**Filmrekommendationssystem med MovieLens Dataset**

# ****Introduktion****

Detta projekt syftar till att utveckla ett avancerat filmrekommendationssystem med hjälp av MovieLens-datasetet. Systemet använder en kombination av collaborative filtering och content-based filtering för att analysera användarbeteenden och föreslå relevanta filmer baserat på tidigare betyg och genrepreferenser. Målet är att skapa en modell som effektivt kan identifiera mönster i användardata och därigenom förbättra personaliseringen av filmrekommendationer.

# ****Metod****

## ****Dataset****

MovieLens-datasetet används som underlag och innehåller tre huvudsakliga filer:

* **movies.csv** – innehåller film-ID, titel och genrer, vilket ger en överblick av tillgängliga filmer.
* **ratings.csv** – innehåller information om användarbetyg, inklusive användar-ID, film-ID, betyg och tidsstämpel.
* **tags.csv** – innehåller användargenererade taggar kopplade till filmer, vilket möjliggör en djupare analys av filmers innehåll.

För att förstå datasetets struktur och egenskaper genomförs en inledande dataanalys och visualisering av nyckelstatistik.

## ****Dataanalys och Förbehandling****

### Grundläggande statistik

* Antal filmer, användare och betyg undersöks för att förstå datasetets storlek och variation.
* Fördelningen av betyg analyseras för att identifiera mönster och eventuella bias i data.
* Genrefördelningen undersöks för att avgöra vilka genrer som dominerar.

### Förbehandling av data

* Genrer konverteras till en lista för enklare bearbetning i analysen.
* En användar-film-matris skapas för att underlätta filtrering och beräkning av likheter.
* Cosine similarity beräknas mellan filmer för att skapa en likhetsmatris som används vid rekommendationsgenerering.
* Taggdata används för att förbättra precisionen genom att ta hänsyn till användargenererade beskrivningar av filmer.

# ****Implementerad Modell****

Rekommendationssystemet bygger på **en hybridmodell som kombinerar collaborative filtering och content-based filtering**:

1. **Collaborative Filtering**
   * Cosine similarity används för att mäta hur lika två filmer är baserat på användarbetyg.
   * Filmer som ofta betygsätts liknande av användare identifieras som närbesläktade.
2. **Content-Based Filtering**
   * Filmer analyseras baserat på deras genre och taggdata.
   * Filmer med liknande innehållsegenskaper identifieras även om de inte har betygsatts av samma användare.
3. **Hybrid-Rekommendationer**
   * Kombinationen av collaborative och content-based filtering förbättrar precisionen genom att använda både användarbaserade preferenser och filmens innehåll.
   * Detta gör systemet mer robust mot cold-start-problem för nya filmer och användare.

# ****Resultat****

## ****Analys av Datasetet****

* Datasetet innehåller en stor mängd filmer och användardata, vilket möjliggör tillförlitliga rekommendationer.
* En analys av betygsfördelningen visar att användare tenderar att ge högre betyg snarare än att vara neutrala.
* Drama, Comedy och Action är de mest representerade genrerna, vilket kan påverka modellens förmåga att generera mångsidiga rekommendationer.

## ****Prestanda och Utvärdering av Modellen****

* Rekommendationerna visar hög genreöverensstämmelse, vilket tyder på att systemet är effektivt för att identifiera filmer av liknande typ.
* Hybridmetoden ger förbättrad precision jämfört med enbart collaborative filtering.
* Testning visar att användarpreferenser fångas upp väl av modellen, och föreslagna filmer överensstämmer ofta med användarens tidigare betygsatta filmer.
* Cold-start-problemet minskar genom användning av content-based filtering.

## ****Exempel på Rekommendationer****

För filmen **"Toy Story"** föreslår modellen:

1. Liknande animerade filmer med familjevänliga teman.
2. Filmer som delar genrer och berättarstil med originalfilmen.
3. Filmer med höga likhetspoäng, baserat på både betyg och innehållsegenskaper.

# ****Diskussion och Framtida Förbättringar****

## ****Styrkor****

1. Enkel och effektiv implementering av hybrid-rekommendationer.
2. Bättre precision jämfört med enbart collaborative filtering.
3. Skalbar metod som kan tillämpas på större dataset utan större prestandaförlust.

## ****Begränsningar****

1. **Cold-start problem** – även om content-based filtering hjälper, är rekommendationer fortfarande begränsade för helt nya filmer.
2. **Begränsad användning av metadata** – ytterligare faktorer som regissör och skådespelare kan inkluderas.
3. **Ignorering av temporala aspekter** – systemet tar inte hänsyn till förändringar i användarpreferenser över tid.

## ****Möjliga Förbättringar****

1. **Vidareutveckling av hybrid-modellen** – inkludera fler faktorer, såsom användarhistorik och betygstrender.
2. **Inkludering av fler metadata-fält** – exempelvis regissör och produktionsbolag.
3. **Implementering av en tidsberoende modell** – justera rekommendationerna baserat på förändringar i användarbeteenden över tid.

# ****Slutsats****

Detta filmrekommendationssystem, baserat på en hybridmodell av collaborative och content-based filtering, kan generera relevanta och användarcentrerade rekommendationer. Modellen visar sig vara effektiv på att hitta liknande filmer baserat på både användarbetyg och innehållsegenskaper. Vidare förbättringar kan göras genom att inkludera fler metadata-fält och implementera tidsberoende modeller för att ytterligare förbättra rekommendationernas precision.

# ****Referenser****

* MovieLens dataset: https://grouplens.org/datasets/movielens/
* Sklearn dokumentation: https://scikit-learn.org/
* Pandas dokumentation: https://pandas.pydata.org/