



Big Data & Map Reduce Project CS561

Movie Recommendation System

Using PySpark

Team Members :

Harsh Gahlot T21041

Ravi Yadav T21050

Sucheta Panda T21046

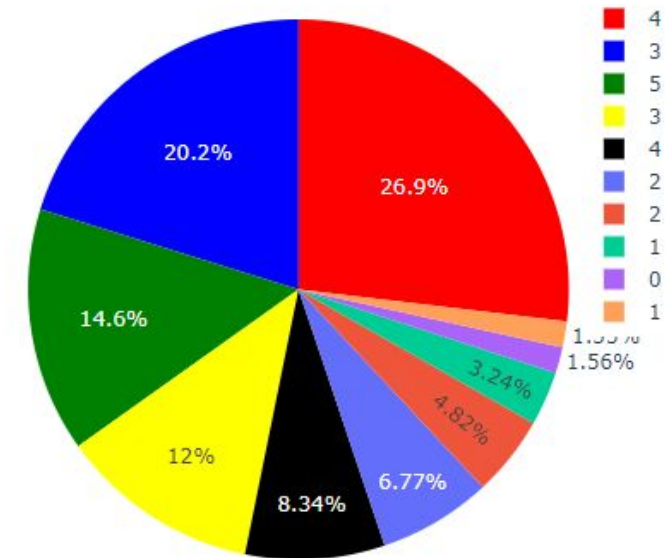
Abbas Husain T22057

Gaurav Shukla T22052

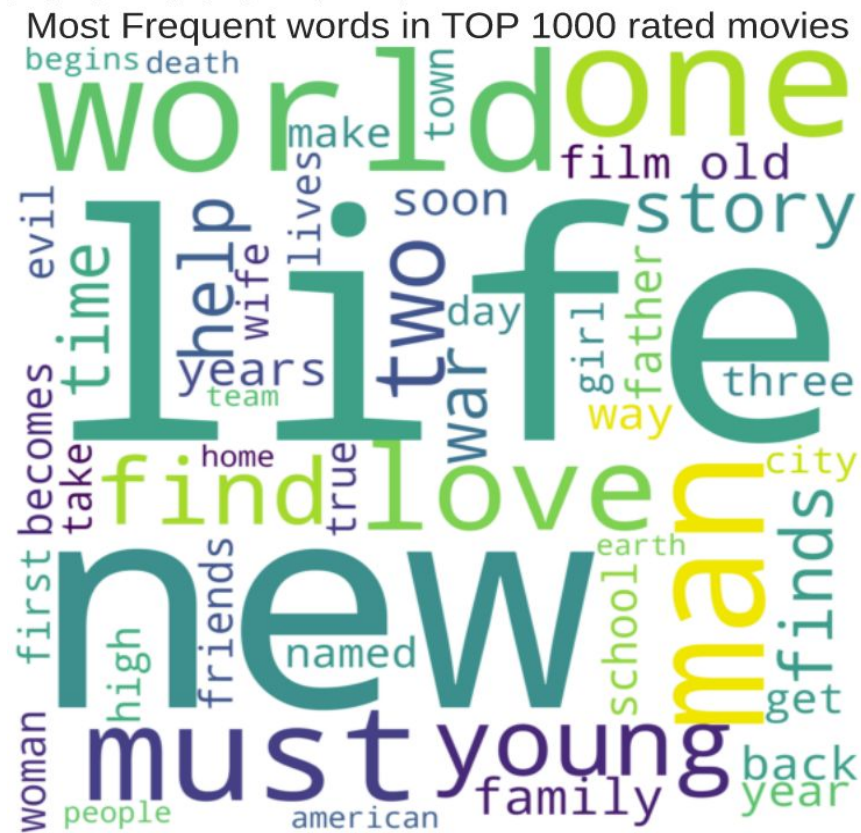
Regarding the Dataset

- Dataset contains metadata for all 45,000 movies listed in the Full MovieLens Dataset. The dataset consists of movies released on or before July 2017. Data points include cast, crew, plot keywords, budget, revenue, posters, release dates, languages, production companies, countries, TMDb vote counts and vote averages.
- This dataset also has files containing 26 million ratings from 270,000 users for all 45,000 movies. Ratings are on a scale of 1-5 and have been obtained from the official GroupLens website.

Ratings by Total Percent



Most frequently used words in top rated movies

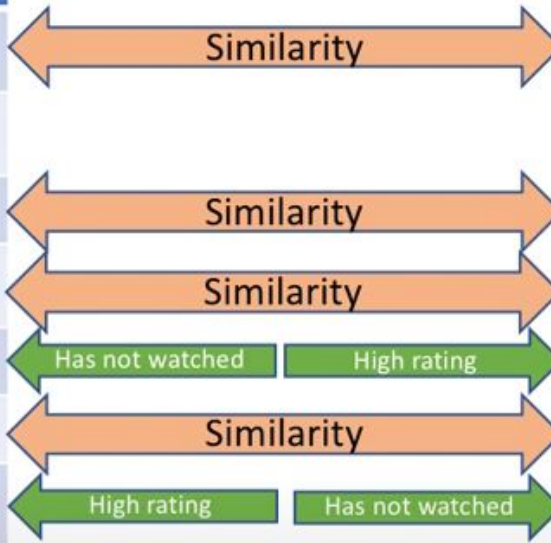


Recommendation System

Customer A



Movie	Rating
Dark Knight	5
The Notebook	-
Iron Man	4
Finding Nemo	3
Shrek	-
Tangled	1
Jurassic Park	4



Movie	Rating
Dark Knight	5
The Notebook	-
Iron Man	5
Finding Nemo	3
Shrek	4
Tangled	2
Jurassic Park	-

Customer B



You might also like:



You might also like:



Why ALS ?

Limitations in other Algorithms :

- **Popularity Bias**
- **Item Cold-Start Problem**
- **Scalability Issue**

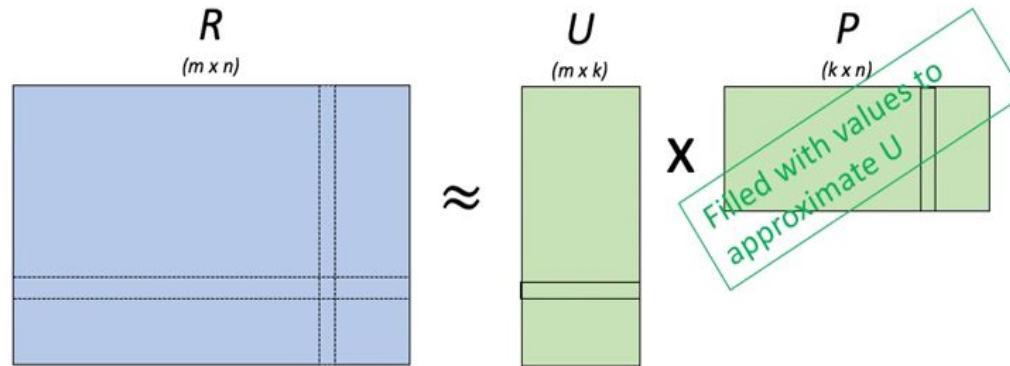
ALS Algorithm

	Movie 1	Movie 2	Movie ...	Movie N
User 1	1	BLANK	BLANK	3
User 2	BLANK	5	BLANK	3
User 3	BLANK	BLANK	1	BLANK
User 4	2	3	BLANK	BLANK
User 5	BLANK	BLANK	1	BLANK
User 6	4	BLANK	5	BLANK
User 7	BLANK	4	BLANK	BLANK
User ...	BLANK	3	BLANK	BLANK
User m	BLANK	BLANK	BLANK	4



	Movie 1	Movie 2	Movie ...	Movie N
User 1	1	4	2	3
User 2	1	5	3	3
User 3	2.5	2.8	1	3.5
User 4	2	3	2	3.5
User 5	2.5	2.8	1	3.1
User 6	4	1.2	5	1.4
User 7	1	4	2.5	3
User ...	2	3	2	3
User m	1	4	2	4

ALS Algorithm



$$Error_{ij} = \underbrace{\sum w_{ij}}_{\text{Completion Term}} \cdot \underbrace{(R_{ij} - u_i \times p_j^T)}_{\text{Cost Function}} + \underbrace{\lambda(\|U\|_2 + \|P\|_2)}_{\text{Regularization Term}}$$

Completion Term

where $w_{ij} = \begin{cases} 1 & R_{ij} \text{ is known} \\ 0 & R_{ij} \text{ is unknown} \end{cases}$

Cost Function

Minimizes the difference between the product of our factor matrices and the original ratings matrix.

The matching solutions for u_i and p_j are:

$$u_i = (P^T \times w_i \times P + \lambda I)^{-1} \times P^T \times w_i \times r_i$$

$$p_j = (U^T \times w_j \times U + \lambda I)^{-1} \times U^T \times w_j \times r_j$$

Regularization Term

Prevents overfitting by applying a small amount to the error which requires more time/iterations to full minimize.

Output

User 472's Movie Ratings

movieId	userId	rating	title	genres
1285	472	5	Heathers (1989)	Comedy
342	472	5	Muriel's Wedding (1994)	Comedy
3671	472	5	Blazing Saddles (1974)	Comedy Western
3146	472	5	Deuce Bigalow: Male Gigolo (1999)	Comedy
2490	472	5	Payback (1999)	Action Thriller
1721	472	5	Titanic (1997)	Drama Romance
2144	472	5	Sixteen Candles (1984)	Comedy Romance
2248	472	5	Say Anything... (1989)	Comedy Drama Romance

Recommendations for User 472

movieId	ratings	title	genres
67504	5.609112739562988	Land of Silence and Darkness (Land des Schweigens und der Dunkelheit) (1971)	Documentary
83411	5.609112739562988	Cops (1922)	Comedy
83318	5.609112739562988	Goat, The (1921)	Comedy
83359	5.609112739562988	Play House, The (1921)	Comedy
40412	5.151800632476807	Dead Man's Shoes (2004)	Crime Thriller
80599	5.048202037811279	Buster Keaton: A Hard Act to Follow (1987)	Documentary
4405	5.048202037811279	Last Laugh, The (Letzte Mann, Der) (1924)	Drama
25764	5.048202037811279	Cameraman, The (1928)	Comedy Drama Romance

References

- J. Zhang, Y. Wang, Z. Yuan and Q. Jin, "Personalized real-time movie recommendation system: Practical prototype and evaluation", April 2020Ut fermentum a magna ut eleifend. Integer convallis suscipit ante eu varius.
- M. Ahmed, M. T. Imtiaz and R. Khan, "Movie recommendation system using clustering and pattern recognition network," 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), 2018
- Bhalse, N., & Thakur, R. (2021). Algorithm for movie recommendation system using collaborative filtering. Materials Today: Proceedings.

Thank You.

