	2.61

Apart from movies sequels, there are correlated movies as it can be seen from the table above. For instance if taken the first association rule, 70% of times that a user watches L.A. Confidential and The Godfather, he/she also watches American Beauty and event is 1.5 times more likely than a user watches both L.A. Confidential and The Godfather, according to the lift. The leverage, measures the independence of both events, i.e., watching L.A. Confidential and The Godfather and watching American beauty. The leverage varies between -1 and 1, and it is preferable values near 1. In this case, it suggests that those two events are independent which goes against what was said before, but this can be justified by the fact that the occurrence of the events are small, especially watching L.A. Confidential and The Godfather (with a rating of 4 or higher) .

Another relevant exercise is grouping the users based on their interests.

Cluster 0

Cluster size	1183 (19.6 %) users		
Relevant Genres	Comedy	Drama	Action
	18.4 %	12.3 %	11.8 %
Gender	Male	Female	
	73.0 %	27.0%	
Age	25-34	18-24	35-44
	37.7%	24.1%	18.5%
Job	Coll. Stud.	NS	Technician/Engineer
	15.6%	10.9%	9.6%
Mean rating	3.55		
Movies	Star Wars – Episode IV, VI, V / Back to the Future / Toy Story		

Cluster 1

Cluster size	1309 (21.7 %) users		
Relevant Genres	Drama	Thriller	Action
	21.1 %	13.14 %	12.13 %
Gender	Male	Female	
	80.7 %	19.3%	
Age	25-34	35-44	18-24
	36.7%	18.2%	15.7%
Job	Exec./Mgr.	Coll. Stud.	NS
	14.4%	12.4%	11.4%
Mean rating	3.55		
Movies	American Beauty / L.A. Confidential / The Silence of the Lambs / Fargo / Saving Private Ryan		

Cluster 2

Cluster size	1016 (16.8 %) users		
Relevant Genres	Action	Sci-Fi	Thriller
	22.1 %	14.8 %	11.3 %
Gender	Male	Female	
	89.5 %	10.5 %	
Age	25-34	35-44	18-24
	34.4 %	21.1 %	19.3 %
Job	Tech./Eng.	Exec./Mgr.	Coll. Stud.
	14.6 %	12.6 %	12.0 %
Mean rating	3.60		
Movies	Terminator 2: Judgment Day / Star Wars: Episode IV, V, VI / The Matrix		

Cluster 3

Cluster size	1016 (16.8 %) users		
Relevant Genres	Drama	Comedy	Romance
	24.9 %	21.2 %	10.4 %
Gender	Male	Female	
	58.3 %	41.7 %	
Age	25-34	35-44	18-24
	34.7 %	22.0 %	17.0 %
Job	NS	Coll. Stud.	Academic/Educator
	12.5 %	12.0 %	11.1 %
Mean rating	3.62		
Movies	American Beauty / Shakespeare in Love / Fargo / Being John Malkovich / Groundhog Day		

Cluster 4

Cluster size	631 (10.4 %) users		
Relevant Genres	Drama	Comedy	Romance
	40.7 %	12.3 %	7.9 %
Gender	Male	Female	
	58.3 %	41.7 %	
Age	25-34	35-44	56 +
	28.4 %	18.9 %	15.5 %
Job	Acad./Ed.	NS	Exec./Mgr.
	14.6 %	11.1 %	9.7 %
Mean rating	3.67		
Movies	American Beauty / Schindler's List / Shawshank Redemption / The Silence of the Lambs / Fargo		

Cluster 5

Cluster size	456 (7.5 %) users		
Relevant Genres	Comedy	Drama	Romance
	41.3 %	14.5 %	10.4 %
Gender	Male	Female	
	59.6 %	40.4 %	
Age	25-34	18-24	35-44
	32.2 %	20.2 %	18.2 %
Job	NS	Coll. Stud.	Executive/Managerial
	13.8 %	13.2 %	10.1 %
Mean rating	3.59		
Movies	American Beauty / Groundhog Day / Shakespeare in Love / Being John Malkovich / Toy Story		

Cluster 6

Cluster size	101 (1.7 %) users		
Relevant Genres	Horror	Thriller	Comedy
	33.6 %	11.7 %	10.4 %
Gender	Male	Female	
	77.2 %	22.8 %	
Age	25-34	18-24	35-44
	27.7 %	26.7 %	22.8 %
Job	NS	Coll. Stud.	Exec./Mgr.
	20.8 %	10.9 %	8.9 %
Mean rating	3.45		
Movies	Alien / Scream / Jaws / Psycho / The Shining		

The users in cluster 5 are essentially fans of comedy movies and this group of users can be described as 60 % men and 52.2 % is between 18 and 34 years old. Some of the most seen movies are American Beauty, Shakespeare in Love or Toy Story. The same analysis can be extended for other clusters, for example the cluster 6 is composed by users that mainly like horror and thriller movies and almost 80 % are men and the majority is between 18-34 years old. This cluster has way more male users than the previous one, suggesting that women included in the data do not like horror and thriller movies that much. In fact, women are more prone to like drama, comedy and romance movies according to clusters 3 and 4.

3. Collaborative Filtering

The first model applied was collaborative filtering. Collaborative filtering is based on the fact that similar users display similar patterns of rating behavior and similar items receive similar ratings. There are two types of neighborhood-based algorithms: user-based and item-based. Focusing on the first which was the one applied, the method is described as

> The ratings provided by similar users to a target user A are used to make recommendations for A. The predicted ratings of A are computed as the weighted average values of these "peer group" ratings for each item.

With this in mind, the goal is then to suggest three to five movies that are relevant to the user. However, during this process we might face some issues, such as, some users might only have rated a few movies, making it very hard to belong to some neighborhood because for a user to be included in a neighborhood he/she must have a minimum number of common rated movies to all the other users, and also some movies might have been rated a couple of times as figure 5 suggest – long tail distribution. The fact that there is a restricted selection of movies that can be considered popular might not be of our interest, not only because the algorithm will be more prone to suggest these movies but also because studies suggest that items in the long tail might be more profitable. In general, the sparsity in the data can lead to recommendations with limited coverage.

Some of the advantages are

- The suggested movies for each user come from a neighborhood of users who shares similar tastes
- The recommendations are stable with the addition of new items and users
- Easy to understand and implement

Note that the previous clustering exercise can be used to determine the neighborhoods of users with similar tastes, however the recommendations might not be as personalized as the following algorithm.

3.1 Computing Similarity Matrix

Consider the ratings matrix R with size *m x n*, containing *m* users and *n* movies.

The next step is computing the similarity among users using a matrix S. In this phase, the similarity between two users is computed if the users have enough overlapping movie ratings. For this it is considered a parameter *mncmr* , which is the minimum number of common rated movies and the formula to compute the similarity is

$$\mathrm{msd}(u, v) = \frac{1}{|I_{uv}|} \cdot \sum_{i \in I_{uv}} (r_{ui} - r_{vi})^2$$

$$\mathrm{msd\_sim}(u, v) = \frac{1}{\mathrm{msd}(u, v) + 1}$$

Iuv is the set of movies that were rated by the user *u* and *v*, and |*Iuv*| >= *mncmr*.

Through (1) it is guaranteed that users must have similar ratings to be in the same neighborhood, which does not happen when using cosine similarity formula. Then it is used (2) to make sure that the similarity between 2 users is a metric in [0,1].

3.2 Computing Neighborhoods

For each user is computed a neighborhood, i.e., the most similar users to the user *u* are considered to neighborhood of the user *u*, *Nu*. Each neighborhood must satisfy two properties

- The similarity between each user in *Nu* and the owner of the *Nu*, the user *u*, must at least *simMin*
- *Nu* must have at least *nnei* neighbors

If this two conditions are not satisfied for some user then no neighborhood is computed and so there is no recommendations being made for this user. With this conditions it is expected to provide more accurate recommendations once the similarity between the user u and the its neighbors is being restricted and by also including the a minimum number of users it is expected to have more robust predictions but also a wider range movies to suggest.

3.3 Computing the movies to predict

To kown which movies it is expected to recommend to the user u, we just gather all the movies that were seen by the neighborhood and remove the ones that were seen by the user u.

3.4 Predictions

The predictions is then computed as a weighted average of the movie ratings. Only the users in the neighborhood that have ratings for movie j, will contribute to this prediction. Note that the prediction is based on mean-centered rating of a movie, svj, to decrease the potencial bias in the ratings. The mean rating of the target user is then added back to provide a raw rating prediction r^uj of the target user u and movie j.

$$\hat{r}_{uj} = \mu_u + \frac{\sum_{v \in P_u(j)} \mathrm{Sim}(u,v) \cdot s_{vj}}{\sum_{v \in P_u(j)} |\mathrm{Sim}(u,v)|} = \mu_u + \frac{\sum_{v \in P_u(j)} \mathrm{Sim}(u,v) \cdot (r_{vj} - \mu_v)}{\sum_{v \in P_u(j)} |\mathrm{Sim}(u,v)|}$$

Results

It was performed grid search to look for the best set of parameters in terms of the normalized MAE. The parameters used were:

- Number of common movies between two users, *mncmr*
- Minimum similarity between two users, *minSim*
- Size of each neighborhood, *nnei*

The results are displayed the figure 8. From the following figures one can deduce that maybe there are a different set of parameters that can improve NMAE.

However, as NMAE decreases and so as the accuracy increases also the number of users who receive recommendations also decrease. This is because for each neighborhood it is including more and more similar neighbors to the target-user.

This causes very personalized recommendations for <=50% of the users and leaves the other users without any recommendation. The users that receive recommendations are the most active ones and the ones we have more historical data.

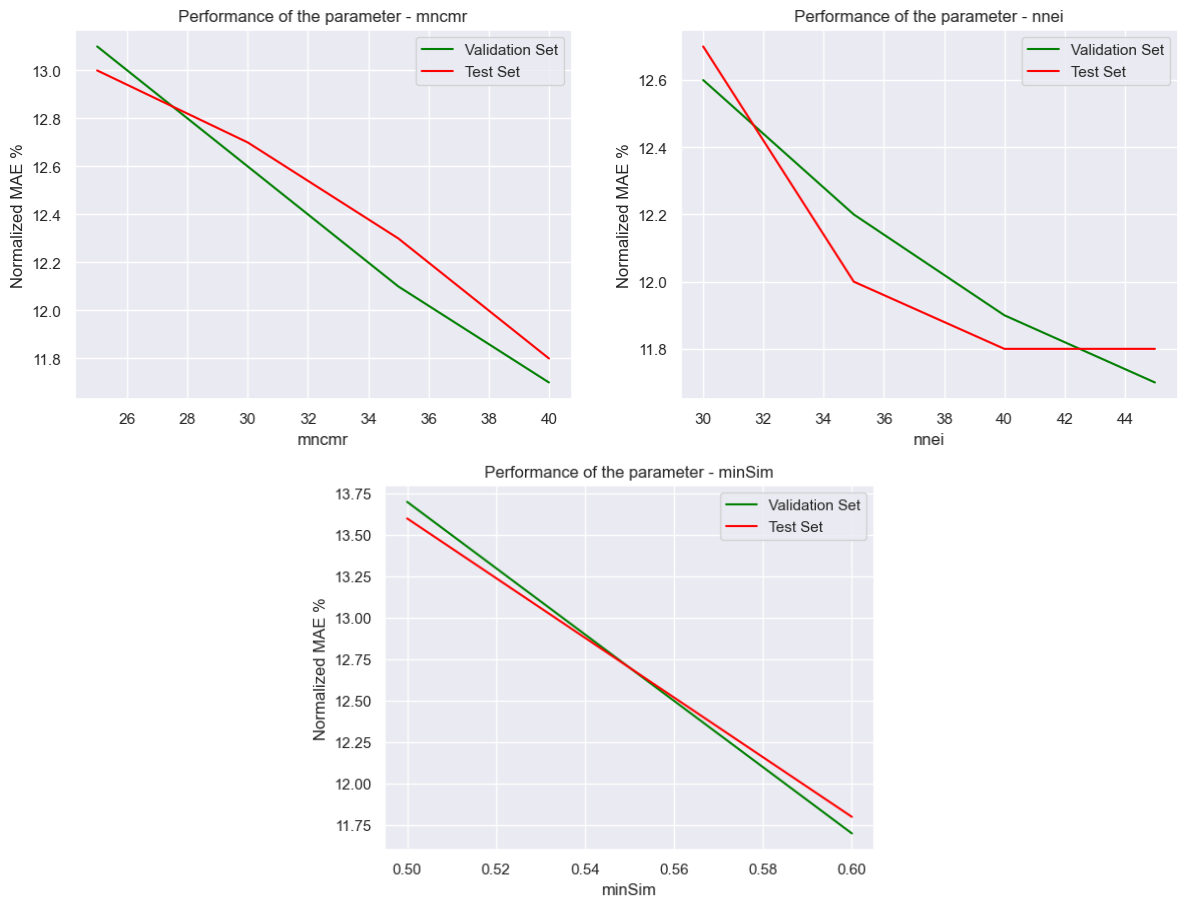


Figure 8

TO DO

I think it's rarely meaningful to consider cosine similarity on sparse data like this, not just because of sparsity (because it's only defined for dense data), but because it's not obvious the cosine similarity is meaningful. For example a user that rates 10 movies all 5s has perfect similarity with a user that rates those 10 all as 1. Magnitude doesn't matter in cosine similarity, but it matters in your domain.

It's much more likely that it's meaningful on some dense embedding of users and items, such as what you get from ALS.