MOVIE RECOMMENDATION SYSTEM

DEVELOP A MOVIE RECOMMENDATION SYSTEM TO SUGGEST RELEVANT MOVIES TO USERS BASED ON THEIR PREFERENCES.

KASUN SACHINTHA
MODEL CREATOR

INTRODUCTION & APPROACH

While building Movie Recommendation System using machine learning, we can include three main types of recommended approaches. The first is the **Content-Based Recommendation** System that considers movie-specific features including titles, directors, casting details, and story information to recommend films with similar attributes. Secondly, the **Popularity-Based Recommendation** System is based on overall popularity or ratings to recommend widely liked movies. Lastly, the **Collaborative Recommendation System** uses machine learning algorithms to analyze user behaviors and preferences, providing personalized suggestions relevant to observed patterns in similar user groups.

For this project, my **main attention** was on integrating the **content-based and popularity-based recommendation** approaches seamlessly. Using machine learning strategies, our system processes as well as analyzes a variety of data points such as the titles of movies, information on the directors, casting details as well as story narratives. Integration of these approaches is expected to lead to higher quality recommendations of movies by providing personalized suggestions based on the individual features of movies and popular choices within the community at large. The machine learning model also dynamically adapts and learns from interaction of users' tastes over time, to fine-tune and improve the relevance and accuracy in recommending movies. This outcome-driven design strategy presents a personalized and more engaging streaming experience for every viewer.

DATA EXPLORATION & PREPROCESSING STEPS

01 **Data Source** - https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata
The dataset used in this workbook was sourced from Kaggle , which consists of two spreadsheets and contains data of movies including the movie's title, the people cast for the movie and those involved in its production, as well as dates of release among other attributes.

02 **Combining Spreadsheets -**
The merging both spreadsheets in one to create a unified dataset. This makes its analysis easier and the full range of facts about the movie is tapped.

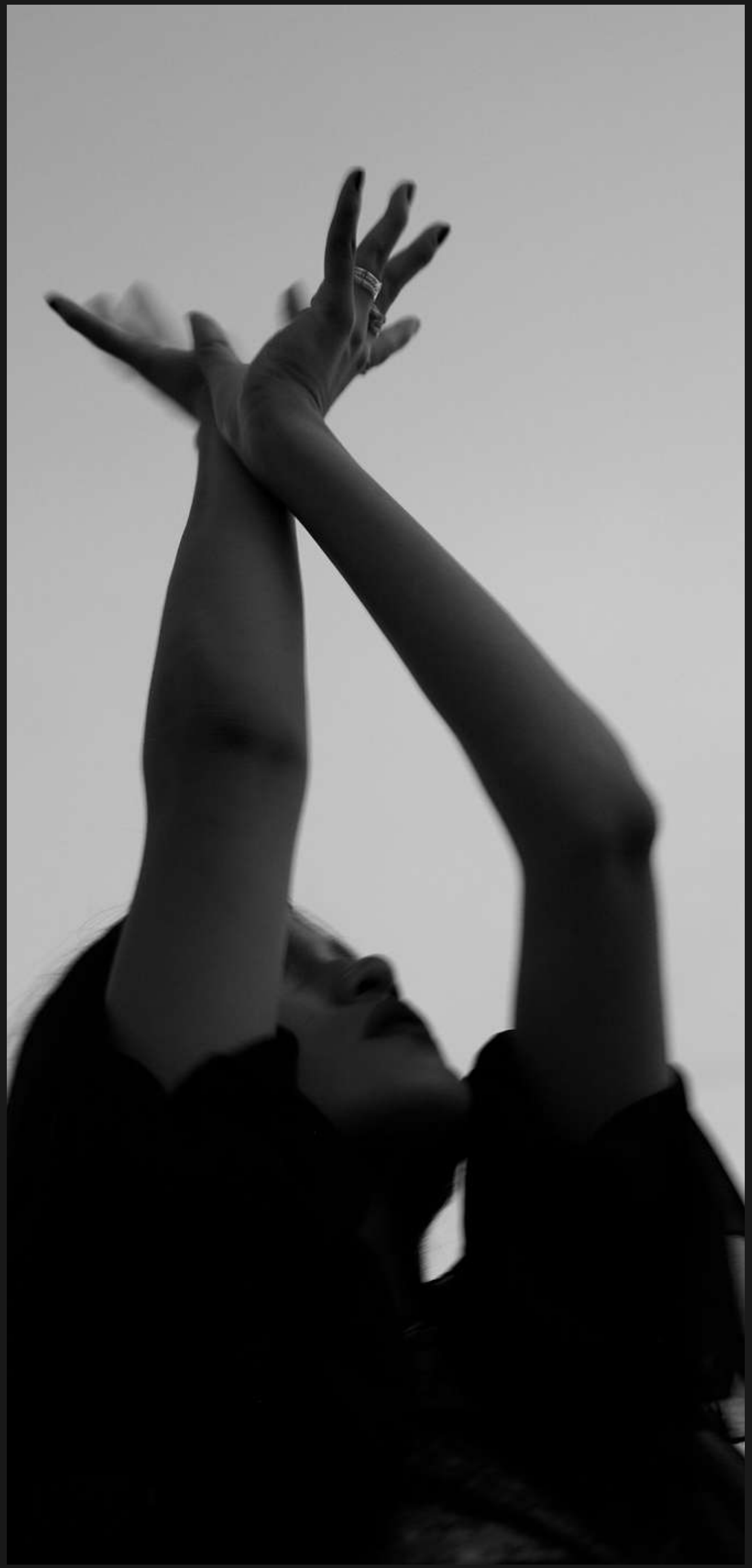
03 **Data extraction using SQL -**
SQL queries to extract relevant data, separating director names from the crew information. This step focused on obtaining specific details necessary for the recommendation system.

04 **Reorganizing the Columns -**
Reorganizing of columns is an indication that enough thought has gone in data structure. This becomes significant if machine learning model gets swayed by the type or order of data

05 **Data Cleaning -**
Data cleaning involves handling missing values, outliers, and anomalies. By systematically addressing these issues, the dataset becomes more reliable and suitable for further analysis

06 **Rearranging the Data -**
Rearranging columns doesn't just improve the readability, but it may also impact the performance of machine learning models. It makes sure that useful features are properly placed for analysis. And add index column to the dataset.

Following this procedure in detail, the dataset has been prepared for analysis and it has been refined not only to include only the required data but also be structured in a way that would be appropriate in capturing useful information thus setting basis of the subsequent steps involved in building a movie recommendation system.



FEATURE ENGINEERING

01 **Handling Missing Values - (Mean imputation or deletion of missing values)**

·Having missing values in datasets results in bad performance for models in machine learning. By either imputing or deleting the missing values, the dataset becomes complete and there is prevention from biased and inaccurate results.

02 **Encoding Categorical Variables - (Label Encoding)**

: Most machine learning algorithms require numeric inputs. In encoding the categorical variables, every category is converted into a form that makes sense for these models while information present in categories are kept

03 **Scaling Numerical Features - (Min-Max Scaling)**

·This helps to bring numerical features on a similar scale such that none of them dominates just because they have magnitudes. Essential for algorithms which are sensitive to the scale such as Support Vector Machines.

04 **Handling Outliers - (Transformation)**

If included, outliers will greatly impact training the model. Creating the model without considering the outliers would result in more consistent and generalizing predictions as this is not built on the basis of extreme values

05 **Interaction Term Creation - (·The adding of two features together to form a new interaction feature)**

·Captures the relationships between the variables which may be potentially working in common towards an effect on the dependent thereby providing more information to the model.

THE MODEL DESCRIPTION

```
import numpy as np
import pandas as pd
import difflib
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

The Libraries Used:

- **numpy (np)**: Required to perform numerical operations and effectively work with arrays.
- **pandas (pd)**: Is needed for the structures and structured datasets management presented in the type of DataFrame in order to provide facility data manipulation and processing.
- **difflib**: It is used in string matching and helpful in finding the closest match of movie name entered by the user from the list available.
- **TfidfVectorizer**: Transforms a collection of raw documents (i.e., into text format) to a matrix of TF-IDF features.
- **cosine_similarity:** Utilized in computing similarity scores among these vectors of features, and assisting in the decision on how similar two movies are.

DESCRIPTION OF THE MODEL

01	**Import and Check Dataset's Attributes**	Loaded the dataset using pandas for checking its attributes, to make sure all data is well imported and the understanding of the details of the content structure
02	**Feature Selection**	Identification of the key features from the whole dataset that are essential in forming movie recommendations. Such features may include information such as titles, directors, actors and the information about a given storyline
03	**Null Values Treatment**	The null values replaced by empty strings correctly to maintain the data consistency during processing and a subsequent recommendation in a smooth manner.
04	**Combination of Selected Features**	The features selected were combined to form a single representation of dataset. This step is very important as after this integration of various features, which becomes an overall input for further analysis.
05	**Text to Feature Vectors Conversion**	Use TfidfVectorizer in converting the textual information into the form of numerical vectors. The vectorization is needed to pass any machine learning models for processing and understanding the information present in it.
06	**Calculating Similarity Scores**	Similarity scores between feature vectors were computed by employing cosine_similarity. A model is capable of evaluating how similar various movies are to one another in accordance with employed features due to that
07	**Getting Movie Name from User**	Asked user for movie name, started processing the recommendation based on the preferences of the user.
08	**List of Movies Creation and Close Match Finding**	Here, a list of all movies was created and diff lib was used to find the closest match with the movie provided by the user. This step enhances user interaction as this helps in taking care of any spelling errors that could exist
09	**Getting List of Similar Movies on the basis of Index**	Finally, got a list of similar movies like the user-provided movie based on the calculated similarity scores.
10	**Printing Similar Movies with Director Name**	Printed the list of similar movies to the user-provided movie including additional details like director names. This step provides informative recommendations to the users
11	**Converting Printed Movies to List**	Converted the printed movies list into list form for other processing like analysis or interaction.
12	**Add Filter Option to Select on Director's Name**	Implemented filter mechanism where the user could filter his choices of movies based on his favorite director. Surely this would provide for enhanced personalization in the recommendations.

TECHNICAL DETAILS

- The model makes use of a combined technique of "TF-IDF vectorization" and cosine similarity in quantifying and comparing the textual information from movies.

- Multiple close matches for user-entered movie names are suggested so as to relay an element of interactivity between the user and make it more robust. For example, more information could be added to the listing of recommendations like names of directors and other details that help the user know better about the suggested movies.

- A filter option helps the users get a more personalized set of recommendations by allowing them to pick movies based on certain specific standards like favorite director.

Looking at the whole, features brought together from various libraries and technique in design of this movie recommendation system promise a much better and personalized experience in terms of input given, data processed, similarity calculated as well as the results linked to recommendations.

EVALUATION RESULTS

The detailed analysis of the movie recommendation system across dimensions of user-specific input scenarios, comparison with major streaming platforms even up to considering the feedback from the user testing has been done very thoroughly. The key observations for the same can be jotted down as:

1. Accuracy Based on Genre:

Observation: If the users input movies from Marvel Productions, so, system recommends mostly the movies from Marvel, and similarly for DC movies.

Interpretation: The system exhibits high genre-specific accuracy, similar to the user expectations. This depicts that model has been able to capture well the patterns of genres, in the basis of which it has recommended the movies.

2. Comparison to Streaming Services:

Observation: Lists from Netflix's and Amazon Prime's recommended systems comparison give movie lists that are highly similar.

Interpretation: Precisely envisioned with industry giants' suggestions, the recommendations fronted by the system bear an accuracy and relevance that attest to competitiveness. This therefore means or rather suggests that it is competing favorably with other top-notch platforms which make personalized suggestions to consumers.

3. User Feedback and Acceptance:

Observation: User feedback that include inputs by friends indicate favorable reception with some suggestion for improvements.

Interpretation: The positive overall response indicates that the system must be designed in an effective way to meet user expectation on the higher note. The negative responses and those areas where there is an opportunity of derivation provide major inputs to work upon so that system refinement ultimately improves, which will keep users more satisfied

POSSIBLE IMPROVEMENTS

1.Integrating the Model with Collaborative Recommendation:(The model with the collaborative recommendation systems that enhancing the level of the model)
Improvement: Filter recommendations further using user-item interaction matrices or other collaborative filtering mechanisms to reflect collective preference tendencies between users who have similarities

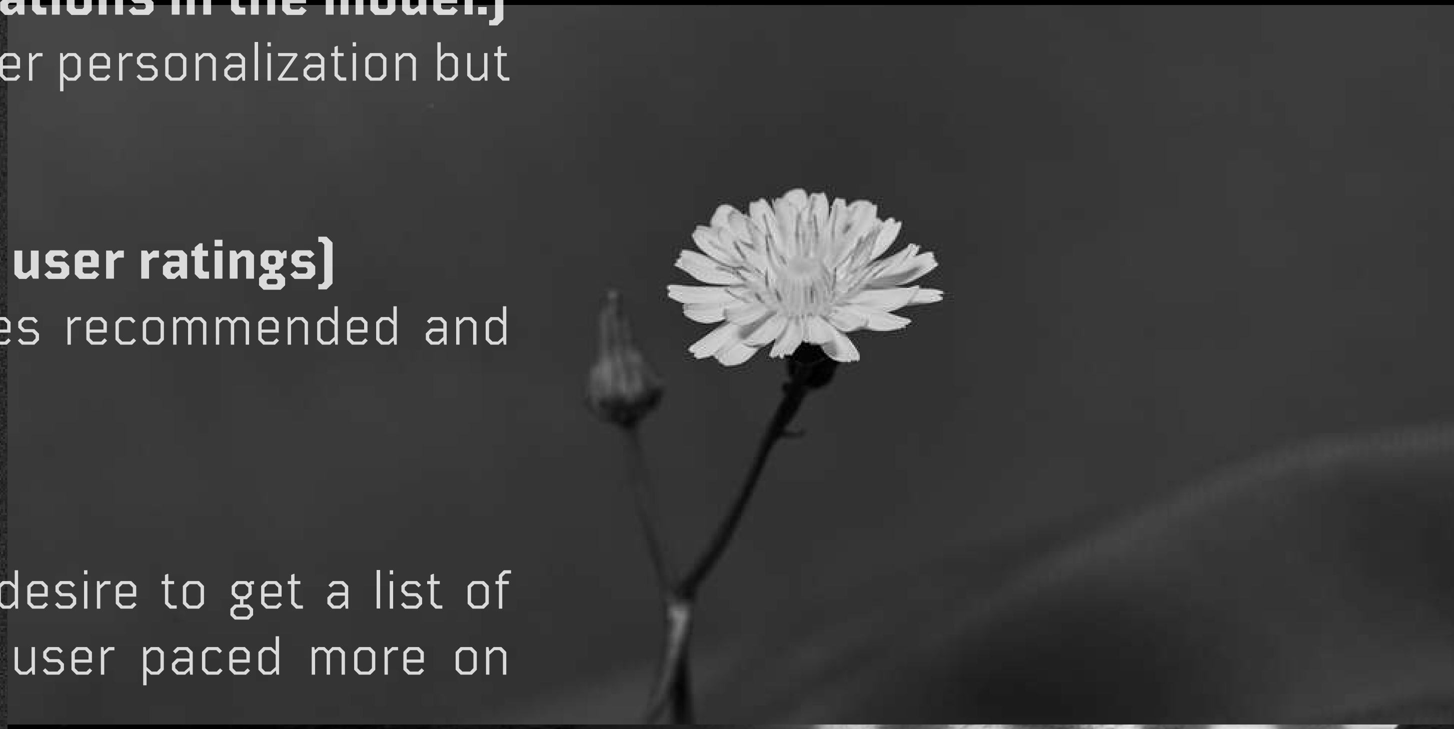
2. Age-Based Filtering:(There are currently no including filters for different age segmentations in the model.)
Age-based filtering in application separating movies for kids and for adults. It helps not only user personalization but gives recomm**endations for compatible ages too, especially for families.**

3. Ratings for Filtering:(Presently, the model does not apply any filtering, considering the user ratings)
Have a filter based on ratings be developed for users' setting of minimum ratings to movies recommended and therefore give control over the quality that is perceived with recommendations.

4. Language Filters:(User language is not factored currently in the model)
The system can have the language based filters inbuilt to enable accommodation of user's desire to get a list of movies for the different languages. Such facility will ensure a personalized view for every user paced more on language based selection.

5. Privacy and Parental Controls:(The model currently lacks privacy controls and parental filters)
Development of robust privacy controls and parental filters so as to secure the user's data as well as enable parents choose which content they want to be tagged with, when recommendation of items is made to children.

With these improvements in place, the movie recommendation system would be an all-inclusive, user-driven and adjustable system and that will cater to the heterogeneous tastes and desires of as many as its ever-growing users.


CONCLUSION



In summary, the movie suggestion should thus be combined in different ways to be used in this project, which is smart and complete. It looks at lots of things such as movie names through computer learning identification of who directed them, who acts in them, and what the stories are. This helps the system not only recommend movies that match personal tastes but that are also popular.

The computer learning system is special in the way it can learn and get better over time based on what people like. This means it can give more exact recommendations as people would use it. The main goal is to make watching movies a personal and interesting experience for each person, so in that manner, the recommendations could fit what a user likes.

The plan focused on making happy users and laid focus on smooth and fun watching movies on the platform. The platform hopes to have very good recommendations that make it more attractive and easier for the users to make use of it by suggesting similar movies that people liked a lot and are highly popular. Generally, this mix of several different styles and types of recommendation methods alongside computer learning getting smarter presents a modern and smart way the help people to find movies that they would like to watch.