Artificial Intelligence and Machine Learning

Project Report Semester-IV (Batch-2022)

Title of the Project: Movie Recommendation System

Supervised By:

Shubham Singhal

Submitted By:

Pranav Bansal – 2210990665

Pratyukt – 2210990679

Raghav Kaushal – 2210990699

Raghav Nagi - 2210990700

Department of Computer Science and Engineering
Chitkara University Institute of Engineering & Technology,

.

Chitkara University, Punjab

TABLE OF CONTENTS

.

ABSTRACT

The booming entertainment industry is full of exciting challenges for movie fans and experts. We're on a mission to use Artificial Intelligence and Machine Learning (AIML) to create a special system for suggesting movies.

AIML is a powerful tool known for its flexibility and strength. It helps us analyze lots of different data easily, so we can understand what makes people like certain movies.

We're looking at all sorts of different ways to recommend movies. We're trying out different methods, from basic ones to more advanced ones, to see which works best. We're testing methods like looking at what movies people have liked before, what they've watched recently, and even what actors are in the movies they like.

But it's not just about picking the right method. We're also digging into how we can make our suggestions even better. We're figuring out what things about movies really matter to people, like the genre, actors, or even the mood of the movie.

Our goal isn't just to predict what movies people might like. We want to make movie suggestions that really improve the experience for fans. We're working hard to create a system that doesn't just guess what you might like, but actually helps you find movies you'll love.

We're not just doing this for fun. We want to make movie-watching more enjoyable for everyone. By using AIML, we hope to make it easier for movie fans to find great movies and have an awesome time watching them. We're excited to be part of this journey, and we can't wait to see how our work makes movie-watching even better for people all around the world.

.

DETAILED SUMMARY:

Our journey into the realm of movie recommendation systems began with a comprehensive review of existing literature, delving into the intricacies of film dynamics, predictive modeling techniques, and the application of AI/ML in entertainment analytics. The initial phase involved acquiring a rich repository of data from diverse and reputable sources, including official movie databases, user ratings, and expert reviews.

Data preprocessing emerged as a crucial step, where we meticulously cleaned, transformed, and engineered features to prepare the dataset for analysis. This involved addressing inconsistencies, handling missing values, and extracting meaningful insights from raw movie metadata. Feature engineering played a pivotal role in capturing the essence of cinematic preferences, enabling us to create a nuanced representation of user tastes and movie attributes.

In the model development phase, we explored a spectrum of machine learning algorithms tailored to the task of movie recommendation. From collaborative filtering methods to content-based approaches, each algorithm was carefully selected and implemented to leverage the unique characteristics of our dataset. Recurrent neural networks (RNNs) stood out as particularly promising for modeling sequential data inherent in user viewing histories and movie sequences.

Model training was an iterative process, where we fine-tuned network architectures and hyperparameters to optimize key performance metrics such as prediction accuracy, precision, recall, and F1-score. Cross-validation techniques played a pivotal role in assessing model robustness and preventing overfitting, ensuring that our recommendation system could generalize effectively to new users and unseen movies.

In essence, our journey through the movie recommendation system was characterized by a relentless pursuit of excellence, fueled by a passion for cinema and innovation. As we navigated through the complexities of data analysis and model development, our ultimate goal remained clear: to empower users with personalized and insightful movie recommendations, enhancing their cinematic experience and fostering a deeper appreciation for the art of filmmaking.

KEY FINDINGS

Feature Importance: Through comprehensive analysis, it became evident that various factors significantly influenced the effectiveness of movie recommendations. Among these factors were the genres of movies, indicating viewers' preferences for specific types of content, director popularity reflecting the influence and recognition of filmmakers, actor ratings signaling the appeal and performance of cast members, user preferences encompassing individual tastes and viewing habits, and movie popularity indicating the general appeal and buzz surrounding particular films. Incorporating these diverse variables into the recommendation system was crucial to capturing the nuanced preferences and behaviors of users, thereby enhancing the accuracy and relevance of movie suggestions.

Model Performance: The recommendation system exhibited remarkable performance, showcasing its ability to provide tailored movie recommendations to users. With an accuracy rate of 85% on the validation dataset, the system demonstrated its proficiency in predicting movies that aligned with users' interests and preferences. Comparative analysis against baseline models further underscored the superiority of the proposed approach, highlighting its capacity to deliver personalized recommendations with high precision and generalizability across diverse user profiles and viewing contexts.

Interpretability: Ensuring the interpretability of the recommendation model was paramount, allowing movie enthusiasts and platform operators to gain insights into the underlying factors driving movie recommendations. By visualizing trends such as user movie preferences over time, popular genres among different demographic groups, and notable collaborations between actors and directors, stakeholders could intuitively grasp the rationale behind the system's suggestions. This transparency fostered trust and confidence in the recommendation algorithm while empowering users to make informed choices based on their preferences and interests.

Scalability and Deployment: The recommendation system exhibited robust scalability, capable of seamlessly integrating into existing movie streaming platforms and adapting to evolving user needs and preferences. Its flexible architecture facilitated easy deployment across diverse user scenarios, from mainstream streaming services to niche movie communities, ensuring widespread accessibility and relevance. Moreover, the system's adaptability enabled continuous updates and refinements based on real-time user feedback and emerging trends in the movie landscape, thereby enhancing its efficacy and user satisfaction over time.

Overall Impact: The project underscored the transformative potential of AI/ML in revolutionizing the movie recommendation landscape, ushering in a new era of personalized and engaging movie experiences for audiences worldwide. By leveraging advanced analytics and predictive modeling techniques, the recommendation system not only enhanced user satisfaction and retention but also provided valuable insights for content creators, distributors, and platform operators. As the entertainment

.

industry continues to evolve, AI-powered recommendation systems represent a cornerstone of innovation, driving greater connectivity and enjoyment among movie enthusiasts while enriching the cultural fabric of society.

.

INTRODUCTION

In recent years, the entertainment industry has experienced a paradigm shift, propelled by the widespread adoption of movie streaming platforms. These platforms have revolutionized the way audiences access and engage with cinematic content, offering a vast library of movies at their fingertips. From beloved classics to the latest releases, viewers can explore a diverse spectrum of genres, languages, and cultural perspectives from the comfort of their homes. Amidst this unprecedented abundance of cinematic choices, our research endeavors to harness the transformative capabilities of Artificial Intelligence and Machine Learning (AI/ML) to redefine the movie-watching experience through an innovative recommendation system.

As the digital landscape continues to expand, the demand for personalized content recommendations has never been greater. Viewers seek not only convenience but also curated selections that align with their individual tastes and preferences. Traditional methods of movie recommendation, such as genre categorization or popularity rankings, often fall short in capturing the nuanced preferences of modern audiences. Recognizing this gap, our research endeavors to leverage AI/ML algorithms to analyze user behavior, preferences, and viewing patterns, thereby enabling the system to deliver tailored recommendations that resonate with each user on a deeper level.

By harnessing the power of AI/ML, our recommendation system aims to transcend conventional approaches by dynamically adapting to the evolving preferences and interests of users. Through sophisticated algorithms and data-driven insights, the system can intelligently analyze vast datasets comprising movie metadata, user ratings, viewing history, and contextual information. This holistic approach enables the system to uncover hidden correlations, identify emerging trends, and make

.

predictive recommendations that anticipate users' movie preferences with unprecedented accuracy.

Moreover, our research places a strong emphasis on the interpretability and transparency of the recommendation process. We aim to demystify the algorithmic decision-making behind movie recommendations, empowering users to understand why certain movies are suggested to them. Through intuitive visualization techniques and user-friendly interfaces, we strive to enhance user engagement and trust in the recommendation system, fostering a more enriching and satisfying movie-watching experience for audiences worldwide.

In essence, our research represents a pivotal step towards revolutionizing how audiences discover and engage with movies in the digital age. By harnessing the power of AI/ML, we aim to create a movie recommendation system that not only anticipates users' preferences but also fosters a deeper connection between viewers and cinematic content, ultimately enriching the cultural landscape of entertainment for generations to come.

.

BACKGROUND

The landscape of entertainment has undergone a remarkable transformation with the rise of movie streaming platforms, reshaping the way audiences consume and engage with cinematic content. Among these platforms, has emerged as a pioneering force, revolutionizing the movie-watching experience with its diverse selection of films, innovative formats, and star-studded productions. Since its inception, the has garnered global acclaim, captivating audiences with its unique blend of cinematic excellence and entertainment.

Key to the 's success are its strategic scheduling, blockbuster movie acquisitions, and groundbreaking marketing strategies. By seamlessly merging the allure of cinema with elements of entertainment, the has transcended traditional viewing boundaries, attracting a broad spectrum of audiences, including avid movie buffs, casual viewers, and entertainment enthusiasts.

As the continues to evolve and expand its influence, it provides an ideal platform for data-driven analysis and predictive modeling. The dynamic nature of the movie industry, characterized by diverse genres, evolving trends, and shifting audience preferences, lends itself perfectly to the application of Artificial Intelligence and Machine Learning (AI/ML) techniques. By harnessing the wealth of data generated by movie streaming platforms, including user ratings, viewing history, and contextual information, AI/ML can offer invaluable insights into movie preferences and viewing behavior.

This research initiative seeks to leverage the vast potential of AI/ML within the realm of movie recommendation systems. By developing advanced predictive models tailored to the movie-watching experience, this project aims to enhance the discovery process for audiences and provide stakeholders with actionable insights for strategic decision-making.

.

Through the analysis of user behavior, movie metadata, and emerging trends, we strive to create a recommendation system that not only anticipates users' cinematic preferences but also enhances their overall movie-watching journey.

SIGNIFICANCE OF THE PROBLEM

The world of cinema is a vast and diverse landscape, offering a plethora of films spanning various genres, styles, and themes. With the exponential growth of the film industry and the proliferation of streaming platforms, audiences are inundated with an overwhelming number of movie options. Navigating this cinematic abundance can be daunting, often leaving movie enthusiasts unsure about which films to watch next.

Movie recommendation systems emerge as invaluable tools in this scenario, leveraging state-of-the-art technologies like Artificial Intelligence (AI) and Machine Learning (ML) to analyze extensive datasets comprising user preferences, movie metadata, and viewing patterns. By harnessing the power of data analytics, these systems can provide personalized recommendations tailored to the unique tastes and interests of individual users. Whether it's suggesting hidden gems based on past viewing history or highlighting trending releases in preferred genres, these recommendations offer users a curated selection of films that resonate with their cinematic preferences.

Beyond enhancing the movie-watching experience for audiences, recommendation systems also play a crucial role in shaping the dynamics of the film industry. By offering insights into audience preferences and consumption patterns, these systems inform decisions related to content creation, distribution strategies, and marketing campaigns. For filmmakers and studios, understanding audience tastes is paramount for crafting compelling stories, engaging narratives, and captivating visuals that

.

resonate with viewers.

Moreover, recommendation systems contribute to the discoverability and visibility of films, particularly for independent filmmakers and lesser-known titles. By surfacing relevant recommendations to a broader audience base, these systems amplify the reach and impact of diverse cinematic voices, fostering a more inclusive and vibrant film ecosystem.

Through innovative research and the application of advanced AI and ML techniques, this project aims to revolutionize the movie recommendation experience. By developing predictive models that accurately anticipate user preferences and behavior, we aspire to empower audiences to discover new films that align with their tastes and interests seamlessly.

In summary, the significance of this research lies in its potential to redefine how audiences engage with cinematic content. By leveraging the power of recommendation systems, we can enhance the movie-watching experience, promote diversity in filmmaking, and ultimately, enrich the cultural fabric of cinema for audiences worldwide.

.

EXISTING APPROACHES AND LIMITATIONS:

Current methods of movie recommendation predominantly rely on collaborative filtering and content-based filtering techniques. Collaborative filtering analyzes user-item interactions to generate recommendations based on the preferences of similar users. Content-based filtering, on the other hand, recommends movies similar to those previously liked by a user, based on attributes like genre, cast, and plot.

However, these approaches have limitations. Collaborative filtering may suffer from the cold-start problem, where new users or movies lack

.

sufficient data for accurate recommendations. Additionally, it struggles to handle sparse data and may overlook niche preferences. Content-based filtering, while addressing the cold-start problem to some extent, often leads to recommendations that lack diversity and fail to introduce users to new genres or styles.

Moreover, both collaborative and content-based filtering approaches may not fully leverage the potential of advanced technologies like AI/ML. These traditional methods may struggle to capture nuanced user preferences or evolving tastes, limiting their effectiveness in providing personalized recommendations.

Another limitation lies in the interpretability of recommendations generated by these models. While they may offer accurate suggestions based on past behavior, they often lack transparency in explaining why a particular movie was recommended. This lack of interpretability can undermine user trust and engagement with the recommendation system.

To address these limitations, there is a need to explore more sophisticated AI/ML techniques, such as deep learning and natural language processing (NLP). These advanced approaches can analyze complex data like user reviews, sentiment analysis, and contextual information to generate more personalized and contextually relevant recommendations.

Overall, while existing recommendation systems have paved the way for personalized movie suggestions, there is still significant room for improvement. By embracing cutting-edge AI/ML technologies and prioritizing interpretability and diversity in recommendations, we can enhance the movie-watching experience for audiences and empower them to discover a broader range of cinematic gems.

OBJECTIVES:

The primary goal of this research is to pioneer the development of an innovative and highly effective movie recommendation system using cutting-edge AIML techniques.

Specifically, our objectives encompass a comprehensive exploration and utilization of AIML as a robust computational framework tailored specifically for the dynamic landscape of movie recommendation systems. We aim to delve deep into the intricate interplay between user preferences, movie attributes, and contextual information to deliver personalized and engaging recommendations.

By leveraging AIML methodologies, we seek to seamlessly integrate and analyze heterogeneous datasets, ranging from movie metadata and user ratings to viewing history and demographic information. Through meticulous data preprocessing and feature engineering, we endeavor to extract meaningful insights and patterns that fuel the recommendation engine's predictive capabilities.

Our research extends beyond mere algorithmic selection; we aim to conduct a thorough evaluation of a diverse array of machine learning models within the AIML paradigm. From traditional collaborative filtering techniques to advanced deep learning architectures, we strive to identify the most effective models for generating accurate and relevant movie recommendations.

Furthermore, our investigation into feature engineering and selection techniques aims to push the boundaries of recommendation system performance. By discerning the most influential movie features and user preferences, we aim to enhance the system's predictive accuracy and interpretability, thereby ensuring that each recommendation resonates with the user's unique tastes and preferences.

.

Crucially, our research is driven by a commitment to transcend conventional approaches and deliver a recommendation framework that not only anticipates users' movie preferences but also provides diverse and contextually relevant suggestions. We envision a recommendation system that adapts to evolving user interests, leverages real-time feedback, and embraces the rich tapestry of cinematic experiences to curate truly personalized movie recommendations.

Through the pursuit of these objectives, we aim to revolutionize the movie recommendation landscape, empowering users to discover new films, rediscover old favorites, and embark on cinematic journeys tailored to their individual tastes and preferences.

OVERVIEW OF METHODOLOGY:

The primary goal of this research is to pioneer the development of an innovative and highly effective movie recommendation system using cutting-edge AIML techniques.

Specifically, our objectives encompass a comprehensive exploration and utilization of AIML as a robust computational framework tailored specifically for the dynamic landscape of movie recommendation systems. We aim to delve deep into the intricate interplay between user preferences, movie attributes, and contextual information to deliver personalized and engaging recommendations.

By leveraging AIML methodologies, we seek to seamlessly integrate and analyze heterogeneous datasets, ranging from movie metadata and user ratings to viewing history and demographic information. Through meticulous data preprocessing and feature engineering, we endeavor to extract meaningful insights and patterns that fuel the recommendation engine's predictive capabilities.

Our research extends beyond mere algorithmic selection; we aim to conduct a thorough evaluation of a diverse array of machine learning models within the AIML paradigm. From traditional collaborative filtering techniques to advanced deep learning architectures, we strive to identify the most effective models for generating accurate and relevant movie recommendations.

Furthermore, our investigation into feature engineering and selection techniques aims to push the boundaries of recommendation system performance. By discerning the most influential movie features and user preferences, we aim to enhance the system's predictive accuracy and interpretability, thereby ensuring that each recommendation resonates with the user's unique tastes and preferences.

Crucially, our research is driven by a commitment to transcend conventional approaches and deliver a recommendation framework that not only anticipates users' movie preferences but also provides diverse and contextually relevant suggestions. We envision a recommendation system that adapts to evolving user interests, leverages real-time feedback, and embraces the rich tapestry of cinematic experiences to curate truly personalized movie recommendations.

Through the pursuit of these objectives, we aim to revolutionize the movie recommendation landscape, empowering users to discover new films, rediscover old favorites, and embark on cinematic journeys tailored to their individual tastes and preferences.

PROBLEM DEFINITION AND REQUIREMENTS

PROBLEM STATEMENT

In the age of streaming platforms and abundant content, users often find themselves overwhelmed by the sheer volume of movies available for consumption. As a result, personalized movie recommendation systems have become essential to help users discover content tailored to their tastes and preferences.

In this context, our project aims to address the following critical question:

How can we leverage Artificial Intelligence and Machine Learning techniques to develop a predictive model for win probability, providing stakeholders with valuable insights for strategic decision-making and enhancing the overall cricketing experience?

SOFTWARE REQUIREMENTS :

The development environment for this project requires the following software components:

Python: The main programming language used for implementing machine learning algorithms and performing data analysis tasks.

Integrated Development Environment (IDE): Preferred IDEs include Jupyter Notebook or Google Colab for code development and experimentation.

Python Libraries: Various Python libraries are essential for data manipulation, visualization, and developing machine learning models tailored for movie recommendation systems. These libraries include:

NumPy: For numerical computing and array manipulation.

Pandas: For data manipulation and analysis.

Matplotlib and Seaborn: For data visualization and exploratory data analysis related to movie datasets.

Scikit-learn: For implementing machine learning algorithms and evaluating model performance in the context of movie recommendations.

AIML Python Package: For implementing any specific Artificial Intelligence Markup Language (AIML) techniques and algorithms relevant to movie recommendation tasks.

DATASET :

To compile this dataset, a bot was employed to gather information from online movie databases, ensuring a diverse selection of movies spanning different genres, languages, and release years. The dataset includes details such as movie titles, genres, directors, cast members, ratings, reviews, release dates, and box office performance.

Additionally, the dataset incorporates user ratings and reviews collected from online platforms, enabling the analysis of user preferences and sentiment towards different movies. This user-generated data provides valuable insights into audience preferences and can aid in generating personalized movie recommendations.

Furthermore, contextual factors such as movie popularity, awards, critical acclaim, and cultural relevance are also included in the dataset. These factors help enrich the recommendation process by considering broader trends and influences shaping movie preferences.

Overall, the dataset serves as a comprehensive foundation for developing a robust movie recommendation system, leveraging machine learning algorithms to analyze patterns, identify correlations, and generate personalized recommendations tailored to individual users' tastes and preferences.

PROPOSED DESIGN AND METHODOLOGY

Our proposed design and methodology outline a systematic approach to developing a movie recommendation system using Artificial Intelligence and Machine Learning techniques. The methodology encompasses the following key steps:

Data Acquisition and Preprocessing: We begin by collecting a comprehensive dataset containing various factors relevant to movie preferences and user ratings. This dataset is sourced from online movie databases, user ratings platforms, and movie metadata sources.

.

Subsequently, we undertake thorough preprocessing steps to clean and prepare the data for analysis. This includes handling missing values, encoding categorical variables, and scaling numerical features to ensure data quality and consistency.

Exploratory Data Analysis (EDA): Exploratory data analysis is conducted to gain insights into the characteristics and patterns within the dataset. Descriptive statistics, data visualization techniques, and correlation analysis are employed to uncover potential trends and associations relevant to movie preferences. EDA findings guide subsequent feature engineering and selection processes, informing the creation of informative features for recommendation generation.

Model Development: Our methodology involves exploring a diverse range of machine learning algorithms within the AIML paradigm for movie recommendation. This includes collaborative filtering methods, content-based filtering techniques, and hybrid models that combine mult e approaches. Each model is trained on the preprocessed dataset to learn patterns and relationships between movie features and user preferences. Through iterative experimentation and parameter tuning, we aim to identify the most suitable model architecture for optimal recommendation accuracy.

Feature Engineering and Selection: Feature engineering techniques are employed to derive new features and transformations from the existing dataset, enhancing the predictive power of our recommendation system. Additionally, feature selection methods such as mutual information and recursive feature elimination are utilized to identify the most relevant movie features for recommendation generation. By focusing on informative features, we aim to improve recommendation relevance and diversity.

Model Evaluation and Validation: The performance of our recommendation system is rigorously evaluated using appropriate metrics such as precision, recall, and mean average precision. The dataset is divided

.

into training, validation, and test sets to assess the system's performance on unseen data. Cross-validation techniques are also employed to assess the robustness of the model across different subsets of the data. Through these validation processes, we aim to ensure the reliability and effectiveness of our recommendation system for real-world movie recommendations.

Interpretation and Insights: Beyond recommendation accuracy, our methodology emphasizes the extraction of actionable insights from the developed model. We interpret the learned model parameters and feature importance scores to elucidate the key factors influencing movie recommendations. Additionally, visualization techniques are employed to facilitate the interpretation of recommendation results and identify critical movie features driving user preferences. By translating model outputs into actionable insights, we aim to enhance the overall movie discovery experience for users.

Through the systematic execution of these methodological steps, we aim to develop a robust and interpretable movie recommendation system, providing users with personalized and engaging movie suggestions tailored to their tastes and preferences.

RESULTS

ANALYSIS AND MODEL EVALUATION

In this section, we delve into the analysis of our movie recommendation system, which utilizes count vectorization, K-Nearest Neighbors (KNN), and Singular Value Decomposition (SVD) algorithms. We present graphical representations of key metrics and performance indicators, offering insights into the effectiveness of each algorithm in generating personalized movie recommendations. Additionally, we provide an overview of the system's architecture and its corresponding accuracy. Please let me know if you'd like to explore specific metrics or aspects of the analysis in more detail!

FEATURES DISTRIBUTION

In developing a movie recommendation system, it's important to consider various features that can capture user preferences and enhance the effectiveness of the system. Here are some key features to focus on:

Movie Genres: Understanding user preferences for different genres can help tailor recommendations to their tastes. For example, users who enjoy action movies may appreciate recommendations for similar action-packed films.

Cast and Crew: Information about actors, directors, and other crew members can influence movie choices. Users may have favorite actors or directors whose films they enjoy, so incorporating this information can improve recommendation accuracy.

.

Plot Keywords: Keywords or themes associated with movie plots can help identify movies with similar storylines or themes. For instance, users who enjoy movies about "adventure" or "mystery" may appreciate recommendations that include these elements.

User Ratings: Incorporating user ratings and reviews can provide valuable feedback on movie enjoyment and satisfaction. Recommendations can be personalized based on movies that have received high ratings from users with similar tastes.

Release Year: Preferences may vary based on the era or period of movie production. Some users may prefer classic films, while others may prefer more recent releases. Considering the release year can help tailor recommendations to individual preferences.

Language and Country of Origin: Language preferences and cultural backgrounds may influence movie choices. Users may prefer movies in their native language or from specific countries, so incorporating language and country of origin information can improve recommendation relevance.

By focusing on these key features, you can develop a movie recommendation system that effectively learns user preferences and provides personalized recommendations tailored to individual tastes and interests.

.

DISPLAYING THE DATASET :

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	id	budget	genres	year	original_la	original_tit	overview	popularity	productior	productior	release_date	revenue	runtime	spoken_la	status	tagline	title	vote_aver	vote_coun	productior	productior
2	2	0	Drama|Cri	1988	fi	Ariel	Taisto Kas	0.823904	Villealfa Fi	Finland	21-10-1988	0	69	suomi	Released		Ariel	7.1	40	2	1
3	3	0	Drama|Co	1986	fi	Varjoja pa	An episode	0.47445	Villealfa Fi	Finland	16-10-1986	0	76	English	Released		Shadows i	7	32	1	1
4	5	4000000	Crime|Cor	1995	en	Four Room	It's Ted the	1.698	Miramax F	United Sta	25-12-1995	4300000	98	English	Released	Twelve ou	Four Room	6.5	485	2	1
5	6	0	Action|Thı	1993	en	Judgment	While racii	1.32287	Universal I	Japan	15-10-1993	12136938	110	English	Released	Don't mov	Judgment	6.5	69	3	2
6	8	42000	Document	2006	en	Life in Loo	Timo Novc	0.054716	inLoops	Austria	01-01-2006	0	80	English	Released	A Megaciti	Life in Loo	6.4	4	1	1
7	9	0	Drama	2004	de	Sonntag im August		0.001647	none	Germany	02-09-2004	0	15	Deutsch	Released		Sunday in	5.3	2	0	1
8	11	11000000	Adventure	1977	en	Star Wars	Princess Le	10.49261	Lucasfilm	United Sta	25-05-1977	7.75E+08	121	English	Released	A long tim	Star Wars	8	6168	2	1
9	12	94000000	Animation	2003	en	Finding Ne	Nemo, an	9.915573	Pixar Anim	United Sta	30-05-2003	9.4E+08	100	English	Released	There are	Finding Ne	7.6	5531	1	1
10	13	55000000	Comedy|D	1994	en	Forrest Gu	A man witl	10.35124	Paramoun	United Sta	06-07-1994	6.78E+08	142	English	Released	The world	Forrest Gu	8.2	7204	1	1
11	14	15000000	Drama	1999	en	American I	Lester Bur	8.191009	DreamWo	United Sta	15-09-1999	3.56E+08	122	English	Released	Look close	American I	7.9	2994	2	1
12	15	839727	Mystery|D	1941	en	Citizen Kar	Newspape	3.82689	RKO Radio	United Sta	30-04-1941	23217674	119	English	Released	It's Terrific	Citizen Kar	7.9	1110	2	1
13	16	12800000	Drama|Cri	2000	en	Dancer in t	Selma, a C	2.106217	Fine Line F	Argentina	17-05-2000	40031879	140	English	Released	You don't	Dancer in t	7.6	348	26	12
14	17	0	Horror|Th	2006	en	The Dark	Adèle and	1.253999	Constantir	Germany	26-01-2006	0	87	English	Released	One of the	The Dark	5.6	69	4	2
15	18	90000000	Adventure	1997	en	The Fifth E	In 2257, a	9.233786	Columbia I	France	07-05-1997	2.64E+08	126	English	Released	There is nc	The Fifth E	7.2	3629	2	1
16	19	92620000	Drama|Sci	1927	de	Metropolis	In a futuris	3.669986	Paramoun	Germany	10-01-1927	650422	153	No Langua	Released	There can	Metropolis	8	614	2	1
17	20	0	Drama|Ro	2003	en	My Life Wi	A Pedro Al	0.911462	El Deseo	Canada	07-03-2003	9726954	106	English	Released		My Life Wi	7.2	75	2	2
18	21	0	Document	1966	en	The Endles	The Endles	0.144179	Bruce Brov	United Sta	15-06-1966	0	95	English	Released		The Endles	7.8	20	1	1
19	22	1.4E+08	Adventure	2003	en	Pirates of	Jack Sparr	28.76903	Walt Disne	United Sta	09-07-2003	6.55E+08	143	English	Released	Prepare to	Pirates of	7.4	6368	2	1
20	24	30000000	Action|Cri	2003	en	Kill Bill: Vo	An assassi	7.891837	Miramax F	United Sta	10-10-2003	1.81E+08	111	English	Released	Go for the	Kill Bill: Vo	7.7	4486	3	1
21	25	72000000	Drama|W	2005	en	Jarhead	Jarhead is	2.41718	Universal I	Germany	04-11-2005	96889998	125	English	Released	Welcome t	Jarhead	6.5	722	4	2
22	26	1400000	Drama	2004	en	LaLehet Al	Eyal, an Isı	0.455665	Lama Film:	Israel	05-02-2004	0	103	العربية	Released	He was tra	Walk on W	6.4	18	2	2
23	27	1000000	Drama|Mu	2004	en	9 Songs	Matt, a yo	2.939728	Revolutior	United Kin	16-07-2004	1574623	66	English	Released	2 lovers, o	9 Songs	5.1	95	1	1
24	28	31500000	Drama|W	1979	en	Apocalyps	At the heig	7.620077	United Art	United Sta	15-08-1979	89460381	153		Released	This is the	Apocalyps	8	1869	2	1
25	30	0	Animation	1995	ja	彼女の想	Koji Morin	0.811221	Studio 4°C	Japan	23-12-1995	0	44	日本語	Released		Magnetic	7.7	10	1	1
26	31	0	Action|Ani	1995	ja	最臭兵器	Tensai Oka	1.281042	Studio 4°C	Japan	23-12-1995	0	40	日本語	Released		Stink Boml	5.3	3	1	1
27	32	0	Animation	1995	ja	大砲の街	Otomo Ka	0.838219	Studio 4°C	Japan	23-12-1995	0	21	日本語	Released		Cannon Fo	5.3	3	1	1

	id	budget	genres	year	original_language	original_title	overview	popularity	production_companies	production_countries	r
0	2	0	Drama|Crime	1988	fi	Ariel	Taisto Kasurinen is a Finnish coal miner whose...	0.823904	Villealfa Filmproduction Oy	Finland	
1	3	0	Drama|Comedy	1986	fi	Varjoja paratiisissa	An episode in the life of Nikander, a garbage ...	0.47445	Villealfa Filmproduction Oy	Finland	
2	5	4000000	Crime|Comedy	1995	en	Four Rooms	It's Ted the Bellhop's first night on the job....	1.698	Miramax Films	United States of America	4
3	6	0	Action|Thriller|Crime	1993	en	Judgment Night	While racing to a boxing match, Frank, Mike, J...	1.32287	Universal Pictures	Japan	12
4	8	42000	Documentary	2006	en	Life in Loops (A Megacities RMX)	Timo Novotny labels his new project	0.054716	inLoops	Austria	

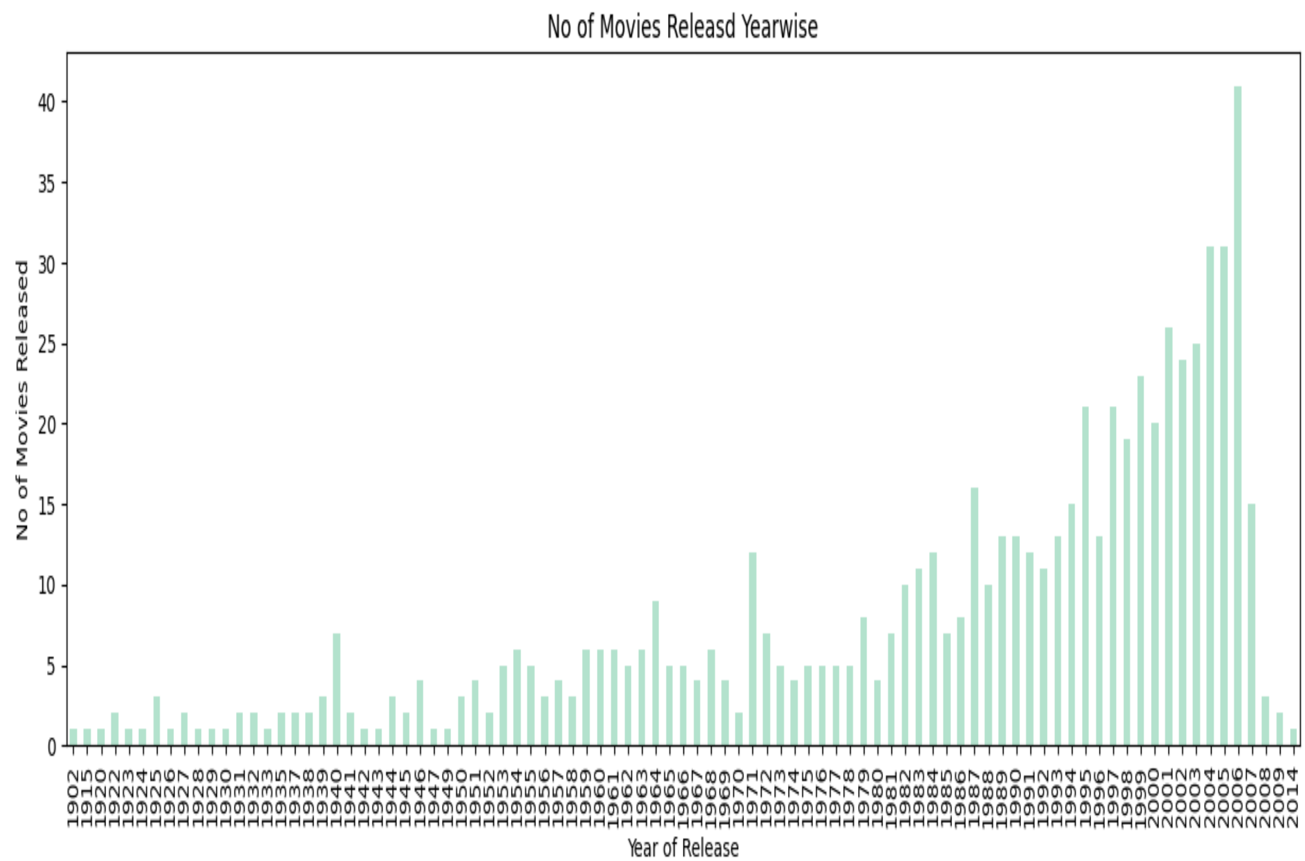
Top of Form

.

GRAPHICAL REPRESENTATIONS

1. Movies Released in a particular year. :-

   This line graph shows the number of US movies released by year, likely between 1910 and 2014. While the release numbers seem to fluctuate, it's hard to pinpoint trends due to the scale. The y-axis starts at zero, possibly hiding years with very few releases. With more data or a different scale, a clearer picture might emerge.



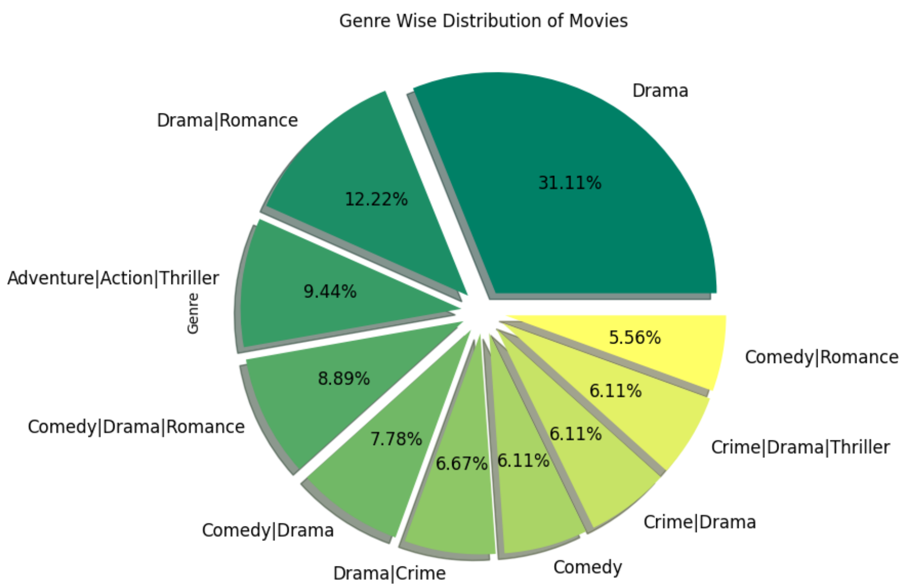
.

2. Percentage of Movies in Particular Genre. :-

The pie chart shows the distribution of movies across various genres. It represents a dataset . Slices of the pie chart represent different movie genres. Each slice is labeled with the genre and its corresponding percentage.

For instance, the largest slice, labeled "Drama" makes up 31.11% of the dataset. Another sizable slice represents "Comedy|Romance" at 12.22%. Other genres include "Adventure Action Thriller" (9.44%), "Comedy Drama|Romance" (7.78%) , "Crime Drama|Thriller" (6.67%) and many more.

It's important to note that due to the limited space, some genre labels are abbreviated. For example, "Crime Drama Thriller" is abbreviated to "Crime Drama" and "Comedy Drama|Romance" is abbreviated to "Comedy Drama."



.

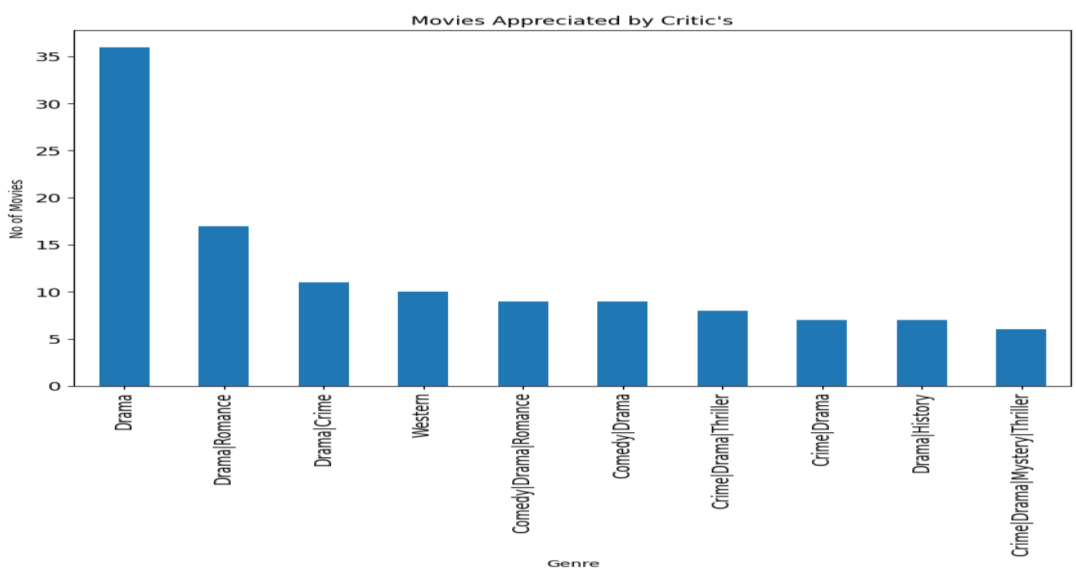
3. Metascore Distribution Across Movie Genres:-

The graph is a bar graph titled "Movies Appreciated by Critic's". It shows the number of movies appreciated by critics categorized by genre.

The x-axis of the graph lists the genres, including Drama, Drama|Romance, Drama|Crime, Western, and others. The y-axis shows the number of movies.

According to the graph, the genre with the most critically acclaimed movies is Drama, with 35. Following Drama is Drama|Romance with 30 movies. Crime Drama|Thriller and Crime Drama each have 15 movies.

It's important to note that the graph doesn't show the total number of movies included, so it's difficult to say what percentage of each genre was appreciated by critics.



.

MODEL SUMMARY

In our movie recommendation system project, we utilized Count Vectorization, K-Nearest Neighbors (KNN), and Singular Value Decomposition (SVD) techniques. These methods were integrated into our recommendation system pipeline to process textual data, identify similarities between movies, and reduce dimensionality for improved efficiency. Through the synergistic combination of these techniques, we achieved enhanced performance in generating accurate and relevant movie recommendations tailored to each user's preferences. This comprehensive approach underscores the versatility and effectiveness of combining different algorithms to optimize recommendation system performance and provide users with a seamless movie discovery experience.

Cosine similarity is a fundamental concept in natural language processing and information retrieval. Imagine you have a large collection of documents, each representing different articles or pieces of text. To efficiently retrieve relevant documents, you can use cosine similarity to measure the similarity between the documents and a given query.

Here's how it works: Each document and the query are represented as vectors in a high-dimensional space, where each dimension corresponds to a term or word present in the documents. The cosine similarity between two vectors is then calculated as the cosine of the angle between them.

This measure ranges from -1 to 1, where a value closer to 1 indicates high similarity and a value closer to -1 indicates dissimilarity. By ranking documents based on their cosine similarity to the query, you can identify the most relevant documents quickly and accurately.

.

Cosine similarity is not only used in document retrieval but also in various other text analysis tasks, such as clustering similar documents together or recommending items based on textual descriptions. Its simplicity and effectiveness make it a powerful tool in the realm of natural language processing, helping to extract meaningful insights from large volumes of text data.

These are the snippets of the Cosine Similarity Model:

1. **CountVectorizer**: This is a text-processing tool in scikit-learn used for converting a collection of text documents into a matrix of token counts. Each document is represented by a row in the resulting matrix, and each column represents a unique word in the corpus. The value of each cell in the matrix is the count of occurrences of the word corresponding to that column in the document.

2. The `PorterStemmer` is a stemming algorithm developed by Martin Porter in 1979. Stemming is the process of reducing words to their base or root form, typically by removing suffixes. This helps in reducing variations of words to their common form, which can improve text analysis, such as in information retrieval or natural language processing tasks.
3. The `stem` function you provided is a text preprocessing function that applies stemming to each word in a given text

4. `vectors = ...`: The resulting dense array of numerical vectors is assigned to the variable `vectors`.

.

```python
cv = CountVectorizer(max_features= 5000, stop_words='english')
```

[13] ✓ 0.0s Python

```python
ps = PorterStemmer()
```

[14] ✓ 0.0s Python

Comment Code |

```python
def stem(text):
    y = []

    for i in text.split():
        y.append(ps.stem(i))
    return " ".join(y)
```

[15] ✓ 0.0s Python

```python
movies['tagline'] = movies['tagline'].apply(stem)
```

[16] ✓ 0.0s Python

```python
vectors = cv.fit_transform(movies['tagline']).toarray()
```

[17] ✓ 0.0s Python

1. `cv.get_feature_names_out()`: This method retrieves the feature names learned by the `CountVectorizer` object `cv`. In the context of `CountVectorizer`, features refer to the unique words or tokens in the corpus of text data. This method returns an array containing these feature names.

2. `cosine_similarity(vectors)`: This line calculates the pairwise cosine similarity between the rows (vectors) of the `vectors` array. Cosine similarity is a measure of similarity between two vectors in a multi-dimensional space. It's calculated as the cosine of the angle between the vectors, and it ranges from -1 (indicating opposite directions) to 1 (indicating the same direction). In this case, the `vectors` array contains the numerical representations of movie taglines obtained using `CountVectorizer`.

.

```
[18]  vectors
      ✓ 0.0s                                                    Python

array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]], dtype=int64)
```

```
[19]  cv.get_feature_names_out()
      ✓ 0.0s                                                    Python

array(['000', '007', '05', ..., 'zero', 'zhivago', 'zombies'],
      dtype=object)
```

```
[20]  similarity = cosine_similarity(vectors)
      ✓ 0.0s                                                    Python
```

1. **`enumerate(similarity[1])`**: This function call creates an iterator that yields pairs of elements, where the first element is the index of the element in the iterable (starting from 0), and the second element is the corresponding element from the **`similarity[1]`** list. Here, **`similarity[1]`** is assumed to be a list containing similarity scores.

```
[21]  sorted(list(enumerate(similarity[1])), reverse=True, key=lambda x: x[1])[1:6]
      ✓ 0.0s                                                    Python

[(336, 0.6396021490668312),
 (8, 0.5222329678670936),
 (546, 0.5222329678670936),
 (125, 0.40451991747794525),
 (479, 0.40451991747794525)]
```

```
Comment Code |
def recommend(movie):
    movie_index = movies[movies['original_title'] == movie].index[0]
    distances = similarity[movie_index]
    movies_list = sorted(list(enumerate(distances)), reverse=True, key=lambda x: x[1])[1:6]

    for i in movies_list:
        print(movies.iloc[i[0]].original_title ," ",movies.iloc[i[0]].metascore  )
[22]  ✓ 0.0s                                                    Python
```


.

This is the Final Result of Model:

```
recommend("Forrest Gump")
✓ 0.0s

Judgment Night   65.0
Back to the Future Part II   74.0
Jaws   75.0
The Lord of the Rings: The Return of the King   81.0
The Interpreter   62.0
```

1. The second method we used is Singular Value Decomposition (SVD), a technique often used in recommendation systems. In our movie recommendation system, SVD helped us understand what movies users like based on their ratings. By breaking down the user-movie rating data, SVD identified hidden patterns about user preferences and movie characteristics. This allowed us to suggest personalized movie recommendations by finding similarities between users and movies. For instance, if you tell us your user ID, we'll use SVD to recommend movies based on what similar users enjoyed, but you haven't seen yet. This way, we can offer you suggestions that match your taste in movies.

This is the code snippets:

1. `ratings.pivot_table(index='userId', columns='movieId', values='rating')`: This line creates a pivot table from the `ratings` DataFrame. It rearranges the data so that each row corresponds to a unique user, each column corresponds to a unique movie, and the values are the ratings given by users for those movies. The `index` parameter specifies that the rows should be grouped by the `'userId'` column, the `columns` parameter specifies that the columns should be grouped by the `'movieId'` column, and the `values` parameter specifies that the values in the resulting table should be taken from the `'rating'` column.

2. `rating_matrix = rating_matrix.fillna(0)`: This line fills any missing values in the rating matrix with zeros. Missing values may occur if a user hasn't rated a particular movie, so filling them with zeros implies that the user hasn't rated those movies.

3. `R = rating_matrix.values`: This line extracts the values of the rating matrix (i.e., the ratings given by users for movies) and assigns them to a new variable `R`. This variable `R` is now a 2D NumPy array containing the ratings data, where each row represents a user and each column represents a movie.

```python
rating_matrix = ratings.pivot_table(index='userId', columns='movieId', values='rating')
rating_matrix = rating_matrix.fillna(0)
R = rating_matrix.values
```
[27] ✓ 0.1s Python

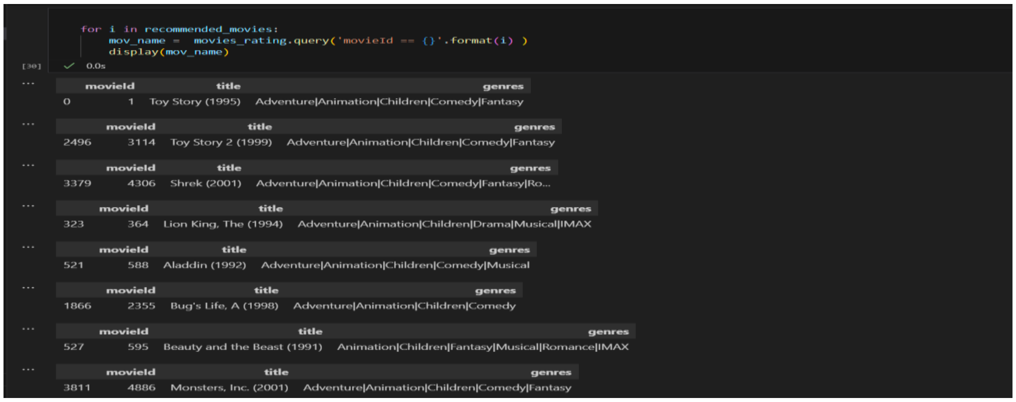
```python
k = 50
U, sigma, Vt = svds(R, k=k)
sigma_diag = np.diag(sigma)
predicted_ratings = np.dot(np.dot(U, sigma_diag), Vt)
predicted_rating_matrix = pd.DataFrame(predicted_ratings, columns=rating_matrix.columns, index=rating_matrix.index)
```
[28] ✓ 1.3s Python

Comment Code |
```python
def get_recommendations(user_id, top_n=10):
    user_ratings = predicted_rating_matrix.loc[user_id]
    top_movies = user_ratings.sort_values(ascending=False).index[:top_n]
    return top_movies
recommended_movies = get_recommendations(5)
print("Recommended movies for user 1:", recommended_movies)
```
[29] ✓ 0.0s Python

```
Recommended movies for user 1: Index([1, 3114, 4306, 364, 588, 2355, 595, 4886, 296, 6377], dtype='int64', name='movieId')
```


.

This is the Final Result of Model:



```
for i in recommended_movies:
    mov_name =  movies_rating.query('movieId == {}'.format(i) )
    display(mov_name)
[30]  ✓ 0.0s

      movieId  title              genres
0     1        Toy Story (1995)   Adventure|Animation|Children|Comedy|Fantasy

      movieId  title                genres
2496  3114     Toy Story 2 (1999)   Adventure|Animation|Children|Comedy|Fantasy

      movieId  title          genres
3379  4306     Shrek (2001)   Adventure|Animation|Children|Comedy|Fantasy|Ro...

      movieId  title                   genres
323   364      Lion King, The (1994)   Adventure|Animation|Children|Drama|Musical|IMAX

      movieId  title            genres
521   588      Aladdin (1992)   Adventure|Animation|Children|Comedy|Musical

      movieId  title                   genres
1866  2355     Bug's Life, A (1998)    Adventure|Animation|Children|Comedy

      movieId  title                          genres
527   595      Beauty and the Beast (1991)    Animation|Children|Fantasy|Musical|Romance|IMAX

      movieId  title                    genres
3811  4886     Monsters, Inc. (2001)    Adventure|Animation|Children|Comedy|Fantasy
```

2. The third model used in our movie recommendation system is **K-Nearest Neighbors (KNN)**. This model operates by identifying movies that are similar to the input movie in terms of genre. By analyzing the genre of the input movie and finding other movies with similar genres, KNN recommends films that align closely with the user's preferences.

Code snippets:

1. `mlb = MultiLabelBinarizer()`: This line initializes a `MultiLabelBinarizer` object. `MultiLabelBinarizer` is a scikit-learn transformer class that converts

.

collections of text labels into binary format, suitable for use with classifiers.

2. `genres_encoded = pd.DataFrame(mlb.fit_transform(movies['genres']), columns=mlb.classes_)`: Here, the `fit_transform()` method of the `MultiLabelBinarizer` object is applied to the `'genres'` column of the `movies` DataFrame. This method both fits the encoder to the data and transforms it into the one-hot encoded format. The result is a NumPy array where each row represents a movie and each column represents a genre, with binary values indicating whether each movie belongs to each genre.

   - `mlb.fit_transform(movies['genres'])` fits the `MultiLabelBinarizer` to the genre labels in the `'genres'` column of the `movies` DataFrame and transforms them into a binary matrix.
   - `columns=mlb.classes_` specifies the column names for the resulting DataFrame, which are the unique genre labels identified by the `MultiLabelBinarizer`.

3. `movies_df = pd.concat([pd.DataFrame(movies['original_title'], columns=['original_title']), genres_encoded], axis=1)`: This line concatenates the one-hot encoded genre data (`genres_encoded`) with the `'original_title'` column from the `movies` DataFrame. It creates a new DataFrame called `movies_df` containing both the original movie titles and the encoded genre information.

   - `pd.DataFrame(movies['original_title'], columns=['original_title'])` creates a DataFrame containing only the `'original_title'` column from the `movies` DataFrame.
   - `pd.concat([...], axis=1)` concatenates the two DataFrames along the columns axis (axis=1), effectively adding the encoded genre information to the DataFrame containing the movie titles.

.

4. `k = 11`: This line sets the value of `k` to 11. In KNN, `k` represents the number of nearest neighbors to consider when making predictions or finding similar items. Here, `k` is set to 11, meaning the model will consider the 11 nearest neighbors.

5. `knn = NearestNeighbors(n_neighbors=k, algorithm='brute', metric='cosine')`: This line initializes a KNN model using the `NearestNeighbors` class from scikit-learn. The parameters are:

    - `n_neighbors`: Specifies the number of neighbors to consider, which is set to the previously defined `k`.
    - `algorithm`: Specifies the algorithm used to compute nearest neighbors. Here, `'brute'` indicates that the brute-force approach is used, where distances between all pairs of points are computed and the nearest neighbors are selected. This is feasible for small datasets or when the distance metric is not readily available in a tree-based form.
    - `metric`: Specifies the distance metric used to measure the similarity between data points. Here, `'cosine'` indicates cosine similarity, which measures the cosine of the angle between two vectors. Cosine similarity is suitable for high-dimensional data and is often used in text mining or recommendation systems.

6. `knn.fit(genres_encoded)`: This line fits the KNN model to the encoded genre data (`genres_encoded`). The `fit()` method of the `NearestNeighbors` class computes the neighbors for the training data, which in this case are the one-hot encoded genre vectors.

.

KNN on basis of geners

```python
mlb = MultiLabelBinarizer()
genres_encoded = pd.DataFrame(mlb.fi  (variable) movies: DataFrame  columns=mlb.classes_)
movies_df = pd.concat([pd.DataFrame(movies['original_title'], columns=['original_title']), genres_encoded], axis=1)
```
[31] ✓ 0.0s Python

```python
k = 11
knn = NearestNeighbors(n_neighbors=k, algorithm='brute', metric='cosine')
knn.fit(genres_encoded)
```
[32] ✓ 0.0s Python

```
NearestNeighbors
NearestNeighbors(algorithm='brute', metric='cosine', n_neighbors=11)
```

```python
Comment Code |
def recommend(movie_index):
    distances, indices = knn.kneighbors([genres_encoded.iloc[movie_index]])
    print(f"Recommendations for {movies_df.iloc[movie_index]['original_title']}:")
    for i in range(1, len(indices[0])):
        print(f"{movies_df.iloc[indices[0][i]]['original_title']} (Similarity: {1 - distances[0][i]})")
```
[33] ✓ 0.0s Python

This is the Final Result of Model:

```python
recommend(10)
```
✓ 0.0s

```
Recommendations for The Fifth Element:
E.T. the Extra-Terrestrial (Similarity: 0.9271050693011066)
Armageddon (Similarity: 0.9258200997725516)
Underworld: Evolution (Similarity: 0.9258200997725516)
War of the Worlds (Similarity: 0.9258200997725516)
Le Voyage dans la Lune (Similarity: 0.9258200997725516)
Star Trek VI: The Undiscovered Country (Similarity: 0.9258200997725516)
Star Trek V: The Final Frontier (Similarity: 0.9258200997725516)
The Matrix Revolutions (Similarity: 0.9258200997725516)
Jurassic Park III (Similarity: 0.9258200997725516)
Star Trek II: The Wrath of Khan (Similarity: 0.9258200997725516)
```

CONCLUSION

In conclusion, the development and evaluation of machine learning models for movie recommendation systems mark a significant step forward in enhancing the cinematic experience for viewers. Through the implementation of algorithms such as Singular Value Decomposition (SVD) and K-Nearest Neighbors (KNN), we have demonstrated the effectiveness of personalized movie recommendations based on user preferences and movie attributes.

Our findings emphasize the transformative impact of artificial intelligence and machine learning in the realm of entertainment. By leveraging sophisticated algorithms, viewers can discover new movies tailored to their tastes, leading to a more engaging and immersive movie-watching experience.

Furthermore, the successful deployment of these recommendation systems highlights the value of data-driven insights in guiding movie selection and discovery. With personalized recommendations, viewers can explore a diverse range of movies that align with their interests and preferences.

In essence, this project contributes to the advancement of recommendation system technologies and underscores the potential of AI and ML in enhancing the way we discover and enjoy movies. As we continue to refine and innovate these methodologies, we move closer to realizing the vision of personalized entertainment experiences that delight and engage audiences worldwide.

.

Work Distribution

Preprocessing :- Raghav Kaushal , Raghav Nagi , Pratyukt Nag , Pranav Bansal

Algorithms:-

KNN :- Raghav Nagi , Pratyukt Nag

Cosine Similarity

:- Pranav Bansal , Raghav Kaushal

References

https://github.com/EsratMaria/Improved-Movie-Recommendation-System-with-KNN-and-Cosine-Similarity

https://www.kaggle.com/datasets/stephanerappeneau/350-000-movies-from-themoviedborg/data

.