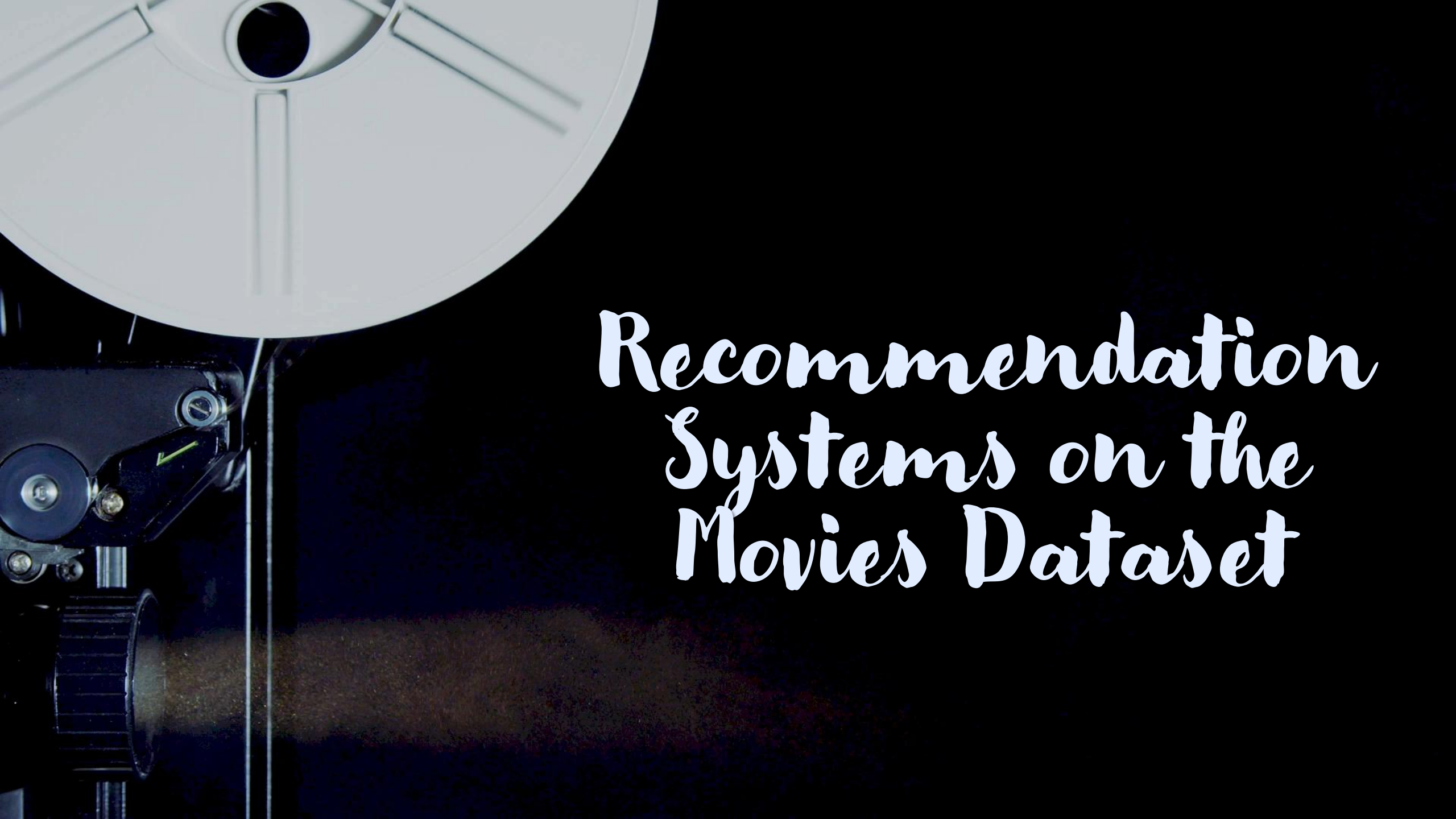


Recommendation Systems on the Movies Dataset

CRISP-DM

- Stage I - Business understanding
- Stage II - Data Understanding
- Stage III - Data Preparation
- Stage IV - Modeling
- Stage V - Evaluation
- Stage VI - Deployment

Stage 1

Business Understanding

Introduction

A study based on the Recommendation System Algorithms in Machine Learning:

- **Content-Based Filtering**
- **Collaborative Filtering**
- **Neural Collaborative Filtering**

Objective

- To develop a movie recommendation system that enhances user engagement by delivering personalized movie recommendations.
- By evaluating recommendation algorithms (content-based, collaborative filtering), the objective is to determine the most effective method for providing high-quality recommendations, with a focus on user satisfaction.

Benefits

Scalable Solution: Implementing a highly efficient recommendation system can provide personalized experiences for millions of users, leading to higher engagement and potential revenue growth.

Research Contribution: By comparing multiple algorithms, the project can contribute to the literature, demonstrating which methods work best for movie recommendation scenarios.

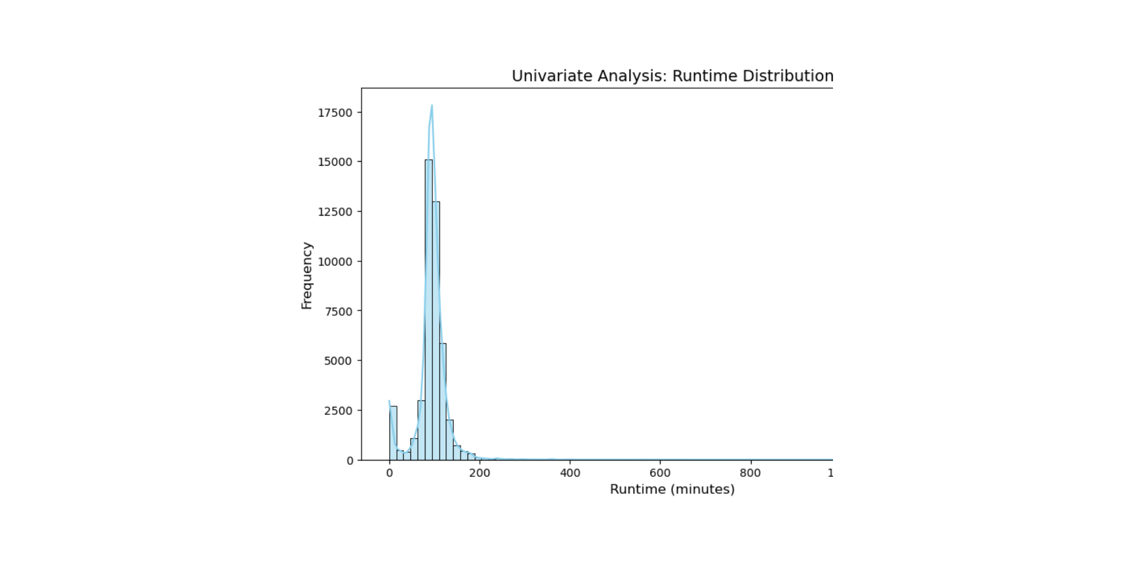
User Engagement: Enhanced user satisfaction through accurate recommendations could lead to increased platform usage and reduced churn.

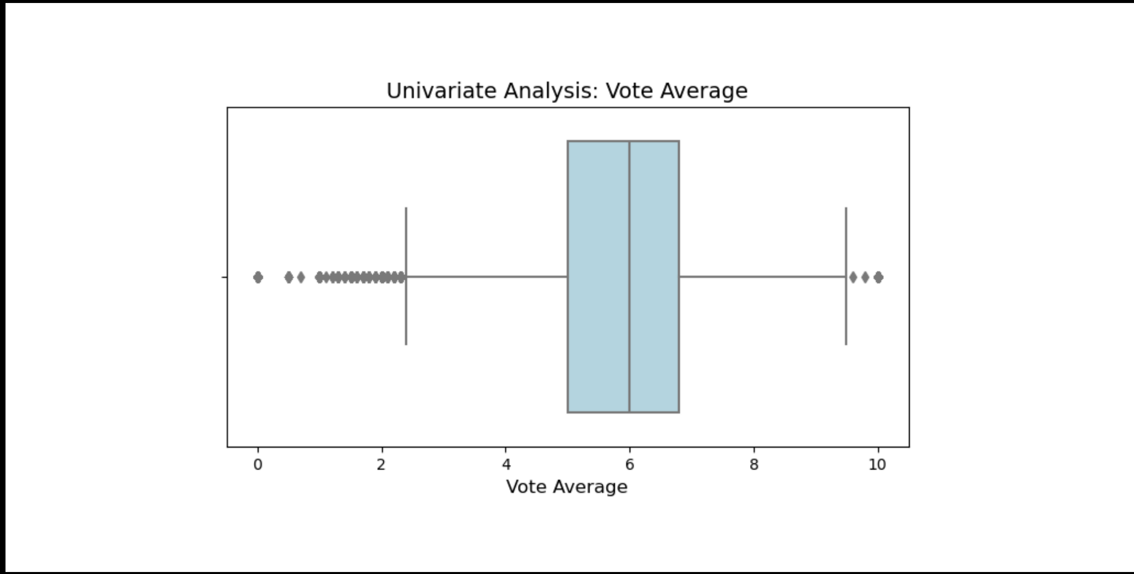
Stage II

Data Understanding

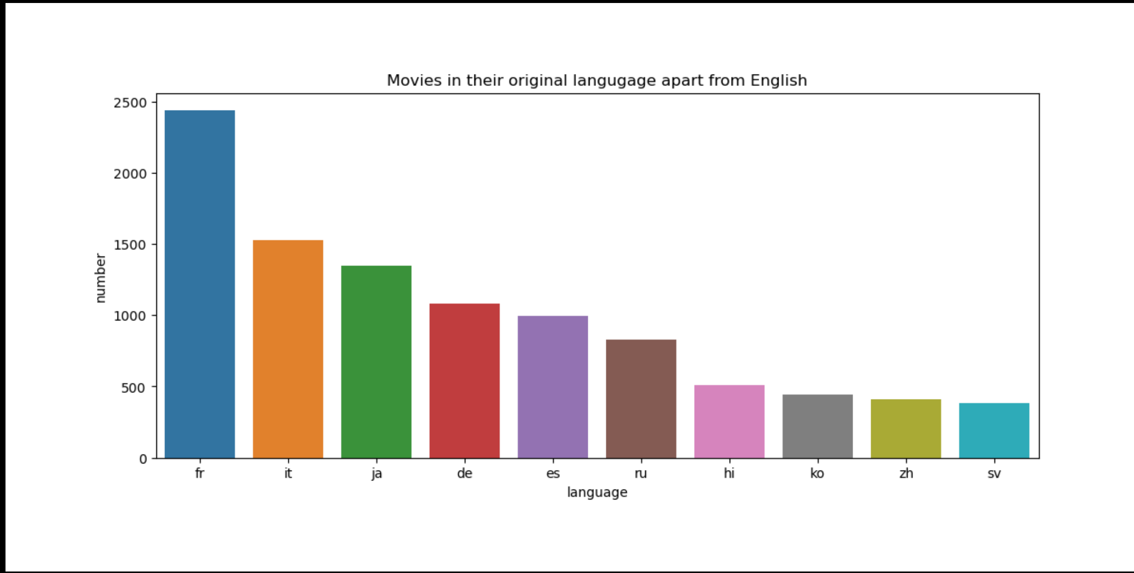
Data

- The raw data file consists of 45,466 movies and 24 corresponding features.
- Final features: Adult, Budget, id, original_language, popularity, production_companies, production_countries, runtime, spoken_languages, title, vote_average, vote_count


Univariate Analysis: Runtime Distribution
Frequency
17500
15000
12500
10000
7500
5000
2500
0
0
200
400
600
800
Runtime (minutes)


Univariate Analysis: Vote Average
0
2
4
6
8
10
Vote Average


Love
Man
Girl
Night
Day
Life
Dead
Movie
Story
Last
World
Time
House
Death
Little
Blood
Black
Boy
One
War
Christmas
American
Return
Heart
Blue
Great
King
Moon
Men
Island
Woman
Live
Home
Dark
Star
Dog
Sea
Year
City
New
Devil
Secret
Big
White
Summer
Road
Tale
Wild
Good
Red
Killer
People


Movies in their original langugage apart from English
number
2500
2000
1500
1000
500
0
fr
it
ja
de
es
ru
hi
ko
zh
sv
language

Stage III

Data Preparation

Data Preprocessing

Handling missing values:

- The data contained a total of 105,562 cells with missing values.
- Columns such as 'homepage','belongs_to_collection', 'original_title', 'overview', 'poster_path', 'release_date', 'revenue', 'status', 'tagline', 'video', 'imdb_id were dropped as they are not needed for the analysis.
- The rows which has no movie title were also dropped.
- These two steps reduced the missing value count to 268.

Handling Duplicate values:

- 30 movies were duplicated in the file which were removed.

Data imputation:

- 11 values for 'original_language' were imputed manually
- 257 null values for the 'runtime' column were filled with 0.

Data Transformation:

- For content-based filtering, the data for columns such as 'genres', 'production_companies' , 'production countries', 'spoken_langugages' needs to be in a list format. The key-value pairs were transformed to a simple list.

After pre-processing, the shape of the final data was (45466, 13)

Stage IV

Modeling

Recommendation System algorithms:

Content Based

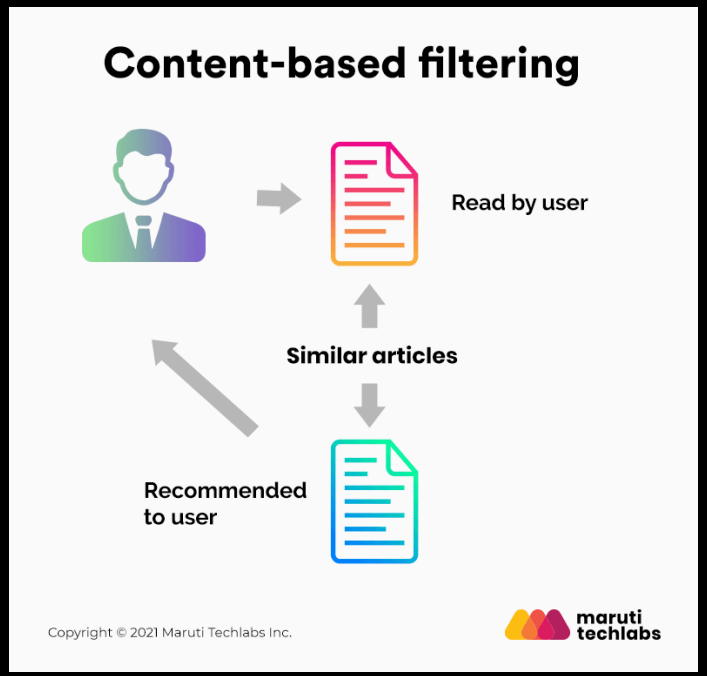
- Content-based filtering methods are mainly based on the description of an item and a profile of the user's preferred choices. In content-based filtering, keywords are used to describe the items, whereas a user profile is built to state the type of item this user likes.
- Focus is on the category used for filtration such as the genre, actor, director, rating etc
- For example, if a user likes to watch movies such as Mission Impossible, then the recommender system recommends movies of the action genre or movies of Tom Cruise.
- Evaluation Metrics: Cosine Similarity Matrix

Collaborative

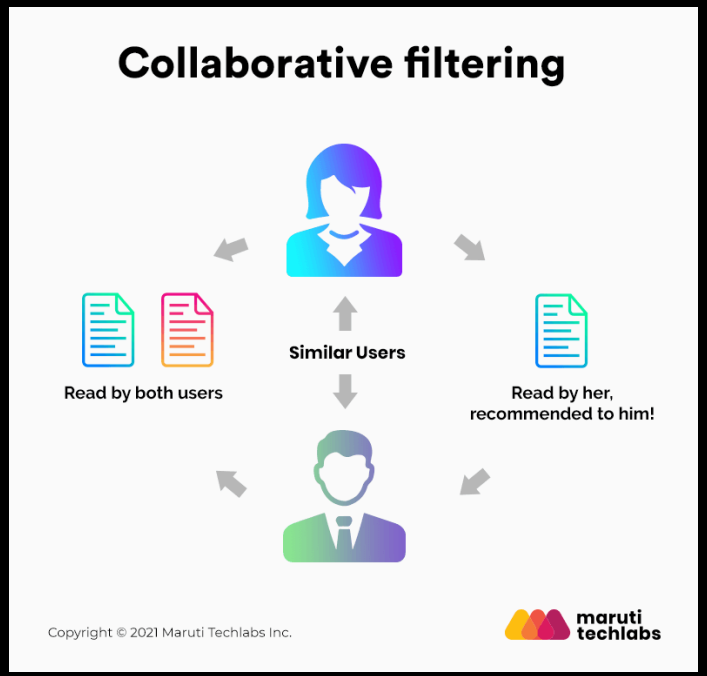
- The collaborative filtering method is based on collecting and analyzing information based on behaviors, activities, or user preferences and predicting what they will like based on the similarity with other users.
- For example, if X likes Comedy and Thriller, Romance while user Y likes Romance, Comedy, and Sci-fi, they have similar interests. So, there is a high probability that X would like Sci-fi and Y would enjoy Thriller.
- One of the main advantages of the collaborative filtering approach is that it can recommend complex items accurately, such as movies, without requiring an understanding of the item itself as it does not depend on machine analyzable content.

Recommendation System algorithms

Content Based



Collaborative



Content Based Filtration

Cosine Similarity Matrix

$$cos(x, y) = \frac{x.y}{||x|| * ||y||}$$

In this formula:
x: The embedding vector of an item you've liked in the past
y: The embedding vector of another item
$(x.y)$: The dot product between the two vectors
||x||*||y||: The cross product between the two vectors

Dataframe:

	adult	budget	id	original_language	popularity	production_companies	production_countries	runtime	spoken_languages	title	vote_average	vote_count	cast	crew	keywords	combined_features
0	False	30000000	862	en	21.946943	[Pixar Animation Studios]	[United States of America]	81.0	[English]	Toy Story	7.7	5415.0	[Tom Hanks, Tim Allen, Don Rickles, Jim Varney...	[John Lasseter]	[jealousy, toy, boy, friendship, friends, riva...	Tom Hanks Tim Allen Don Rickles Jim Varney Wal...
1	False	65000000	8844	en	17.015539	[TriStar Pictures, Teitler Film, Interscope Co...	[United States of America]	104.0	[English, Français]	Jumanji	6.9	2413.0	[Robin Williams, Jonathan Hyde, Kirsten Dunst,...	[Joe Johnston]	[board game, disappearance, based on children'...	Robin Williams Jonathan Hyde Kirsten Dunst Bra...

Movies recommended based on the cast/specific actor:

```
Movies featuring Tom Hanks:
['Toy Story', 'Apollo 13', 'Forrest Gump', 'Philadelphia', 'Sleepless in Seattle', 'The Celluloid Closet', 'That Thing You Do!', 'Saving Private Ryan',
"The 'Burbs", 'Splash', 'The Money Pit', 'Nothing in Common', "You've Got Mail", 'Big', 'Return with Honor', 'Toy Story 2', 'The Bonfire of the Vanitie
s', 'The Green Mile', 'A League of Their Own', 'Volunteers', 'Bachelor Party', 'Punchline', 'Cast Away', 'Turner & Hooch', "He Knows You're Alone", 'Joe
Versus the Volcano', 'Road to Perdition', 'Catch Me If You Can', 'Radio Flyer', 'Dragnet', 'The Ladykillers', 'The Terminal', 'The Man with One Red Sho
e', 'The Polar Express', 'From the Earth to the Moon', 'The Da Vinci Code', 'Cars', 'Who Killed the Electric Car?', 'The Simpsons Movie', "Charlie Wilso
n's War", 'The Great Buck Howard', 'Angels & Demons', 'Shooting War', 'Toy Story 3', 'The Pixar Story', 'Larry Crowne', 'Extremely Loud & Incredibly Clo
se', 'The War', 'Cloud Atlas', "The Rutles 2: Can't Buy Me Lunch", 'Captain Phillips', 'Toy Story of Terror!', 'Saving Mr. Banks', 'Killing Lincoln', 'H
awaiian Vacation', 'Small Fry', 'Elvis Has Left the Building', 'Partysaurus Rex', 'Toy Story That Time Forgot', 'And the Oscar Goes To...', 'The Circl
e', 'Bridge of Spies', 'Everything Is Copy', 'The Sixties', 'A Hologram for the King', 'The Extraordinary Voyage', 'Sully', 'Inferno', 'Prohibition', 'I
thaca', 'Band of Brothers']
```

Movies recommended based on the cast, keywords, crew:

[63]:
```
# Test the recommendation function
recommend_movies('Dilwale Dulhania Le Jayenge')
```

[63]:
```
13998         Stone of Destiny
1006        The Sound of Music
993                 Cinderella
32308      Coo of The Far Seas
6161          Winged Migration
32023                 Robotrix
27110    The Mysterious Island
36418              Cabin Fever
36419              Son of Mine
5946             Honkytonk Man
Name: title, dtype: object
```

Since the movies have been recommended based on the keywords, cast and crew. These suggestions seem somewhat valid as all of them are musical hits.

[69]:
```
# Test the recommendation function
recommend_movies('Zero Effect')
```

[69]:
```
1866          Friday the 13th Part 2
37024                         Female
37025                    Loco Fever
412                       Barcelona
12760              Zombie Strippers!
1802                 Doctor Dolittle
38819    The Philadelphia Experiment
26197              Hate Thy Neighbor
6586                         The Rose
7670                  Little Murders
Name: title, dtype: object
```

Collaborative Recommendation

SVD Algorithm:

- Builds user-item matrix
- Matrix factorization with SVD
  - SVD factorizes this user-item matrix into lower-dimensional matrices (typically user factors and item factors) that represent users and items in a latent feature space.
  - This allows the algorithm to make predictions for missing entries (unseen movies for a user) by estimating the ratings based on patterns observed across the dataset.
- Cross-validation: The model was evaluated using RMSE and MAE on multiple folds, which is an excellent metric for understanding how well your model generalizes.
- When the model makes predictions on the test set, Surprise calculates how close these predictions are to the actual ratings in the test set using RMSE (Root Mean Square Error) and MAE (Mean Absolute Error).

Made Predictions for Specific Movies:

- Ratings for specific user-movie pairs were predicted to see personalized recommendations.

Generated Top-N Recommendations

- Create a function to recommend top-N movies for each user based on predicted ratings.

Dataframe:

	userId	movieId	rating	timestamp
0	1	110	1.0	1425941529
1	1	147	4.5	1425942435
2	1	858	5.0	1425941523
3	1	1221	5.0	1425941546
4	1	1246	5.0	1425941556

Predicted ratings for a movie based on user's activity:

```
Predicted rating for movie 5 by user 1: 3.4338074002756267
```

Recommend movies to a user 1 using collaborative filtering alone without relying on specific user pairs.

```
Top 5 movie recommendations for user 1 based on collaborative filtering:
   movieId  predicted_rating
0    64241          5.000000
1   159819          5.000000
2    26453          4.973530
3   159817          4.946916
4     5194          4.916187
```

Neural Collaborative Recommendation

This method uses neural networks to learn latent factors in a user-item interaction matrix. It's a more complex approach to collaborative filtering.

Neural Collaborative Filtering (NCF): Builds a multi-layer neural network to predict user-item interactions by learning the non-linear relationship between users and items.

Implementation: Used Tensorflow to build a neural network model for collaborative filtering.

Train and Test Data: 90:10 ratio

Algorithm:

Inputs: User Input: A single integer representing the user ID.

Item Input: A single integer representing the item (movie) ID.

Embedding Layers: Each input is passed through an embedding layer:

User Embedding: Maps user IDs to dense vectors of size embedding_size (50 here).

Item Embedding: Maps item IDs to dense vectors of size embedding_size.

Flattening: The embeddings for users and items are flattened into 1D vectors.

Concatenation: The flattened user and item embeddings are concatenated to form a single feature vector.

Fully Connected Layers: A feed-forward neural network processes the concatenated vector:

Dense Layer 1: 128 units, ReLU activation.

Dense Layer 2: 64 units, ReLU activation.

Output Layer: A single neuron outputs the predicted rating (a continuous value).

Loss and Optimization:

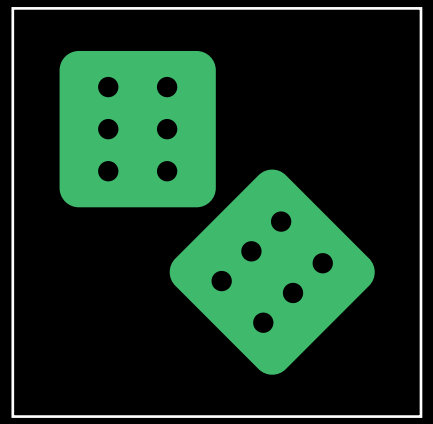
Loss: Mean Squared Error (MSE) to minimize the difference between predicted and actual ratings.

Optimizer: Adam for efficient training.

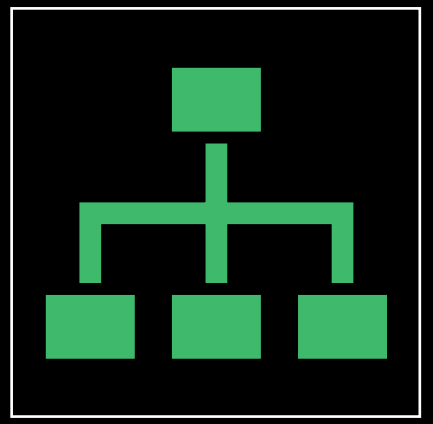
Stage V

Evaluation

Metrics for Evaluation



Root Mean Squared Error (RMSE): Measures how much the predicted ratings deviate from the actual ratings.



Mean Absolute Error (MAE): Measures the average absolute difference between predicted and actual ratings.

Performance Evaluation of Collaborative Filtering:

- The low standard deviation across folds in RMSE and MAE shows that the model is performing consistently, indicating good predictive stability.

- Both RMSE and MAE values are below 1 which indicate that the average difference between predicted and actual ratings is low. This shows that the model is more accurate.

Evaluating RMSE, MAE of algorithm SVD on 5 split(s).

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE (testset)	0.7960	0.7963	0.7969	0.7966	0.7959	0.7964	0.0004
MAE (testset)	0.6022	0.6022	0.6027	0.6025	0.6021	0.6023	0.0002
Fit time	574.57	42685.13	418.09	429.53	350.19	8891.50	16896.97
Test time	110.28	119.40	110.20	127.62	79.28	109.35	16.37

Performance Evaluation of Neural Collaborative Filtering:

- Lower RMSE suggests the model predicts ratings closer to the actual values, but it does not directly reflect the proportion of variance explained.
- These results indicate an R^2 value of 0.3664. The RMSE (0.8481) and MAE (0.6398) values align with this R^2 score, showing the model is not perfect but reasonably accurate in its predictions.
- The model is moderately good for providing personalized recommendations but may struggle with edge cases or less popular items.
- Specifically useful for long term deployments

```
RMSE: 0.8481489160023922
MAE: 0.6397985924028495
```

Stage VI

Deployment

Deployment and Challenges:

Recommendation systems improve user satisfaction by making it easier to discover relevant content, reduce churn rates for platforms, and drive revenue through enhanced engagement and retention.

This recommendation system with an advanced training has future scope for deployment on platforms where users can input their preferences to get tailored suggestions.

- Linkedin search recommendation
- Streaming services
- Media and Entertainment platforms
- Ticket booking platforms: Suggest movies currently playing in theaters based on past bookings or preferences.
- Online learning platforms(Udemy, Coursera): Suggest educational films or documentaries related to courses or interests.

Limitations:

•The project uses a limited dataset and simplified models due to resource constraints.

Thank you

Shivani Battu
sbattu1@kent.edu