Prepared by: Jayesh Zambre, Soumya Mohan, Natalie Greener, Ryan Schmitzer, Agustin Arcos

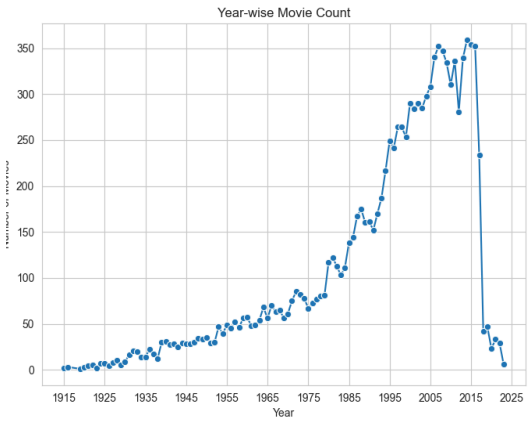
Movie Recommender Systems

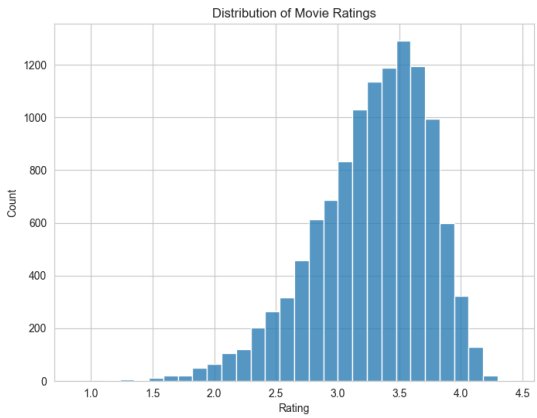
Introduction

In the rapidly evolving landscape of digital entertainment, our movie recommendation system utilizes Python machine learning techniques and integrates data from MovieLens and IMDb to create personalized, content-specific recommendations. By analyzing vast datasets including user ratings, movie attributes, and viewing patterns, our system offers tailored suggestions addressing user preferences and film characteristics. The project aims to develop diverse recommender systems, including Content-based, Collaborative, and Knowledge-based models, addressing challenges such as the cold start problem and data sparsity. Strategies encompass KNN-based recommendations, SVD-based collaborative filtering, and a Softmax Deep Neural Network for varied recommendation approaches. Additionally, integrating scraped IMDB data aids in comprehending user behavior. Movie recommender systems are essential for streaming platforms like Netflix and Disney Plus and can enhance their user experience, facilitate content discovery, and can boost viewer engagement through collaborative and content-based filtering methods and can potentially also increase user retention and revenue through targeted cross-selling.

Exploratory Data Analysis

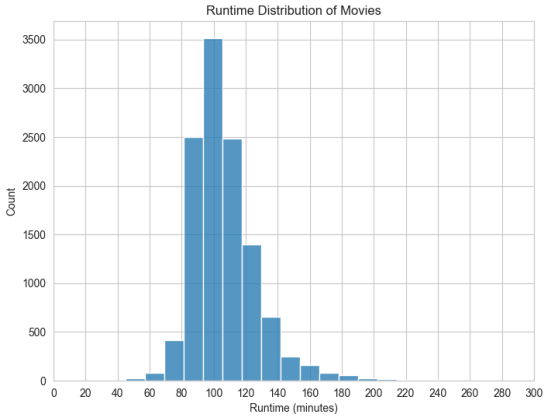
MovieLens provides a vast collection of approximately 33 million ratings, contributed by 330,975 users and covering 86,000 movies. It also features a comprehensive tag genome, which assigns 14 million relevance scores across 1,100 tags for each film. The tags themselves provide brief snippets of content-related information regarding films, so the tag genome enriches this data by assigning the relevance of any particular tag to any film. Meanwhile, the IMDb dataset encompasses around 10 million observations of video media, complemented by rich content-based data. Additionally, IMDb documents the professional contributions of approximately 13 million individuals involved in video

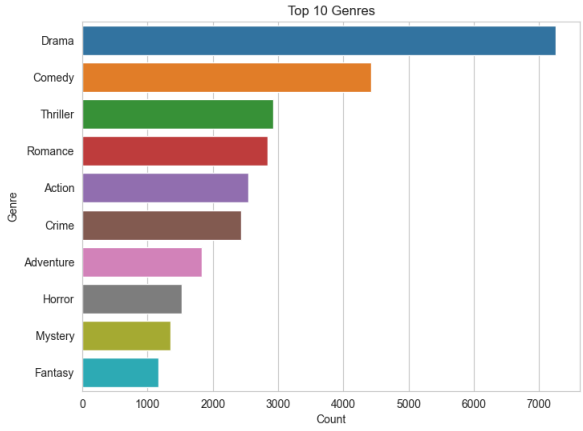




media production, along with the titles they are associated with. The provided exploratory visualizations offer several insights into general trends in the datasets.

The "Year-wise Movie Count" graph depicts an exponential increase in movie production over time with a sharp recent drop, possibly due to the impact of digital streaming platforms on movie production or external factors such as recent global events affecting the film industry. The graph, "Average Rating by Year," reveals a gradual decline in average movie ratings over the last century, with notable variability and occasional peaks suggesting periods of critically acclaimed releases. The "Distribution of Movie Ratings" histogram suggests that most movies receive moderate to high ratings, with a clear skew towards the 3-4 rating range.



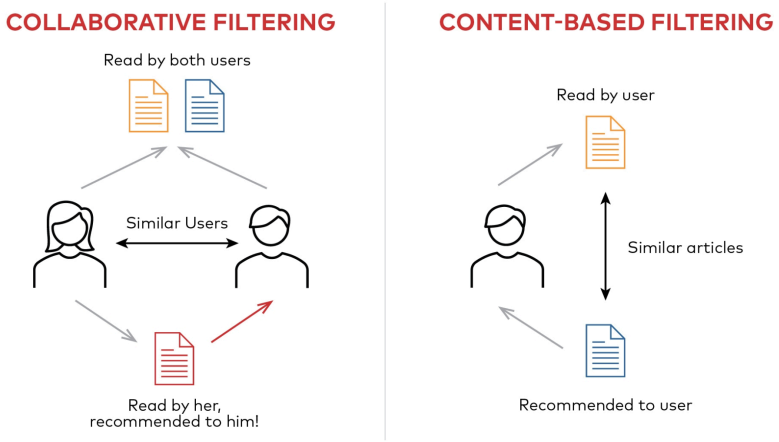


The "Runtime Distribution of Movies" histogram shows a strong concentration of movies around the 90 to 120-minute mark, implying a standardized preference in movie lengths that aligns with traditional cinematic formats and audience attention spans. The "Top 10 Genres" bar chart demonstrates a clear preference for Drama, followed by Comedy and Thriller, indicating these genres are most prevalent or popular among releases.

Content-based filtering

Content-based filtering is a personalized recommendation system that analyzes the intrinsic characteristics and attributes of items (in this case movies) to provide tailored suggestions to users. At its core, content-based filtering aims to understand the unique tastes and interests of individual users and recommend items that align with their preferences. For example, consider a movie recommendation system that utilizes content-based filtering. By analyzing content attributes such as genre, actors, directors, and plot keywords, the system can create profiles for movies. When a user expresses interest in a particular movie, the system can recommend other movies with similar content traits, such as the

same genre or featuring the same actors. By considering the content features that each user finds appealing, content-based filtering can tailor recommendations to their unique tastes and preferences. This personalized approach enhances user satisfaction and engagement, leading to a more enjoyable user experience.



Key Concepts and Methodology

Content-based recommender systems leverage several key concepts to generate personalized recommendations for users based on the intrinsic characteristics of items and the preferences of users. Some of the fundamental concepts used in content-based recommender systems include:

- **Item Profiles:** Item profiles encapsulate the essential qualities or features of each item, such as genre, directors, relevant tags, ratings, and other attributes. These features are used to distinguish between items and characterize their content, enabling the recommender system to make personalized recommendations.
- **Term Frequency (TF) and Inverse Document Frequency (IDF):** TF measures the frequency of a word or term in a document, while IDF measures the importance of a term in the entire corpus of documents. TF-IDF weighting is used to determine the relative importance of terms in documents, considering both their frequency in the document and their rarity across the corpus. TF-IDF helps to mitigate the influence of common words and highlight the importance of rare or unique terms in characterizing the content of items.
- **Vector Space Model:** The Vector Space Model represents items and users as vectors in a multi-dimensional space, where each dimension corresponds to a feature or attribute. Items are represented by vectors that capture their intrinsic characteristics, such as genre, actors, directors, and other descriptive elements. User profiles are created based on their interactions with items, and similarity between users and items is computed based on the angle between their vectors.

Cosine similarity is commonly used to measure the similarity between vectors, with smaller angles indicating greater similarity.

- **Cosine Distance:** Cosine distance is used to quantify the similarity between user preferences and item attributes. It measures the cosine of the angle between user and item vectors, with smaller angles indicating greater similarity. Cosine distance helps identify items that align closely with user preferences based on their intrinsic characteristics and content traits.

Recommendation Result

The recommendations provided by the content-based recommender for "Doctor Strange" encompass a mix of superhero and fantasy movies, reflecting similar themes and genres. Films such as "Thor: The Dark World" and "X2: X-Men United" closely align with the superhero narrative and action-packed sequences characteristic of "Doctor Strange." Additionally, the inclusion of "Doctor Strange in the Multiverse of Madness," the sequel to the original film, suggests continuity and relevance in the recommendations, catering to fans of the franchise who seek further exploration of the mystical and adventurous world depicted in the series.

```
Recommendations for Doctor Strange:
16922                           Thor: The Dark World
5534                                X2: X-Men United
38368    Doctor Strange in the Multiverse of Madness
16349                         Star Trek Into Darkness
11000                                        Watchmen
27941                       Kubo and the Two Strings
28849                 Rogue One: A Star Wars Story
15720                  Conscientious Objector, The
37166                                   The Courier
19768                                       Ant-Man
Name: title_name, dtype: object
```

Benefits of Content-Based Filtering

- **Independent of Other User Data**: Content-based filtering makes personalized recommendations based solely on a user's own activity, independent of other users' data.
- **Tailored to User's Preferences**: Content-based filtering aligns recommendations with the user's interests by matching database object attributes with the user's profile, ensuring highly personalized suggestions.
- **Transparency in Recommendations**: Content-based filtering provides transparent recommendations directly tied to the user's actions, fostering user trust compared to collaborative filtering.

- **Overcoming the "Cold Start" Problem**: Content-based filtering efficiently delivers quality recommendations with minimal user data, addressing the "cold start" problem often encountered in collaborative filtering.

Limitations of Content-Based Filtering

- **Lack of Diversity:** Content-based recommenders tend to suggest items that closely resemble those the user has previously interacted with, leading to a narrow selection of options. For instance, if a user enjoys action movies, the recommender may primarily recommend similar action-packed films, limiting exposure to other genres. This restricts users to their existing preferences, hindering exploration of different content.
- **Scalability & Consistency Issues:** As the number of items in the system grows, defining and tagging attributes for each new item becomes increasingly time-consuming and challenging. Additionally, subjective attribute assignment can lead to inconsistencies among individuals, affecting recommendation accuracy. This scalability issue and lack of consistency may impact the effectiveness of the recommender system.

User-based collaborative filtering

User-based collaborative filtering is a recommendation approach that leverages the collective behavior of users to make personalized recommendations. Instead of analyzing the intrinsic characteristics of items, user-based collaborative filtering focuses on similarities between users based on their interactions with items. The underlying assumption is that users who have interacted similarly with items in the past are likely to have similar preferences and interests in the future.

Key Concepts and Methodology:

- **User-Item Interaction Matrix**: The user-item interaction matrix represents the historical interactions between users and items, in the form of ratings or preferences. Each row corresponds to a user, each column corresponds to an item, and the entries represent the user's rating or preference for that item.
- **User Similarity Calculation**: User similarity is calculated based on the similarity of their interaction patterns. Similarity metrics, such as cosine similarity or Pearson correlation (strength and direction of linear relationship), can be used to quantify the resemblance between users.
- **Neighborhood Selection**: Once user similarity is calculated, a neighborhood of similar users is selected for each target user. This neighborhood comprises users whose interaction patterns closely resemble that of the target user.

- **Rating Prediction**: To generate recommendations for a target user, the ratings of items by similar users are aggregated and weighted based on their similarity to the target user. The weighted average of these ratings is used to predict the target user's ratings for unrated items.
- **Prediction of Movies**: After calculating the similarity between users and selecting a neighborhood of similar users, the next step in user-based collaborative filtering is to predict the ratings of unrated items for the target user. This process involves aggregating the ratings of similar users for each unrated item and computing a weighted average.

Recommendation Generation: Once the ratings of unrated items are predicted for the target user, the system generates recommendations by selecting the top-rated items. These recommendations represent the movies that are predicted to be most suitable for the user based on the behavior of similar users.

```
Top 10 movie recommendations for User 4:
Parasite
Joker
Apollo 11
Klaus
Roger Waters: Us + Them
Piranhas
Motherless Brooklyn
Article 15
Gully Boy
Hit-and-Run Squad


Movies already watched by User 4:
Avengers: Infinity War - Part II
John Wick: Chapter 3 – Parabellum
Pokémon: Detective Pikachu
Ford v. Ferrari
Fast & Furious Presents: Hobbs & Shaw
```

The predicted model demonstrates an understanding of the user's varied interests, extending recommendations beyond the initially observed genres of Action, Animation, Adventure, and Thriller to recommend movies spanning multiple themes including Drama, Crime, and Music. This suggests that the model recognizes the user's potential interest in a wide range of themes and aims to cater to these diverse preferences and provide a well-rounded selection that resonates with the user's eclectic tastes.

Benefits of User-based collaborative filtering:

- **Personalized Recommendations**: Personalized recommendations tailored to the preferences of individual users, based on the behavior of similar users.

Limitations of User-based collaborative filtering:

- **Data Sparsity**: This happens in scenarios where users have rated only a small fraction of the available items which can lead to inaccuracies in similarity calculations recommendations.
- **Cold Start Problem**: The system struggles with the cold start problem, where new users or items lack sufficient interaction data for accurate recommendations.
- **Scalability**: As the number of users and items increases, the computational complexity of user-based collaborative filtering also grows, making it less scalable for large datasets.

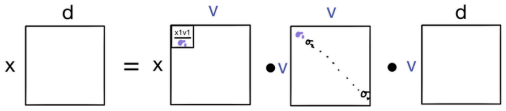
Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) is a key collaborative filtering method for personalized movie recommendations. SVD excels in predicting how users might rate unseen movies, which is crucial for customizing content suggestions to individual user preferences. This method adeptly addresses the complexities of user-item interactions, laying the foundation for a more sophisticated content recommendation experience.

Key Concepts and Methodology

SVD achieves this by decomposing the user-item rating matrix into three matrices, revealing latent features, which are the intrinsic yet hidden statistical patterns in user ratings. These features, while not directly observable, encapsulate user preferences across various dimensions. The decomposition yields:

- User Matrix (U): Associates users with latent features, with rows representing individual users and columns representing the abstract dimensions.
- Singular Value (Σ, Sigma) Matrix: Contains the singular values in a diagonal layout, signifying the importance of each latent feature.
- Item Matrix (V^T): Establishes the relationship between movies and latent features, where each column is linked to a movie and rows to latent features.

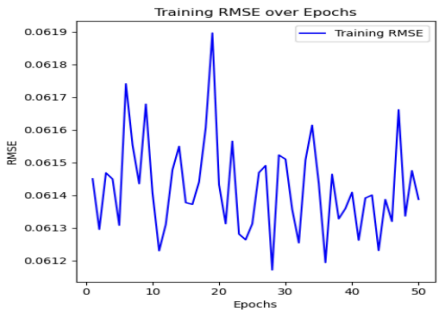


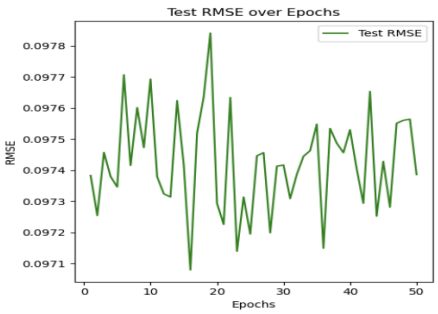
$$A = U \cdot \Sigma \cdot V^T$$

The SVD algorithm, optimized through a 3 fold grid search over multiple parameters such as latent factors, epochs, learning rate, and regularization, demonstrated its prediction capability over numerous training epochs.

Prepared by: Jayesh Zambre, Soumya Mohan, Natalie Greener, Ryan Schmitzer, Agustin Arcos

Recommendation Results






```
Training RMSE: 0.12
Test RMSE: 0.12
Training MAE: 0.05
Test MAE: 0.06
```

The model's performance, as evaluated by RMSE and MAE metrics indicates a high level of accuracy for both the training and test sets. Additionally, these graphs of the training and test RMSE's over 50 epochs, demonstrates the model's strong ability to generalize well without clear signs of overfitting. The model's efficacy is evident through specific examples: it predicted User ID 136701's rating for "Toy Story" as 3.88, aligning closely with the actual rating of 3.89. Likewise, it estimated User ID 155146's rating for "Mute" at 2.82, nearly matching the actual rating of 2.77. These precise predictions showcase the model's capability in offering streaming services an effective tool for content curation tailored to viewer preferences.

```
User ID: 136701
Movie Name: Toy Story
Predicted Rating Score: 3.88
Actual Rating: 3.89
```

```
User ID: 155146
Movie Name: Mute
Predicted Rating Score: 2.82
Actual Rating: 2.77
```

Benefits of Singular Value Decomposition

- **Enhanced Content Curation:** SVD significantly improves content curation, enabling streaming services to deliver highly personalized recommendations that cater to individual viewer preferences.
- **Effective in Data Sparsity:** SVD effectively handles scenarios with sparse data, common in large-scale streaming services. By identifying underlying patterns in limited user interactions, it maintains the accuracy of recommendations despite minimal data.

Limitations of Singular Value Decomposition

- **Computational Complexity:** The approach may involve considerable complexity and computational demand, especially when processing extensive datasets.
- **Nuance Recognition Challenges:** There's a risk of oversimplification, as SVD might not always capture the subtle and varied nuances of user preferences fully.

Deep Neural Network

DNN is an advanced form of artificial neural network that consists of multiple layers between the input and output layers, known as hidden layers. These networks are designed to model complex patterns and relationships in data by passing information through these layers, where each layer progressively extracts higher-level features from the input data. DNNs leverage non-linear activation functions to allow for the learning of non-linear mappings between inputs and outputs, making them highly effective for a wide range of tasks such as image and speech recognition, natural language processing, and predictive analytics. The depth of these networks, signified by the number of hidden layers, enables them to learn intricate patterns in large datasets, but also requires sophisticated techniques to train effectively, including backpropagation and regularization methods to prevent overfitting.

Regression Approach

- We tested many activation functions, including Relu, LeakyRelu, Tanh, Softmax, and Softplus. The two best-performing activation functions were Relu and Softplus.
- This DNN features three hidden layers with 128, 64, and 32 neurons, respectively, each using 'Relu' (Rectified Linear Unit) as the activation function to introduce non-linearity. We used MAE, MSE and RMSE to find the best fit model.
- We used Adam optimizer and an MSE loss function for training, combined with the activation functions in the hidden layers to minimize prediction errors by adjusting weights to better capture the underlying data patterns.

Performance Metrics for Relu

Metric	Test Accuracy
MSE	0.886
RMSE	0.929
MAE	0.707

- These metrics indicate the model's average deviation from the true values in terms of both absolute and squared terms, with the RMSE being slightly higher than the MSE due to its nature of penalizing larger errors more heavily.
- The performance metrics suggest the model has achieved a reasonable level of accuracy in predicting continuous values (movie ratings), although further optimization may be needed depending on the specific application's accuracy requirements.

Rating Prediction for Relu

- Looking at the example below we can see that this model's movie ratings prediction for "Toy Story" was 8.567, which is 0.267 away from the actual value of 8.3. Overall, this model predicts movie ratings very accurately.

Regression using SoftPlus

- The second DNN model features the same three hidden layers with 128, 64, and 32 neurons, respectively, each using 'softplus' as the activation function to introduce non-linearity. The choice of 'softplus,' a smooth approximation to the 'relu' function, suggests an attempt to mitigate issues like the dying 'relu' problem while maintaining the ability to model complex, non-linear relationships in the data. The "dying ReLU" problem refers to the issue where ReLU units can become inactive, causing them to output zero for all inputs, often due to large negative inputs, leading to dead neurons that do not contribute to the network's learning. We are also using MSE, MAE and RMSE for metrics to evaluate this model.

Performance Metrics for Softplus

Metric	Test Accuracy
MSE	0.883
RMSE	0.930
MAE	0.715

The model utilizing "softplus" activation has shown promising results and is reasonably accurate in predicting continuous values (movie ratings), with the RMSE and MSE providing insights into the average magnitude of the model's prediction errors. The use of "softplus" activation, combined with the adam optimizer and mean squared error loss function, has contributed to the model's ability to capture the non-linear relationships within the data effectively.

Rating Prediction for SoftPlus

We used the DNN with 'softplus' to predict the movie rating for "Toy Story" and predict a score of 8.3225, which is off the true value by 0.0225.

Overall, both models performed well when trying to predict the score for movie ratings and have similar performance metrics scores (MAE, MSE, RMSE). The model that utilized 'softplus' activation performed slightly better on predicting "Toy Story" movie ratings, but this is just one sample. In the end, DNN was able to perform well and predict movie ratings accurately.

Benefits of Deep Neural Network

- **Pattern Recognition:** Deep neural networks excel at learning intricate patterns in user preferences and movie features, leading to more accurate recommendations.
- **Scalability:** They efficiently process vast amounts of user interaction data and movie attributes, making them suitable for large-scale recommendation systems.
- **Feature Learning:** Deep neural networks automatically learn relevant features from raw movie data, eliminating the need for manual feature engineering.
- **Personalization:** They leverage individual user behavior and historical movie ratings to generate highly personalized recommendations.
- **Flexibility:** Deep neural network architectures can be adapted to various recommendation tasks, including collaborative filtering, content-based filtering, and hybrid approaches.

Limitations of Deep Neural Network

- **Data Dependency**: They require large amounts of labeled training data, posing challenges in acquiring labeled data for niche genres or new releases.
- **Complexity**: Deep neural networks are inherently complex, making it difficult to interpret their decision-making process.
- **Overfitting**: They are prone to overfitting, especially when trained on noisy or sparse movie data, leading to poor generalization performance.
- **Training Complexity**: Training deep neural networks for movie recommendation can be computationally intensive and time-consuming.
- **Cold Start Problem**: Deep neural networks may struggle with the cold start problem, particularly for new users or movies with limited interaction data.

Conclusion

Our recommendation system employs a diverse array of techniques to provide personalized and accurate recommendations. The system gains valuable insights into individual user behavior, allowing for tailored recommendations that closely align with their tastes. Thus a comprehensive approach can enhance the overall recommendation strategy, leveraging both explicit feedback and implicit interactions to capture a wide range of user preferences. As a result, users can receive more relevant and engaging recommendations, leading to higher satisfaction, increased engagement, and improved retention rates. Ultimately, our system cultivates user loyalty through a personalized and enriching recommendation experience.