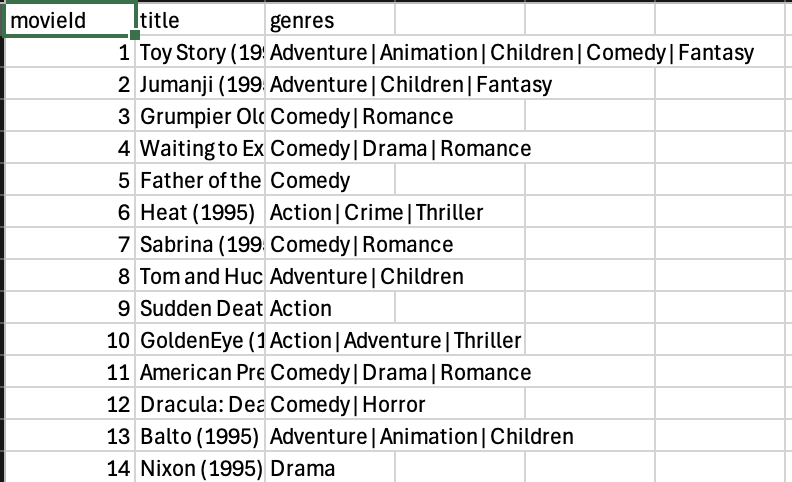
# רעיון לקורס ״סדנה במסדי נתונים״

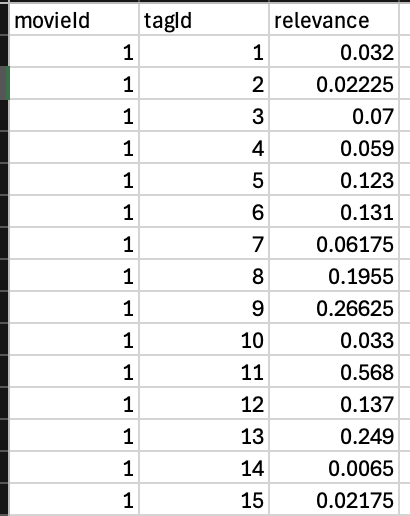
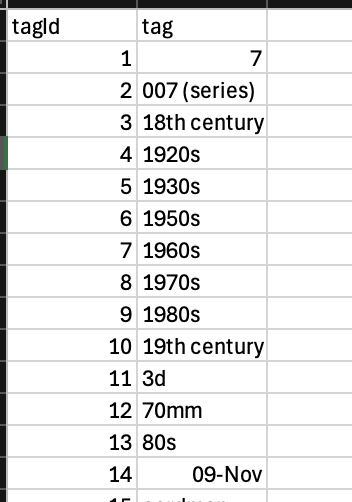
Dataset: [The movielens 25M dataset](https://grouplens.org/datasets/movielens/) (specifically the full dataset for education and development), contains approximately 33,000,000 ratings and 2,000,000 tag applications applied to 86,000 movies by 330,975 users. Includes tag genome data with 14 million relevance scores across 1,100 tags. Last updated 9/2018.

The dataset is composed of the following files:  

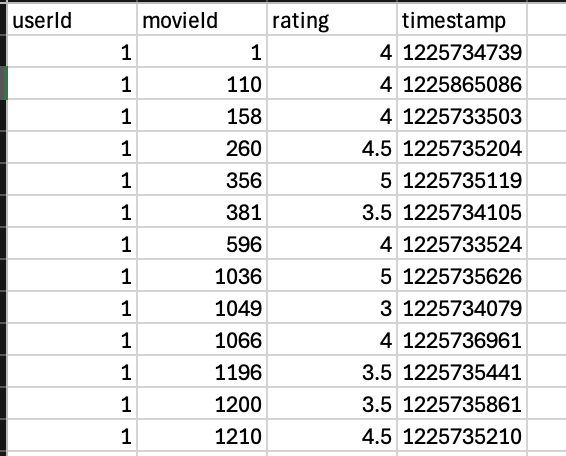

## Explanation and view of the files:

Movies.csv:  


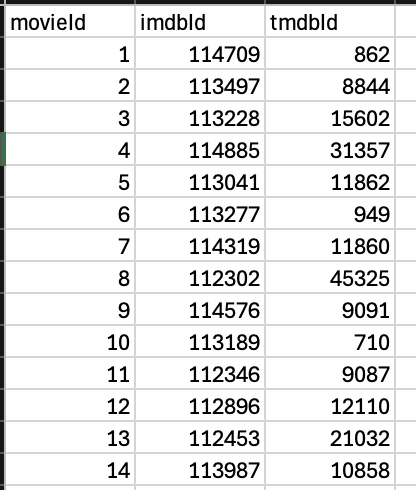
Basic title and genres information for each movieID

Genome-scores.csv and genome tags.csv:  
 

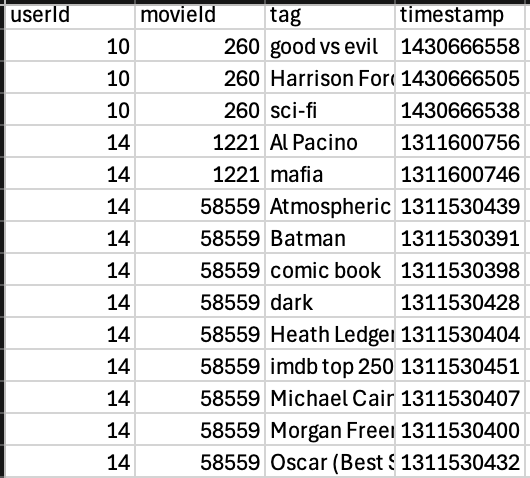
To the best of our understanding, gives advanced genres and community genres for each movie, for example the movieid 1 has gotten a score of 0.02225 for being related to the 007 (James Bond) series.

Rating.csv:  


Rating for each recorded user for each movieid

Links.csv:  


Imdbid’s for each movie id in the dataset

And lastly, tags.csv:  
  
Which tags each user gave to which movie.

# Application Idea:

Our application will provide users with:

1. Similar movie recommendations based on user ratings, movie characteristics, genomes and tags.
2. Search movie by genres , genomes and average user ratings.
3. User analytics dashboard showing personal user analytics.
4. Genre based movie rankings.

# Why we think it fits the project requirements:

Each idea centers on an open dataset (movielens) and is scoped to exceed 100k rows when connected appropriately in the DB and used meaningfully by the application. The dataset does not require registreation.

Further more, just to connect all the files togethere will take some “complex” SQL querys, but the application idea will need even more “complex” querys. Most of our data processing will be on the SQL query side.

For example, the movies table will probably look something like this:

* movie\_id (INT, PRIMARY KEY) - Unique movie identifier
* title (VARCHAR(255)) - Movie title
* imdb\_id (VARCHAR(20)) - IMDb identifier
* avg\_rating (DECIMAL(3,2)) - Cached average rating
* rating\_count (INT) - Number of ratings received

As can be seen, the columns avg\_rating and rating\_count will need to be calculated by us, but are imporntant to correcttly show each movie.

Lastly, this can easily be categorized to the “social media” requested genre of ideas, because we have a lot of user data in here that will help us recommend movies for other used.

# Requested Practical Technologies:

Backend – FastAPI (Python) is an **open-source** pakcage in python that helps build APIS. We will write a small REST API microservice in python that will access the sql DB and return whetever we will decide as part of our API.

Frontend – Angular (Typescript) is an **open-source** one of the most common frontend technologies in the market, and is widely used.

הערות של אוסנת:

1. להוסיף עוד מידע על הסרטים כגון במאים ושחקנים
2. לא חייב לקחת סרטים מאוד ישנים, אפשר לוותר על סרטים ישנים מהסיבה שעל סרטים ישנים יש פחות מידע נוסף שאפשר להביא
3. נדרש לעשות שינויים בסכמה, הדאטא מגיע בנוי מדי וצריך לבנות סכמה משלנו
4. גרפים של הדאטא, כמו לדוגמא טרנדים, או מידע על שחקנים כמו לדוגמא הביקורות של שחקן על סרטים בשנים האחרונות
5. באמצע הסמסטר יהיה בוחן SQL
6. לא חייב docker image, אפשר גם איזשהו script
7. שיהיה הרבה טבלאות, ושהם יהיו עשירות במידע
8. שלא יהיה duplicates
9. שאילתות יעילות ונכונות
10. צריך שיהיה מיוטציות גם, כלומר שינויים גם לדאטא
11. להשתמש באינדקסים! יוריד הרבה נקודות
12. רוב הנקודות הם על החלק של ה DB.
13. הקוד צריך להיות בנוי בצורה שבה יש חלק נפרד לגישה ל DB, חלק נפרד ל UI, וחלק נפרד ללוגיקה, במקרה שלנו גם בסדר לפצל ל DB, באקנד ופרונט
14. החיבור יקרה רק פעם אחת בהתחלה
15. ERROR HANDLING! חשוב מאוד לכל פעולה של ה DB.
16. שיהיה לוגים טובים במקרה של נפילה
17. שלושה מפגשים – הראשון האחרון והאמצע
18. שלושה מסמכים שצריך להגיש, גם כל הקוד וגם user manual שכולל גם איך להריץ וגם מה כל פיצר באתר עושה, והאחרון זה software documentation שכולל הסבר על הקוד עצמו. לדבר עם אוסנת לקראת הסוף כדי שתסביר יותר לעומק.
19. בונוס נקודות על אלגוריתמים או UI מעניין

מטלות:

1. כשמגיעים לחלק של ה DB לבקש מאוסנת סרטון
2. להיסגר על סכמה