BIG DATA MOVIE RECOMMENDATION SYSTEM

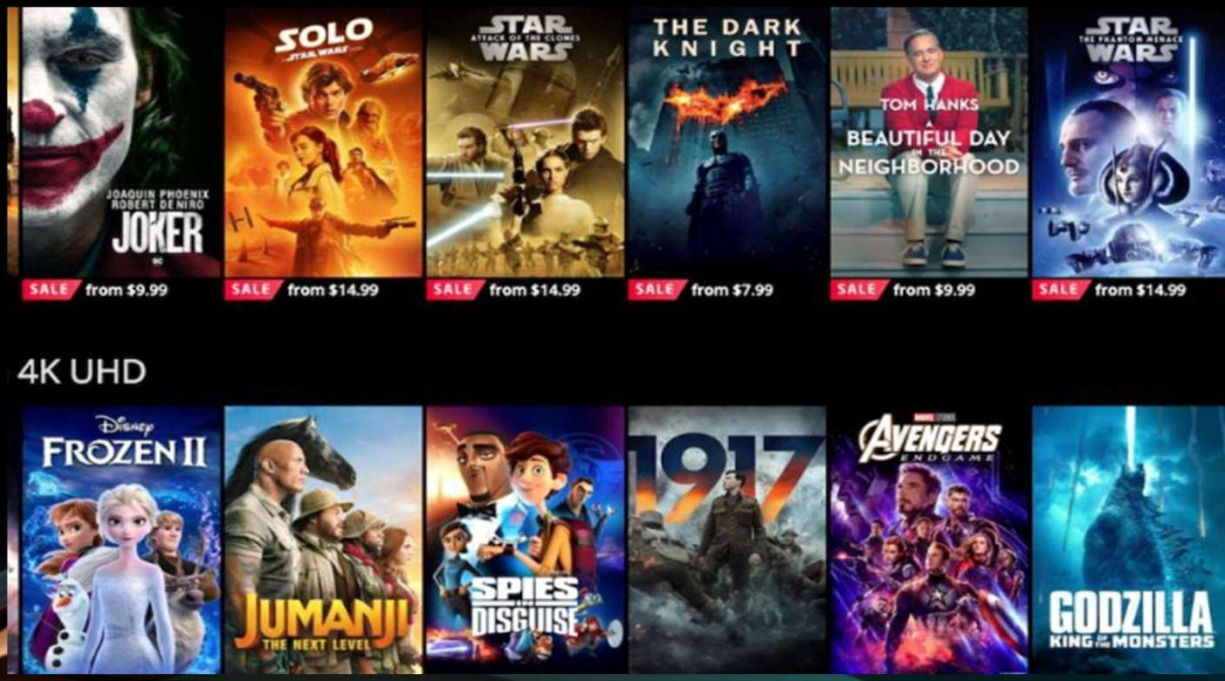
USING SPARK ALS + SENTIMENT ANALYSIS

Group – 6

- Asad Ashraf Shaik
- Kavya Sri Arroju
- Yash Kumar Gajera

Course Intructor

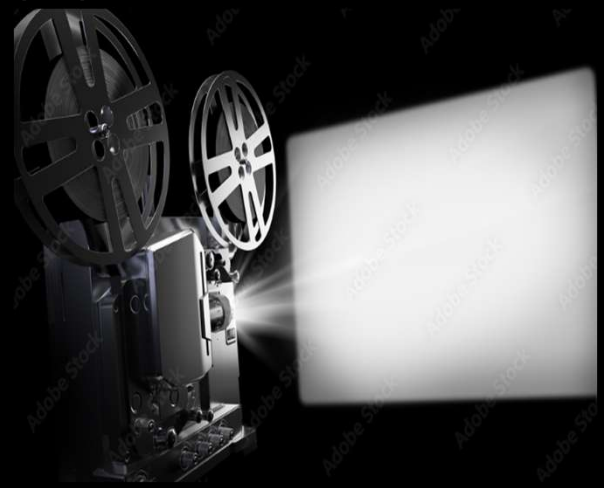
- Dr Najam Hassan



INTRODUCTION

Movie recommendation systems play a critical role in helping users navigate massive catalogs of content. Our project focuses on leveraging big-data processing techniques to create a personalized, sentiment-aware movie recommendation pipeline that mimics modern industry systems like Netflix.

- Reduces user overload in content platforms
- Integrates user behavior + public sentiment
- Designed for scalability using distributed processing
- Applies big-data concepts from this course



SIGNIFICANCE OF THE PROJECT

This work shows how big data platforms can be used to generate actionable insights at scale. The combination of collaborative filtering and sentiment analysis improves the quality, reliability, and interpretability of recommendations for real-world decision support.

- Demonstrates real-world big-data workflows
- Hybrid model → better user experience
- Supports business decisions using data-driven insights
- Matches industry trends in recommender systems

PROJECT OBJECTIVES

Our main objective was to build a system capable of processing millions of ratings to provide personalized movie suggestions. We also aimed to enhance recommendations using sentiment extracted from metadata.

- Train a scalable model using Spark ALS
- Normalize and join large datasets (ratings + metadata)
- Compute sentiment scores from TMDB data
- Build a simple UI for user-level decision making
- Generate reports for insights and evaluation

DATA SOURCES & COLLECTION

We used two major datasets: MovieLens 20M for user behavior and TMDB metadata for sentiment and movie attributes. Both datasets were loaded directly into Spark and cleaned for consistency.

- MovieLens 20M: 20M ratings, 27k movies
- TMDB: genres, year, popularity, score (converted to sentiment)
- Cleaning: punctuation removal, title normalization
- Challenges: title mismatching, missing metadata
- Fallback sentiment = 0.5 for unmatched movies

TOOLS & TECHNOLOGIES

The system was built using a stack of big-data tools that ensure scalability and efficiency. Apache Spark served as the backbone for distributed computation.

- Apache Spark: data processing + ALS model
- PySpark: transformations, SQL joins
- KaggleHub: automated dataset download
- Colab: cloud execution environment
- ipywidgets: UI popup interface

DATA PIPELINE / ARCHITECTURE

Our pipeline combines data ingestion, preprocessing, modeling, integration, and UI generation. Each step uses Spark to maintain performance across large datasets.

- Load MovieLens + TMDB into Spark DataFrames
- Clean titles, extract sentiment, join datasets
- Train ALS collaborative filtering model
- Predict user ratings + combine with sentiment
- Serve results in popup UI

MODELING APPROACH

We employed ALS for collaborative filtering due to its suitability for large sparse datasets. Sentiment scores served as an additional signal for enhancing user recommendations.

- ALS learns latent factors for users & movies
- Predicts ratings for unknown movie-user pairs
- Sentiment = (TMDB score / 10) → normalized 0–1
- Final logic blends ALS + sentiment
- Helps address cold-start and improves accuracy

METHODOLOGY

We followed a simplified CRISP-DM approach, moving from understanding the recommendation problem to preparing data, training the ALS model, integrating sentiment, and delivering the final user-facing system.

- **Data Preparation:** Clean titles, join MovieLens 20M with TMDB metadata, compute sentiment.
- **Modeling:** Train Spark ALS to learn user–movie preferences at scale.
- **Evaluation:** Compare predictions with sentiment; generate summary reports.
- **Business Understanding:** Define the goal of personalized movie recommendations.
- **Deployment:** Present results in a simple Netflix-style popup UI.

REPORTS GENERATED

We generated multiple analytical reports to evaluate system behavior. These reports helped us understand user taste patterns, movie sentiment trends, and alignment between model predictions and public opinion.

- Top 10 movies recommended
- Sentiment score distribution
- Genre-wise sentiment averages
- Rating vs sentiment comparison
- Similar movies (genre-based recommendations)

=== TOP 10 RECOMMENDATIONS FOR USER ===

userId	recommendations
118205	[{129401, 5.360338}, {128812, 5.360338}, {128091, 5.360338}, {34464, 5.1455207}, {82055, 5.1394076}, {117907, 4.9727607}, {112423, 4.7676616}, {101538, 4.6797833}, {107623, 4.667629}, {102107, 4.667629}]

=== SENTIMENT SCORE SAMPLE ===

title	sentiment_score
Toy Story (1995)	0.8300000000000001
Jumanji (1995)	0.7
Grumpier Old Men (1995)	0.67
Waiting to Exhale (1995)	0.6
Father of the Bride Part II (1995)	0.61
Heat (1995)	0.82
Heat (1995)	0.5599999999999999
Sabrina (1995)	0.63
Tom and Huck (1995)	0.55
Sudden Death (1995)	0.58

only showing top 10 rows

=== GENRE SENTIMENT AVERAGES ===

genre	avg(sentiment_score)
Crime	0.5225584547610982
Thriller	0.5207415355269431
Romance	0.5205504587155962
Fantasy	0.5198797736916548
Drama	0.5198363717872085
Action	0.5186915887850471
Comedy	0.5184328447145754
Adventure	0.5181370449678802
Sci-Fi	0.5181121281464529
Mystery	0.5175263157894737
Children	0.5165118317265555
Film-Noir	0.5155454545454545
War	0.5149539748953974
Animation	0.5133852140077821
Musical	0.5116104146576664
Horror	0.5115372848948373
Western	0.5106213017751479
IMAX	0.5067346938775511
(no genres listed)	0.5023983739837398
Documentary	0.5007365439093484

=== RANDOM MOVIE RATING & SENTIMENT COMPARISON ===

title	sentiment_score	avg(rating)
Dracula: Dead and Loving It (1995)	0.5900000000000001	2.616869918699187
Toy Story (1995)	0.8300000000000001	3.9121305775545685
Balto (1995)	0.71	3.2734584450402147
Heat (1995)	0.82	3.839922543166048
Heat (1995)	0.5599999999999999	3.839922543166048
Grumpier Old Men (1995)	0.67	3.1563251201923075
Father of the Bride Part II (1995)	0.61	3.06810551558753
Sudden Death (1995)	0.58	3.0253225035833733
Sudden Death (1995)	0.48	3.0253225035833733
Waiting to Exhale (1995)	0.6	2.8401384083044983
Tom and Huck (1995)	0.55	3.1726190476190474
Sabrina (1995)	0.63	3.3731299066804916
GoldenEye (1995)	0.72	3.4192730910303433
American President, The (1995)	0.5	3.674496288441145
Nixon (1995)	0.71	3.437167032278283

RECOMMENDATIONS FOR C-LEVEL EXECUTIVES

Based on our analysis, a hybrid recommendation system that combines user-behavior modeling (ALS) with sentiment analysis offers measurable business value. It enhances user engagement, improves content discoverability, and provides a scalable foundation for personalized experiences across digital platforms.

Invest in Hybrid Recommendation Systems
Combining collaborative filtering with sentiment analysis leads to higher user satisfaction, more accurate suggestions, and greater platform stickiness.

Adopt Big-Data Infrastructure (Spark/Cloud)
Distributed systems like Spark are essential for processing millions of user interactions in real time, ensuring scalability as the user base grows.

Leverage Metadata & User Signals Strategically
Incorporating public sentiment, genre preferences, and user patterns improves the relevance and trustworthiness of recommendations.

Enhance User Experience with Lightweight, Personalized Interfaces
Pop-up or embedded recommendation modules increase discoverability and reduce decision fatigue, boosting retention and watch-time metrics.

INSIGHTS & FINDINGS

Combining ALS and sentiment offers powerful insights. We observed that personalized recommendations often differ from general public opinion, highlighting the importance of hybrid models.

- ALS strongly personalizes based on user history
- Popular movies ≠ recommended movies for every user
- Sentiment adds a second layer of validation
- Some genres consistently exhibit high sentiment
- Missing metadata affects recommendation depth

CHALLENGES & OPPORTUNITIES

While the system performed well, several improvements could enhance its accuracy, scalability, and user experience in future iterations.

- ALS training slow at full scale on Colab
- Metadata not available for all movies
- Cold-start persists for new users/movies
- Future: NLP sentiment from reviews
- Future: Deploy model on Spark cluster or Databricks

CONCLUSION

This project demonstrated how big-data technologies can be integrated to build a scalable and intelligent movie recommendation system. By combining Spark's ALS collaborative filtering with sentiment information from TMDB metadata, we created a hybrid model capable of generating personalized, sentiment-aware recommendations at scale. The system not only processes millions of ratings efficiently but also delivers meaningful insights through analytical reports and a user-friendly interface. Overall, the project showcases the power of distributed data processing and hybrid modeling in solving real-world recommendation challenges, while laying a strong foundation for future extensions such as NLP-based sentiment analysis, richer metadata integration, and deployment on cloud-based big-data platforms.

REFERENCES

- **GroupLens. (2015). *MovieLens 20M Dataset*.** https://grouplens.org/datasets/movielens/
- **Kaggle. (2024). *Movies Updated Data* [Dataset].** https://kaggle.com/datasets/ashishkumarjayswal/movies-updated-data
- **The Movie Database (TMDB). (2024). *Movie Metadata API*.** https://www.themoviedb.org/
- **Zhou, Y., Wilkinson, D., Schreiber, R., & Pan, R. (2008).** Large-scale parallel collaborative filtering for the Netflix Prize. *Proceedings of the 4th International Conference on Algorithmic Aspects in Information and Management.*
- **Apache Spark. (2024). *MLlib Collaborative Filtering (ALS) Documentation*.** https://spark.apache.org/docs/latest/ml-collaborative-filtering.html
- **KaggleHub. (2024). *Dataset Integration API Documentation*. https://github.com/KaggleHub**