

# **Movies Recommendation System**

The background features a large dark blue trapezoidal shape on the left side, which contains the title text. To the right of this shape is a white area. At the bottom, there is a horizontal orange bar that is partially obscured by a white geometric shape on the left.

# About Our Dataset

1



# Movies Dataset

- These files contain 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.



- This dataset consists of the following files:
  - **movies\_metadata.csv**: The main Movies Metadata file. Contains information on 45,000 movies featured in the Full MovieLens dataset. Features include posters, backdrops, budget, revenue, release dates, languages, production countries and companies.
  - **keywords.csv**: Contains the movie plot keywords for our MovieLens movies. Available in the form of a stringified JSON Object.
  - **credits.csv**: Consists of Cast and Crew Information for all our movies. Available in the form of a stringified JSON Object.
  - **links.csv**: The file that contains the TMDB and IMDB IDs of all the movies featured in the Full MovieLens dataset.
  - **links\_small.csv**: Contains the TMDB and IMDB IDs of a small subset of 9,000 movies of the Full Dataset.
  - **ratings.csv**: Contains about 26 million of users ratings.

## Movies Dataset cont.



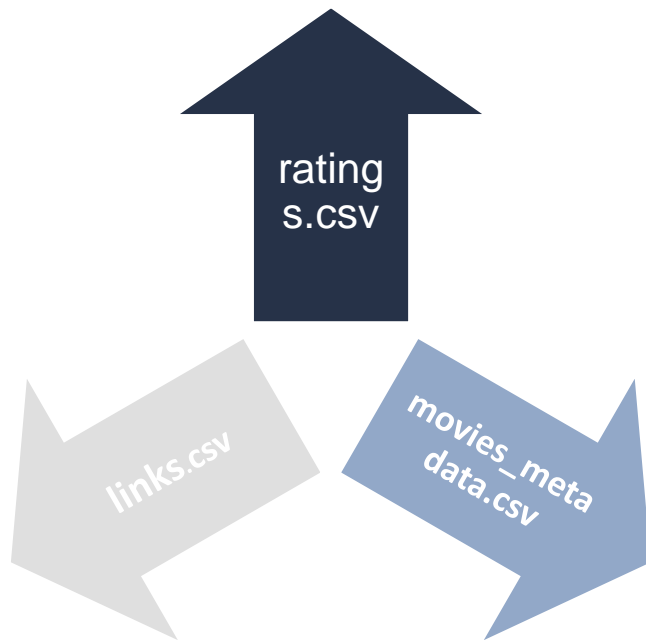
## Used Datasets cont.

- This dataset consists of the following files:
  - **movies\_metadata.csv**: The main Movies Metadata file. Contains information on 45,000 movies featured in the Full MovieLens dataset. Features include posters, backdrops, budget, revenue, release dates, languages, production countries and companies.
  - **links.csv**: The file that contains the TMDB and IMDB IDs of all the movies featured in the Full MovieLens dataset.
  - **ratings.csv**: Contains about 26 million of users ratings.



## Movies Dataset cont.

We will be using the following files only:



	userId	movieId	rating	timestamp
0	1	110	1.0	1425941529
1	1	147	4.5	1425942435
2	1	858	5.0	1425941523
3	1	1221	5.0	1425941546
4	1	1246	5.0	1425941556

# ratings.csv

- Consists of 4 columns and 26024289 rows
- Contains the ratings of each user for multiple movies.

Column	Non-Null Count	Dtype
adult	45466 non-null	object
belongs_to_collection	4494 non-null	object
budget	45466 non-null	object
genres	45466 non-null	object
homepage	7782 non-null	object
id	45466 non-null	object
imdb_id	45449 non-null	object
original_language	45455 non-null	object
original_title	45466 non-null	object
overview	44512 non-null	object
popularity	45461 non-null	object
poster_path	45080 non-null	object
production_companies	45463 non-null	object
production_countries	45463 non-null	object
release_date	45379 non-null	object
revenue	45460 non-null	float64
runtime	45203 non-null	float64
spoken_languages	45460 non-null	object
status	45379 non-null	object
tagline	20412 non-null	object
title	45460 non-null	object
video	45460 non-null	object
vote_average	45460 non-null	float64
vote count	45460 non-null	float64

# Movies\_metadata.csv

- Consists of 24 columns and 45466 rows
- Missing values in multiple columns



# 2

## Data Cleaning



## Handling Missing Values

Movies\_metadata Dataset has multiple columns contain missing values as shown in the following image.

adult	0.000000
belongs_to_collection	90.112864
budget	0.000000
genres	0.000000
homepage	82.883418
id	0.000000
imdb_id	0.037401
original_language	0.024201
original_title	0.000000
overview	2.098871
popularity	0.011000
poster_path	0.849229
production_companies	0.006600
production_countries	0.006600
release_date	0.191407
revenue	0.013200
runtime	0.578620
spoken_languages	0.013200
status	0.191407
tagline	55.100873
title	0.013200
video	0.013200
vote_average	0.013200
vote_count	0.013200



## Handling Missing Values cont.

### STEPS:

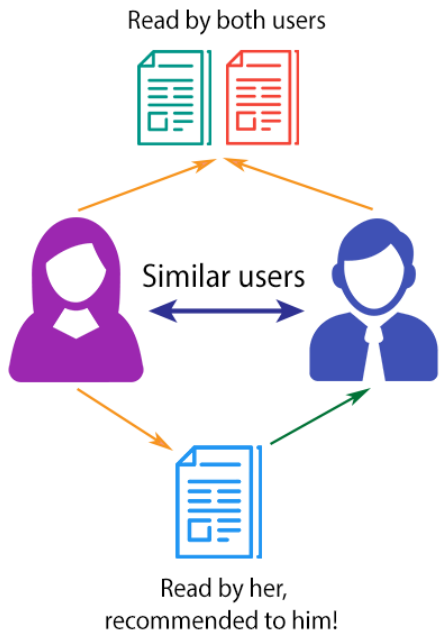
1. belongs\_to\_collection and homepage columns have more than 80% of their data so It's better to be dropped.
2. Budget had missing values embedded in the form of zero which were more that 50% of the dataset so It was better to be dropped.
3. Revenue , runtime and poster path values were found in tmdb website, so I used web scrapping to fill missing values.
4. Original language , status spoken languages popularity and revenue where handled by sklrean simple imputer
5. The rest columns had mixed values, so I dropped them.

# 3

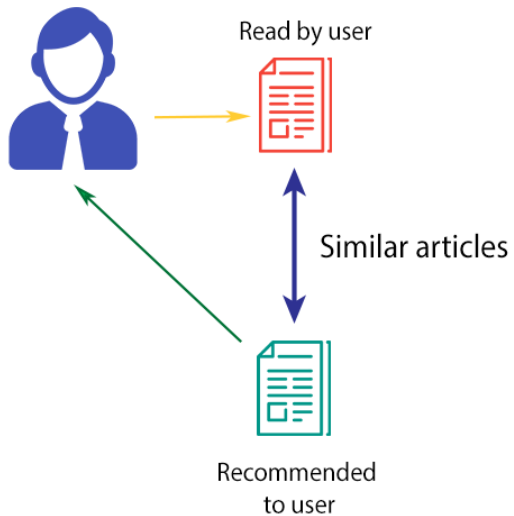
## Project Idea

# Project Idea

## COLLABORATIVE FILTERING



## CONTENT-BASED FILTERING



## Project Idea cont.

■ **Content-based** approach requires a good amount of information of items' own features, rather than using users' interactions and feedbacks. For example, it can be movie attributes such as genre, year, director, actor etc., or textual content of articles that can be extracted by applying Natural Language Processing.

■ **Collaborative Filtering**, on the other hand, doesn't need anything else except users' historical preference on a set of items. Because it's based on historical data, the core assumption here is that the users who have agreed in the past tend to also agree in the future. In terms of user preference, it is usually expressed by two categories.









## Project Idea cont.

■ **Explicit Rating**, is a rate given by a user to an item on a sliding scale, like 5 stars for Titanic. This is the most direct feedback from users to show how much they like an item.

■ **Implicit Rating**, suggests users' preference indirectly, such as page views, clicks, purchase records, whether to listen to a music track, or not and so on.

I will be converting explicit data into implicit data in this dataset.  
Why? As in real world scenarios users barely rate items they saw.

## Project Idea cont.

	Item 1	Item 2	Item 3	Item 4	Item 5
Alice					
Bob					
Charlie					

Bob ~ Charlie





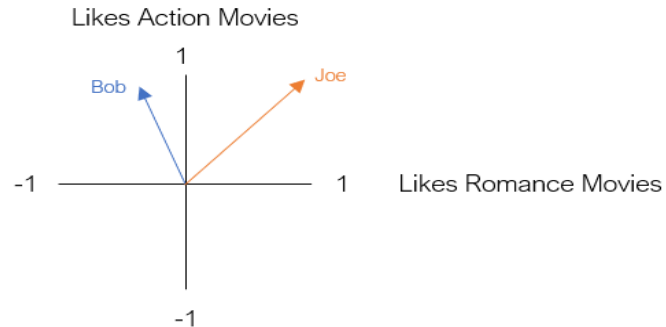
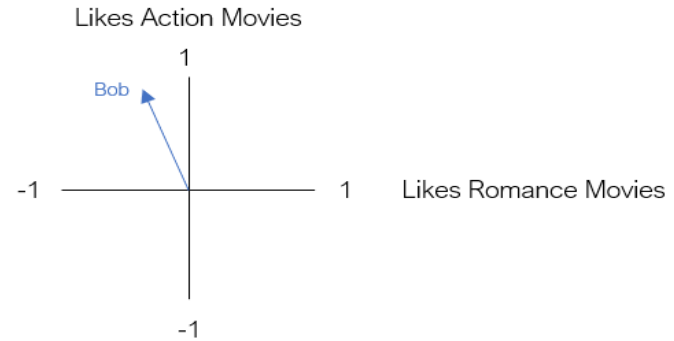
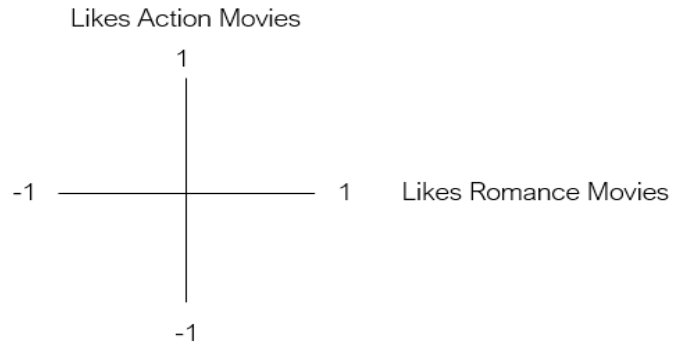
# 4

## Pytorch Model

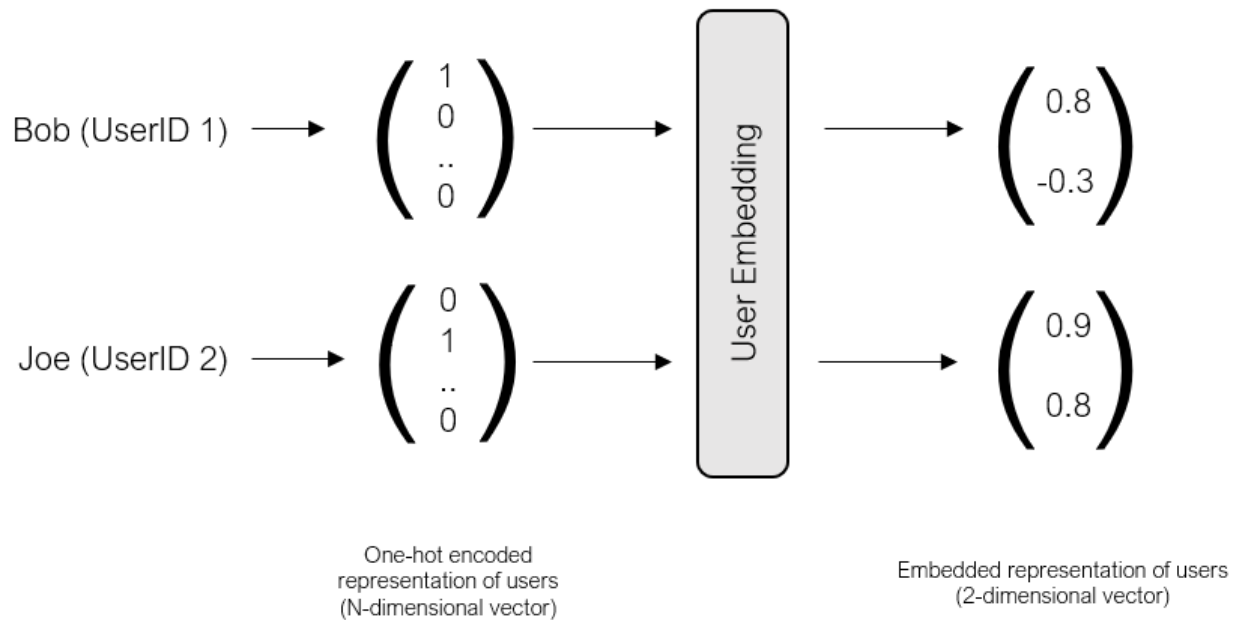
## Leave-One-Out train test split

- Leave-one-out is a method obtained by setting  $k = 1$  in the leave- $k$ -out method. Given an active user, we withhold in turn one rated item.
- The learning algorithm is trained on the remaining data.
- The withheld element is used to evaluate the correctness of the prediction and the results of all evaluations are averaged in order to compute the final quality estimate.

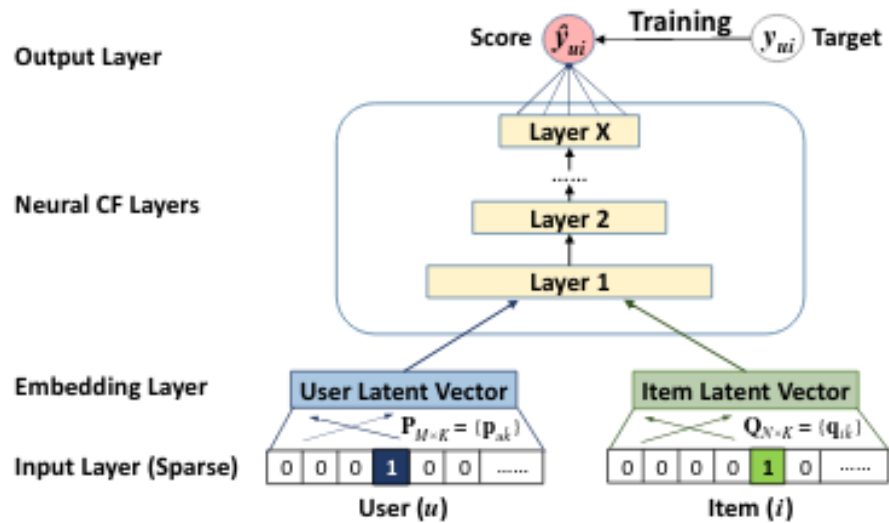
# User Embedding



# User Embedding



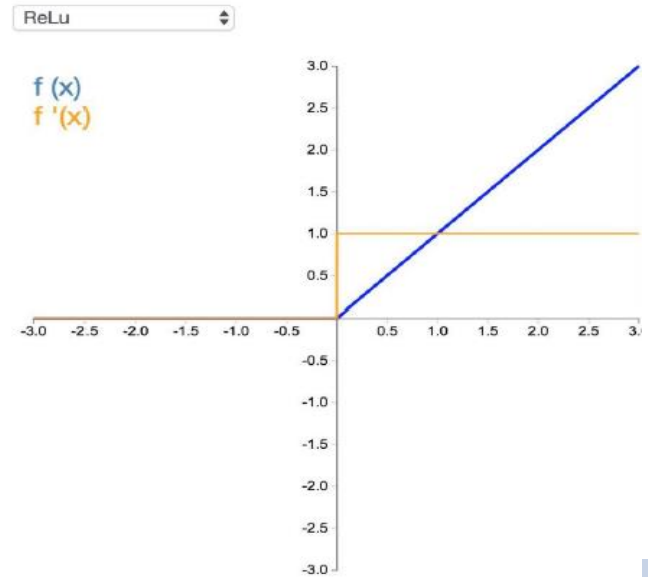
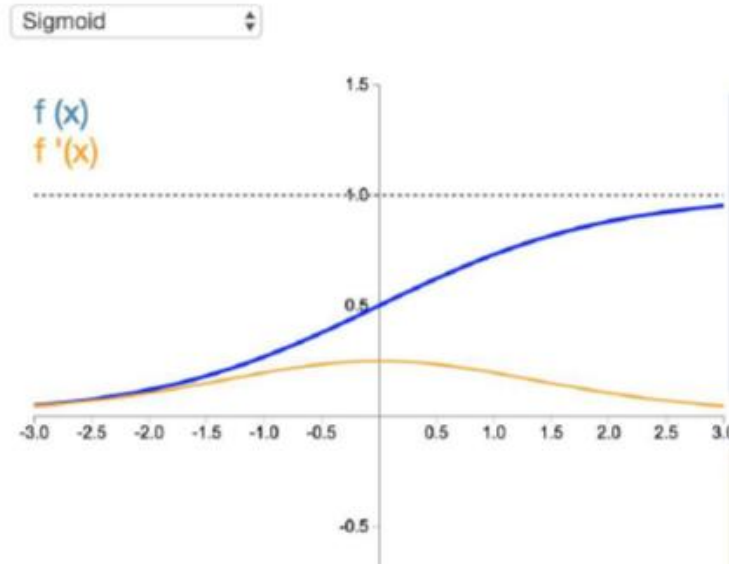
# Neural Network



## Model Summary

- User embedding layer
- Movie embedding layer
- Previous layers concatenated and passed into first hidden layer.
- First hidden layer takes 32 inputs and gives 64 outputs. Uses Relu activation function.
- Second hidden layer takes 64 inputs and gives 32 outputs. Uses Relu activation function.
- Output layer takes 32 inputs and return one output. Uses sigmoid activation function.

# Model Summary



# 5

## Model Evaluation



# Hit Ratio @10

- For each User, I have combined 99 movies that the user haven't interacted with.
- Then, I Added the test Movie that I have for that user.
- Now , The model have 100 movies. I will select top 10 movies for that user. if our test movie included in them , we assume this as a hit.
- Calculate hit for all test user.
- Finally hit ratio will be the average of all hits.

# Model Evaluation

We have reached a very acceptable hit ratio in my opinion

```
Hit Ratio of our model is 84.51817190964783
```

**THANK YOU**