# Movie Recommendation System

Yashwanth Reddy Challa

Winter in Data Science

Winter 2022-2023

# Table of Contents

# Dataset Used

- Dataset of 85,000+ movies in over 50 languages as given by mentor.
- Utilized *genre*, *director*, *actors*, *title*, *description*, *language*, *reviews_from_users*, *reviews_from_critics*, and *year* columns of the dataset.

# Data Preprocessing

- Truncated the dataset to the 20,000 non-duplicate latest movies due to limited computational power in calculating cosine similarity matrix.
- Dropped insignificant columns to reduce clutter: *imdb_title_id*, *original_title*, *date_published*, *duration*, *country*, *writer*, *production_company*, *avg_vote*, *votes*, *budget*, *usa_gross_income*, *worlwide_gross_income*, and *metascore*.
- Replace all *Null* values by *'None'* string, so that an entry may be checked for being empty easily
- Only the top 50 languages get to retain their name. the others fall into a blanket category called *'Other'*.

# Exploratory Data analysis

1. Plotted movies by language to indicate how much data was available per language. It is interesting to note that no other gap comes even close to the gap in the number of movies between the top two languages i.e English and Spanish Language.
2. Plotted the average user and critic review for every genre. More generic themes such as action-adventure, romance, sci-fi, and comedy were enjoyed more by users than critics. However, drama, thriller, and horror were rated by critics almost as much as users.

# Feature Engineering

- My features were the genre, actors, language, director, and description column of each IMdB entry
- While the others could be converted to lists and directly added to a bag of words, the description column had to be Raked for keywords, which were then added to the bag of words
- Only considered lead actors (first three) by slicing the actors list

# Predictive task performed

- Calculated a 20000 by 20000 cosine similarity matrix using the cosine similarity between the bag of words of every pair of movies
- When the user enters the movie they just watched, the ten most similar movies in our dataset are obtained by getting the top ten most similar movies from the cosine similarity matrix
- User can query movies as old as 2013

# Summary of the cosine similarity model

- Due to the heavy computation involved in calculating the cosine similarity matrix, using the full dataset of 85000 movies would have taken too long. I used the 20000 latest movies in that dataset.
- Cosine similarity allows us to build a purely content-based recommender, based on the engineered features.

# Conclusion and scope

- One major lacking of this project is the lack of any collaborative filtering, which is quite heavily used in OTT platforms. If this project was implemented within a large scale OTT platform with tons of user-data, it would be possible to incorporate some collaborative features into the cosine similarity.
- With access to quality computational power, the full dataset with 85000 movies could be analyzed
- The dataset, unfortunately, doesn't have many popular titles. So please don't be disappointed when it throws an error. Just try another title!