FlixShield: Netflix Content Intelligence & Recommendation System

- **Presented by:**
  Anju Yadav
  Roll No.: 242123002
- **Course: DA626 – Recommendation System Using Deep Learning**
  Instructor: Dr. Chiranjib Sur
- **Indian Institute of Technology Guwahati 2025**

Problem Statement

- Netflix hosts thousands of Movies and TV Shows with diverse genres, countries, ratings, and descriptions. This large volume of content makes it difficult to understand global distribution patterns and to recommend relevant titles to users. Additionally, the metadata is highly unstructured—containing multi-valued fields, varying formats, and long textual descriptions—which complicates analysis.
- Therefore, the key problem addressed in this project is:
- **How can we intelligently analyze Netflix’s unstructured metadata using NLP and machine learning to identify content patterns, validate trends statistically, and generate accurate content-based recommendations for users?**

Dataset Description

- The project uses the **Netflix Movies and TV Shows dataset**, which contains **8,807 records** of titles available on the Netflix platform. Each entry includes rich metadata describing various attributes of the content. The dataset consists of the following key fields:
- **show_id** – Unique identifier for each title
- **type** — Movie or TV Show
- **title** — Name of the content
- **director** — Director(s) of the title
- **cast** — Actors involved
- **country** — Country of production
- **date_added** — When the title was added to Netflix
- **release_year** — Year of original release
- **rating** — Content rating (e.g., TV-MA, PG, R)
- **duration** — Movie runtime (minutes) or number of seasons
- **listed_in** — Genre(s) assigned
- **description** — Summary text of the title
- The dataset is suitable for **EDA, NLP processing, clustering, and recommendation modeling** because it contains both structured fields (type, rating, country) and unstructured text (description, cast, director, genre). After unnesting multi-valued categories like cast, genres, and directors, the dataset expands to nearly **200,000 rows**, enabling deeper analysis of metadata relationships.
- The dataset used in this project was obtained from **Kaggle**, titled **“Netflix Movies and TV Shows Dataset.”**

Architecture Overview

1. Dataset Input (Raw Netflix Data)

12-column metadata extracted from the Kaggle Netflix Movies & TV Shows dataset, containing attributes such as *title, type, director, cast, country, date added, release year, rating, duration, genres,* and *description.*

2. Data Preprocessing Layer

3. Text Processing Layer (NLP Pipeline)

4. Feature Extraction Layer

5. Machine Learning Layer (Clustering)

6. EDA & Statistical Analysis Layer

7. Recommendation Engine Layer

8. Output Layer

Data Preprocessing

- **Handle missing values:** Replaced null entries using mode/“Unknown” and dropped rows with minimal missing data.
- **Unnest multi-valued fields:** Split and expanded cast, directors, genres, and country for detailed analysis.
- **Clean duration, date, type:** Standardized duration to numeric, converted dates to datetime, and formatted categorical fields.
- **Convert values into usable formats:** Transformed text and numeric data into structured, ML-friendly formats.

Text Processing

- **Cleaning and normalization:** Removed punctuation, digits, URLs, and standardized text to lowercase for uniformity.
- **Stopword removal:** Eliminated frequently occurring non-informative words to enhance meaningful signal.
- **Tokenization:** Split text into individual words/tokens for structured NLP processing.
- **Lemmatization:** Converted words to their base forms to reduce vocabulary and improve model accuracy.
- **POS tagging:** Identified grammatical roles of words to extract contextual and semantic information.
- **Creation of content_detail:** Combined title, director, cast, genre, and description into a single rich text field for vectorization.

Feature Extraction Layer

- **TF-IDF Vectorization:** Transforms cleaned text (title, cast, director, genre, description) into numerical vectors that represent word importance across the dataset. This helps the model understand semantic similarity between different shows and movies.
- **PCA Dimensionality Reduction:** Reduces thousands of TF-IDF dimensions into a smaller set of principal components, removing noise while preserving meaningful patterns. This boosts processing speed and enhances clustering quality.

Machine Learning Layer (K-Means Clustering)

- **K-Means Clustering:** Groups Netflix titles into clusters based on similarity of content features (TF-IDF vectors reduced using PCA).
- Produces **theme-based clusters** such as horror, documentaries, teen dramas, stand-up comedy, etc.
- **Elbow Method:** Used to choose the optimal number of clusters (k) by plotting WCSS (Within-Cluster Sum of Squares) and finding the “elbow point,” where adding more clusters provides minimal improvement.
- Ensures a **balanced, stable, and meaningful clustering structure** for downstream analysis and recommendations.

Exploratory Data Analysis

- **Movies vs TV:** Analyzes the proportion of movies and TV shows available on Netflix to understand platform content focus.
- **Country-wise Spread:** Identifies which countries contribute the most titles and how content is globally distributed.
- **Genre Distribution:** Examines the most common genres to reveal audience preferences and platform trends.
- **Trend Over Years:** Observes how Netflix's content library has grown annually and which years saw major additions.

Statistical Analysis Layer

- Performs **hypothesis testing** (Z-test, T-test, Proportion tests) to verify statistically significant differences across ratings, content duration, and country-level patterns.
- Helps validate insights found in EDA—such as whether drama vs comedy ratings differ, or whether TV show durations changed between 2020 and 2021.
- Ensures conclusions are **evidence-backed**, not just visually interpreted.

Recommendation System Results

•**TF-IDF + Cosine Similarity:**
Converts each title's *description, genre, cast,* and *keywords* into numerical
•vectors and calculates similarity between all Netflix titles.
•**How It Works:**
Measures how closely two shows match in terms of themes, storyline, language
• patterns, and contextual meaning—without needing user history.
•**Strength:**
Works even for *new or niche* titles because it relies on content rather than use
•r interactions.
•**Example (Indian Title):**
Input **"Kota Factory"** → Recommends shows like **"Yeh Meri Family"**,
•**"Girls Hostel"**, **"Engineering Girls"** because they share
✓ youth-centric narrative
✓ college/school life
✓ middle-class Indian setting
✓ relatable emotional tone
•**Outcome:**
Delivers *personalized, theme-aware* recommendations that help users discover
• content similar in mood and storyline.

Conclusion

•**Comprehensive Content Analysis:**
The project successfully analyzes Netflix content across genres, ratings, countries, and temporal trends, uncovering meaningful insights into viewing patterns and platform diversity.

•**Working Recommendation Engine:**
A fully functional content-based recommender system is built using TF-IDF and cosine similarity, capable of generating accurate, theme-aligned recommendations even for region-specific titles.

•**End-to-End Pipeline:**
From preprocessing and feature engineering to clustering, hypothesis testing, and recommendation, the project demonstrates a complete OTT content intelligence workflow.

Future Work

- **Using Deep Learning Embeddings :**

Implementing BERT or Sentence Transformers to capture richer semantic meaning beyond TF-IDF.

- **Hybrid Recommendation System :**

Combining content-based features with user behavior data to improve personalization.

- **Advanced Clustering Techniques:**

Exploring algorithms like HDBSCAN or spectral clustering for more refined content grouping.

- **Incorporating Sentiment or Review Data:**

Adding user review analysis to strengthen recommendations and content insights

Bibliography

[1] Scikit-learn Documentation. Available: https://scikit-learn.org

[2] NLTK: Natural Language Toolkit. Available: https://www.nltk.org

[3] spaCy Industrial NLP Library. Available: https://spacy.io

[4] Kaggle Netflix Dataset. Available: https://www.kaggle.com