Suggesteria: Netflix Recommendation System

By

Aniket Prakash bhagat 241250008

Pranjal Bopate 241250036

Introduction

The project is to develop a recommendation system similar to Netflix that will recommend 5 movies or TV shows based on the selected title. This system is created using content-based-filtering and deployed using Streamlit web application.

1. Dataset Information -

Source: Netflix_titles.csv

Column Name	Description	Example Values
show_id	Unique identifier for each title.	s1, s2, s3
type	Indicates whether the content is a Movie or TV Show.	Movie, TV Show
title	Name of the movie or TV show.	Stranger Things, Inception
director	Names of the directors. If multiple, they are separated by commas.	Christopher Nolan, Steven Spielberg
cast	List of main actors in the title.	Leonardo DiCaprio, Joseph Gordon-Levitt
country	The country where the movie or show was produced.	United States, India, United Kingdom
date_added	The date when the title was added to Netflix.	September 25, 2021
release_year	The year when the movie/show was released.	2010, 2022
rating	The content rating, indicating age suitability.	PG-13, R, TV-MA, G
duration	Length of the movie in minutes or number of seasons for TV shows.	120 min, 2 Seasons

listed_in	The genre or category of the content.	Drama, Thriller, Comedy, Action
description	A brief summary of the movie/show.	A thief enters dreams to steal secrets

.

- Preprocessing Steps:
  - Handled missing values by replacing them with empty strings.
  - Combined multiple features (title, description, director, cast, listed_in) into a single text column.
  - Converted text data to lowercase and removed special characters.

2. Algorithms Implemented

Concept

Content-based filtering recommends items that are similar to what the user has previously liked or interacted with. It does so by analysing item features and calculating similarity scores.

Implementation Steps

- Feature Engineering:
  - We combined key features such as title, description, director, cast, and listed genres into a single text column.
  - Text data was pre-processed by lowercasing, removing stop words, and applying lemmatization.
- TF-IDF Vectorization:
  - We used Term Frequency-Inverse Document Frequency (TF-IDF) to convert text data into numerical feature vectors.
  - This technique helps in giving importance to unique words while downplaying commonly occurring words.
- Cosine Similarity Computation:
  - We computed the cosine similarity between the TF-IDF vectors of all items.
  - Cosine similarity measures the angle between two vectors, indicating how similar two items are.

- Recommendation Retrieval:
    - For a given input movie, we retrieved the top-N most similar movies based on cosine similarity scores.

3. Challenges & Solutions

CHALLENGE	SOLUTION
LARGE DATASET SIZE	Filtered out irrelevant columns
MISSING VALUES	Used .fillna('') method to handle NaNs
GITHUB FILE SIZE LIMITS	Used Git LFS to track large .pkl files
STREAMLIT DEPLOYMENT ERRORS	Fixed missing dependencies in requirements.txt

4. Streamlit Web Application

Features:

- User Input: Users enter a movie title.
- Real-Time Recommendations: Displays top-5 recommended movies.
- Movie Posters: Integrated TMDb API for fetching images.

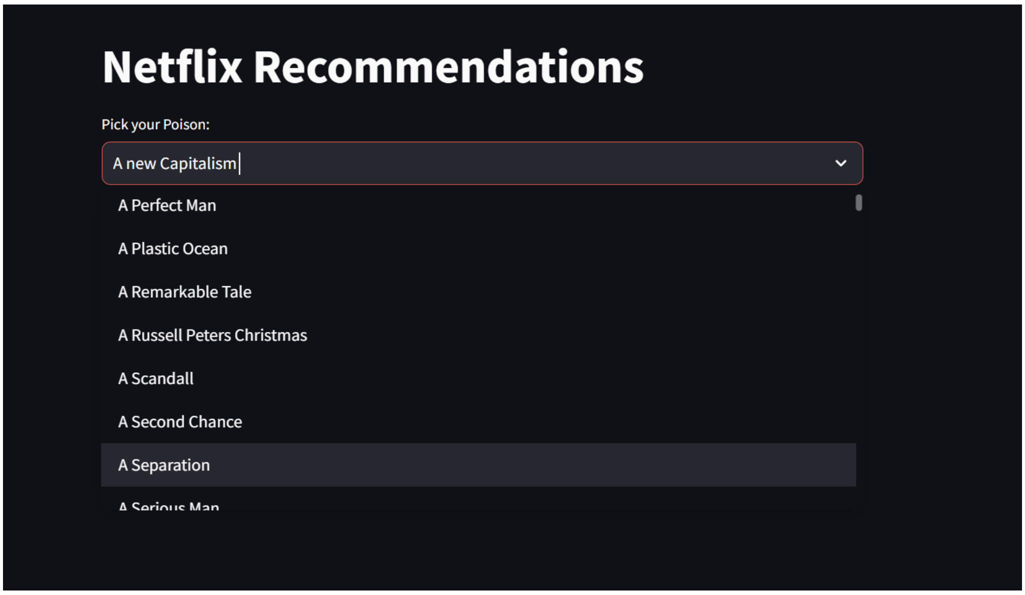
Deployment Link:

Netflix Recommender App

Git-hub

5. Demonstration

- Select the Movie from the drop-down



Netflix Recommendations

Pick your Poison:

A new Capitalism

A Perfect Man

A Plastic Ocean

A Remarkable Tale

A Russell Peters Christmas

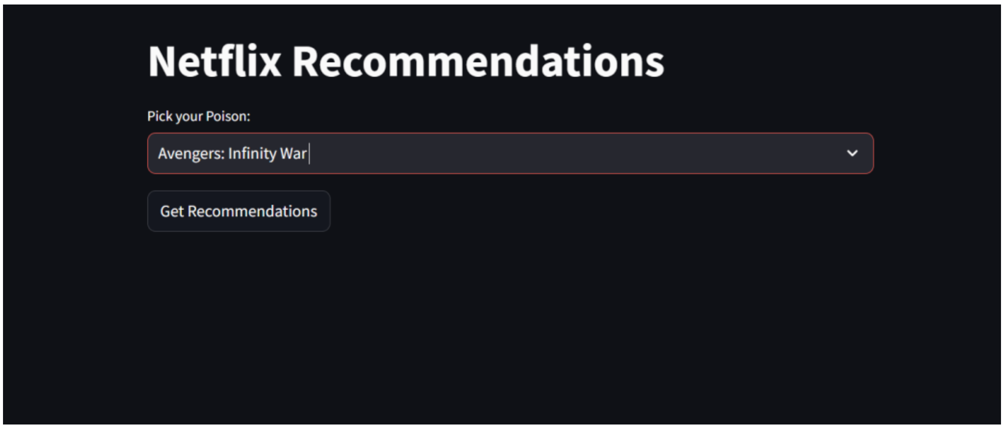
A Scandall

A Second Chance

A Separation

A Serious Man

- Click on the “Get Recommendation button

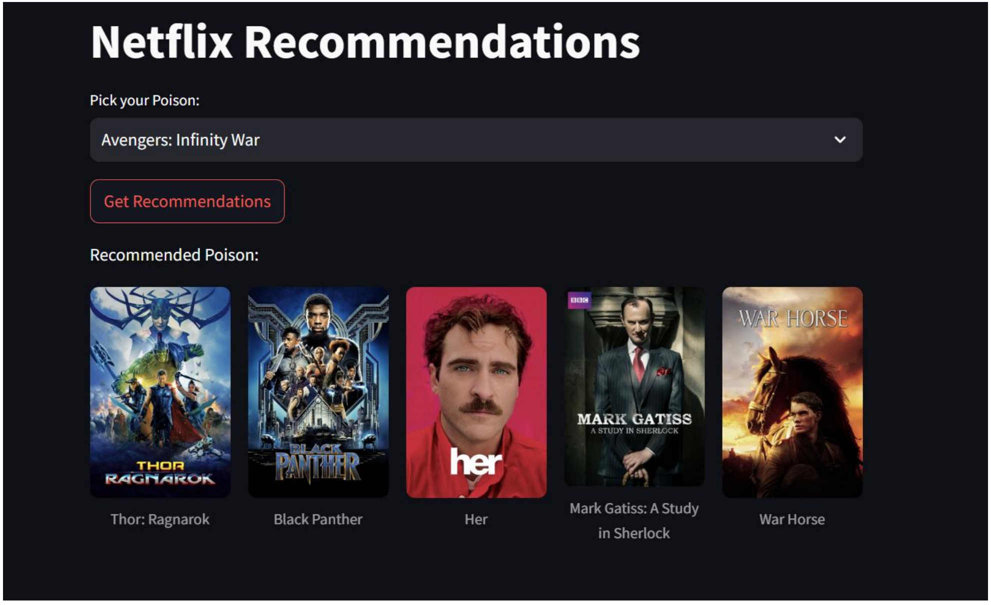


Netflix Recommendations

Pick your Poison:

Avengers: Infinity War

- The top 5 recommendations will be shown –



Netflix Recommendations

Pick your Poison:

Avengers: Infinity War

Recommended Poison:

Thor: Ragnarok

Black Panther

Her

Mark Gatiss: A Study in Sherlock

War Horse