# CDS 492: Data Science Capstone Project Proposal

Avantika Prativadhi

# In this presentation:

- Updated my project idea
- Different techniques used for recommendation systems
- Limitations with the systems
- Research done on the techniques
- My new proposed idea
- Datasets that will be used

# Idea: Movie Recommendation System

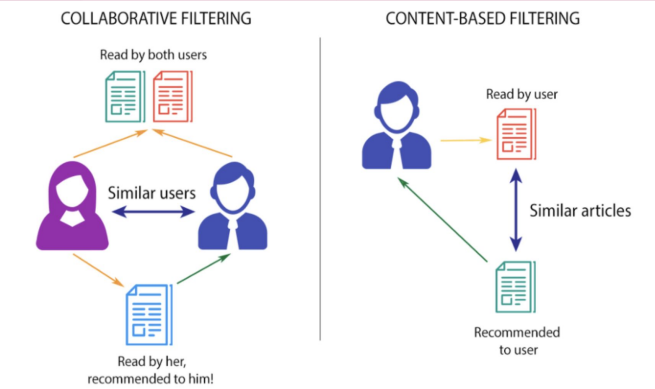
- Three techniques of recommendation systems:
  - **Collaborative Filtering** - matches users with similar preferences and recommends movies that are highly rated by these similar users.
  - **Content-Based Filtering** - user provided data explicitly (rating) or implicitly (clicking on a link)
  - **Hybrid Filtering** - a mix of both



# Literature Review

- **Sparsity Problem:** Most users do not rate most of the items and the availability of ratings are usually sparse.
- **Cold Start Problem**: When a new user enters the system, there's not enough information about the user. For a new movie, there are not many ratings given.
- **Scalability**: With massive number of users and movies, there is large amount of data to fit the model on. Even if performance is improved, most of the time it results in accuracy reduction.
- **Over Specialization Problem**: Users get recs of movies that resemble ones they already know. The model is exactly fit to their taste which prevents them from discovering newer/ different movies.

# Literature Review




*Collaborative v/s Content-based filtering illustration*

- **Content Based Filtering:**
  - Jieun Son and Seoung Bum Kim proposed content- based filtering using multiattribute networks that contain attribute information about item. By using various attributes, prevent overspecialization and recommend various items.
- **Collaborative Filtering:**
  - Ching-She Wu et all. - used collaborative filtering approaches (user based and item based) using Pearson correlation to find similarity. Also used Nearest N User Neighbourhood to obtain N similar users. Stored into Hadoop distributed file system (HDFS) and used Yahoo Research Web Scope Database.

# Other Related Works

- N.Pradeep et al. build a recommendation system based on content on cast, keywords, teams and genres, then a single column is created as the sum of the 4 attributes and it acts as a dominant factor for that system recommendation of films.
- Yadav Vikash et al. develop a movie recommender system with the help of clustering using K-means clustering technique and data pre-processing using Principal Component Analysis.

# Proposed Work :

## K-means clustering algorithm and NLP to improve recommendation system

- Use NLP and clustering on movie descriptions of the three platforms.
- Find patterns in the text and create clusters based on similarity.
- Compare the algorithm between the three platforms to see if one does better.

# Datasets:

Amazon Prime

## Content

This dataset has two files containing the titles (**titles.csv**) and the cast (**credits.csv**) for the title.

This dataset contains **+9k** unique **titles on Amazon Prime** with 15 columns containing their information, including:

- id: The title ID on JustWatch.
- title: The name of the title.
- show_type: TV show or movie.
- description: A brief description.
- release_year: The release year.
- age_certification: The age certification.
- runtime: The length of the episode (SHOW) or movie.
- genres: A list of genres.
- production_countries: A list of countries that produced the title.
- seasons: Number of seasons if it's a SHOW.
- imdb_id: The title ID on IMDB.
- imdb_score: Score on IMDB.
- imdb_votes: Votes on IMDB.
- tmdb_popularity: Popularity on TMDB.
- tmdb_score: Score on TMDB.

And **over +124k** credits of **actors and directors** on Amazon Prime titles with 5 columns containing their information:

- person_ID: The person ID on JustWatch.
- id: The title ID on JustWatch.
- name: The actor or director's name.
- character_name: The character name.
- role: ACTOR or DIRECTOR.

# Datasets:

Hulu:

## Content

This dataset has two files containing the titles (**titles.csv**) and the cast (**credits.csv**) for the title.

This dataset contains **+2k** unique **titles on Hulu** with 15 columns containing their information, including:

- id: The title ID on JustWatch.
- title: The name of the title.
- show_type: TV show or movie.
- description: A brief description.
- release_year: The release year.
- age_certification: The age certification.
- runtime: The length of the episode (SHOW) or movie.
- genres: A list of genres.
- production_countries: A list of countries that produced the title.
- seasons: Number of seasons if it's a SHOW.
- imdb_id: The title ID on IMDB.
- imdb_score: Score on IMDB.
- imdb_votes: Votes on IMDB.
- tmdb_popularity: Popularity on TMDB.
- tmdb_score: Score on TMDB.

And **over +30k** credits of **actors and directors** on Hulu titles with 5 columns containing their information, including:

- person_ID: The person ID on JustWatch.
- id: The title ID on JustWatch.
- name: The actor or director's name.
- character_name: The character name.
- role: ACTOR or DIRECTOR.

# Citations


Choi, S.-M., Han, Y.-S.: A content recommendation system based on category correlations. In: The Fifth International Multi-Conference on Computing in the Global Information Technology, pp. 1257–1260 (2010)

D. Das, H. T. Chidananda and L. Sahoo, "Personalized Movie Recommendation System Using Twitter Data" in Progress in Computing Analytics and Networking, Singapore:Springer, vol. 710, 2018.

SK. Ko et al., "A Smart Movie Recommendation System", *Lecture Notes in Computer Science*, vol. 6771, 2011.

N. Mishra, S. Chaturvedi, V. Mishra, R. Srivastava and P. Bargah, "Solving Sparsity Problem in Rating-Based Movie Recommendation System" in Computational Intelligence in Data Mining, Singapore:Springer, vol. 556, 2017.

https://www.kaggle.com/datasets/victorsoeiro/amazon-prime-tv-shows-and-movies
https://www.kaggle.com/datasets/victorsoeiro/hulu-tv-shows-and-movies?select=titles.csv
https://www.kaggle.com/datasets/unanimad/disney-plus-shows/code