Title: Personalized Film Recommendation and IMDb Prediction System

Team Members:
Lakshmi Dheeraj Oruganti (11601229)
Viswak Sena Palaparthi (11580431)
Harini Popuri (11607363)
Durga Prasad Tulasi (11610834)
Venkateswara Rao Pasupuleti (11659091)

Abstract:
The goal of this project is to use machine learning to construct a personalised Film Recommendation and IMDb Prediction System. The system is constructed as a regression problem, with the goal of predicting film ratings based on characteristics such as genre, director, and user comments. The project uses algorithms such as Random Forest and Support Vector Regression to analyse data from IMDb and Netflix. To assess the accuracy of these predictions, we employ assessment metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE), which we compare to actual IMDb ratings.

Introduction:
The digital revolution in the entertainment industry has resulted in an overwhelming number of movie alternatives, emphasising the need for improved recommendation algorithms that are tailored to individual preferences. This project intends to improve the viewer experience by creating a system that not only customises film recommendations but also predictions IMDb ratings, combining user data with advanced machine learning algorithms. The relevance and possibilities of our approach are further validated by looking at related efforts such as Netflix's recommendation algorithms and numerous IMDb rating prediction models, which demonstrate the project's original approach.

Background:
The emergence of streaming services and digital media, revolutionized how audiences generally engage with film content, significantly affecting their expectations for personalized viewing experiences. Just when audiences face an ever-increasing selection of films, they encounter the so called “paradox of choice”, which can make decision-making very difficult.

Within this changing setting, this initiative strives to simplify viewer choices by providing personalised film suggestions and informative film rating projections. The technology is designed to match content with individual interests using cutting-edge machine learning and data analytics, resulting in increased viewer pleasure and engagement.

Furthermore, as viewing habits and digital consumption evolve, the need for flexible solutions becomes increasingly obvious. This project intends to not only improve the film selection process, but also to keep up with shifting preferences and trends. Its proactive strategy ensures that the recommendation system is relevant and successful for a wide range of audiences. This dedication to continual innovation and adaptation distinguishes the initiative as a watershed moment in the digital entertainment environment, changing how consumers discover and interact with films.

Experiment Methodology:
Data Sets:
The project makes use of massive datasets from IMDb and Netflix, including genre, director, and user reviews. Data preparation consists of addressing missing data and putting categorical information into a usable format in order to prepare for analysis.

Algorithms and tools:

The project makes use of Python and machine learning frameworks such as scikit-learn. Predictive modelling is done with Random Forest Regression and Support Vector Regression, while the recommendation system uses K-Means clustering and Cosine Similarity.

Results:
Visualisation and analysis:
The findings are displayed with a variety of visuals:
Bar graphs depict the distribution of Netflix content by age rating.
3D bar graphs and pie charts depict the distribution of various genres.
Scatter plots investigate the relationships between film runtime, country of origin, and IMDb ratings.

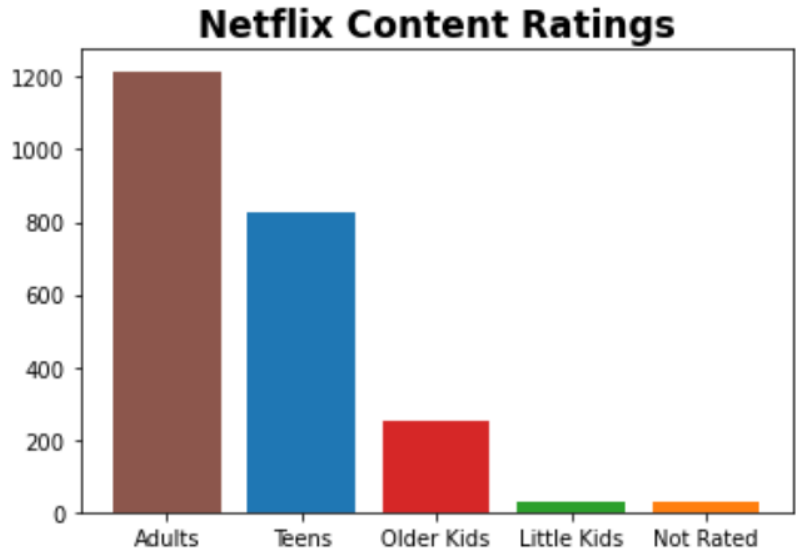
Model Performance:
The initial results suggest that the models have varying levels of accuracy, with Random Forest doing better in categorization. Error metrics are used to evaluate regression models and guide future optimisation.

Related work:
This initiative expands on previously recognised principles, such as Netflix's personalised recommendation tactics and academic studies on predicting IMDb ratings. Insights from hybrid recommendation systems and explainable AI offer important perspectives on improving accuracy and user happiness.

Conclusion:
The study was effective in using machine learning to predict IMDb ratings and personalise film suggestions to individual preferences. Although the results are positive, future efforts will centre on fine-tuning models, gathering additional user feedback, and adjusting to changes in audience preferences and industry trends. This technology could have a big impact on consumer interaction with digital media as well as film industry decision making.

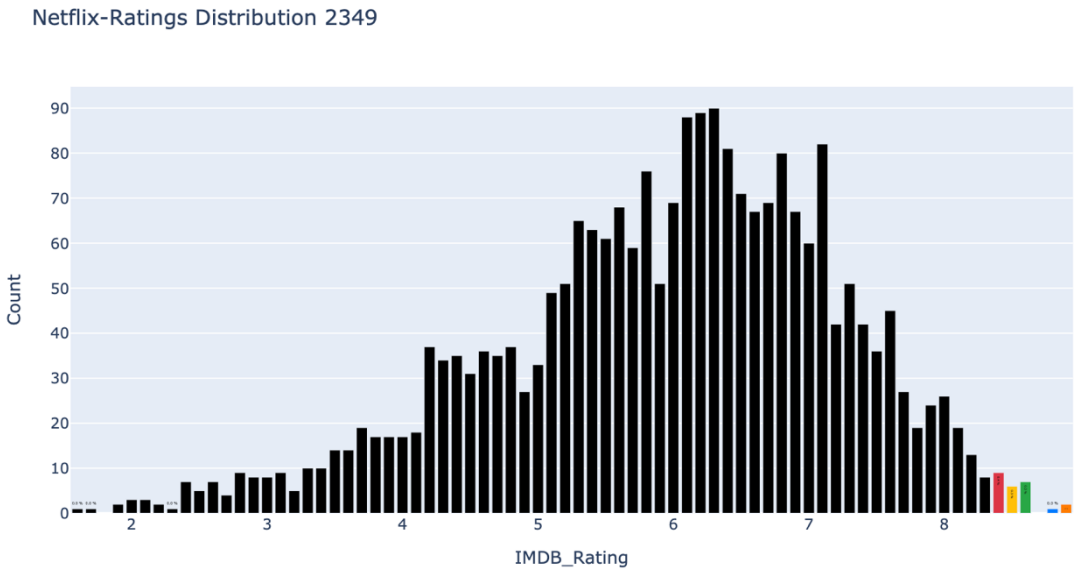
Future work
The project's goal is to enhance algorithms based on user feedback, grow the dataset, and adapt to changing patterns in cinema consumption and industry practices. Future study may look at more powerful machine learning models and real-time data processing to improve the accuracy and relevance of forecasts and suggestions.

Screenshots:
To determine which age group the Netflix content is most relevant to, we have constructed a bar graph. We have drawn a bar graph with rating value counts versus age groups to visually represent it.

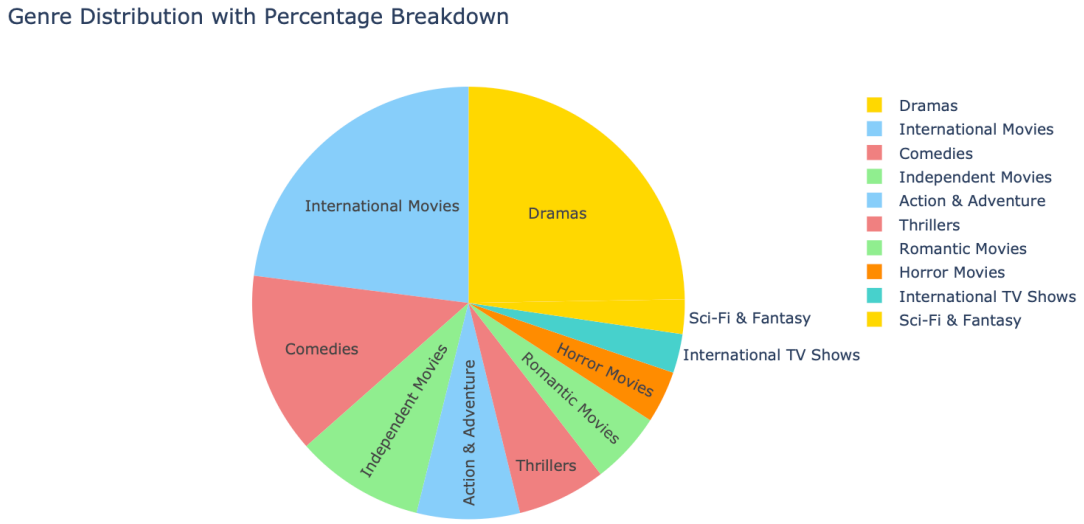


Evidently, the majority of the content (50%)s intended for an adult audience, defined as those who are older than 17 (mature).

Next, we used Plotly to create a 3D bar graph that shows the distribution of every content's ratings according to their counts.



A pie chart illustrating the distribution and percentages of the various genres.



With percentages of 25%, 23%, and 13%, the top three prevalent genres are dramas, foreign films, and comedies.

Detail Design of Methods:

Data Pre-processing:
To create a single dataset based on the titles of movies and TV shows, first load the Netflix Titles dataset and the IMDb Ratings dataset. We must combine them in order to recommend the film or television program.

```
combined_data = pd.merge(left=IMDBmovies_data, right=netflix_data, left_on='title', right_on='title')
combined_data.shape
```

```
]: (2960, 33)
```

<u>Drop the Duplicate Values:</u>
We must remove the numerous duplicate values in the movie titles found in the Combined dataset.

<u>Handling Missing Values:</u>

The dataset's missing values must be located.

<u>Data Cleaning:</u>

Execute cleaning operations on text data, such as cast bios, genres, and movie descriptions:

• Eliminate characters that aren't numbers.

• Change text's case to lowercase.

• Eliminate stop words and punctuation.

To clean up and prepare textual data, apply natural language processing algorithms.

Models:

Unsupervised Models:

For recommendation, we have employed two unsupervised machine learning algorithms. Kmeans and Cosine similarity algorithms have been employed. In order to employ the left attributes in the machine learning models, we first vectorized them by eliminating punctuation, stop words, and alpha numeric characters from the categorical attributes, such as Genere, Description, and Cast.

Using the elbow approach to determine the number of clusters, we first employed Kmeans algorithms to predict the clusters that corresponded to similar television series or motion pictures. PCA was utilized in this to minimize the dimensions. All things considered, KMeans seemed to function fairly effectively.

Supervised Models:

Employed supervised algorithms to predict IMDb ratings. Initially, classification models were employed.KNN and Random Forest are the models we utilized for categorization.

1. Made use of the Random Forest Classifier In order to enhance the efficiency and precision of our algorithm, we employed various techniques, including adjusting hyperparameters and utilizing key features.
Utilized GridSearchCV in 2.

• CNN:

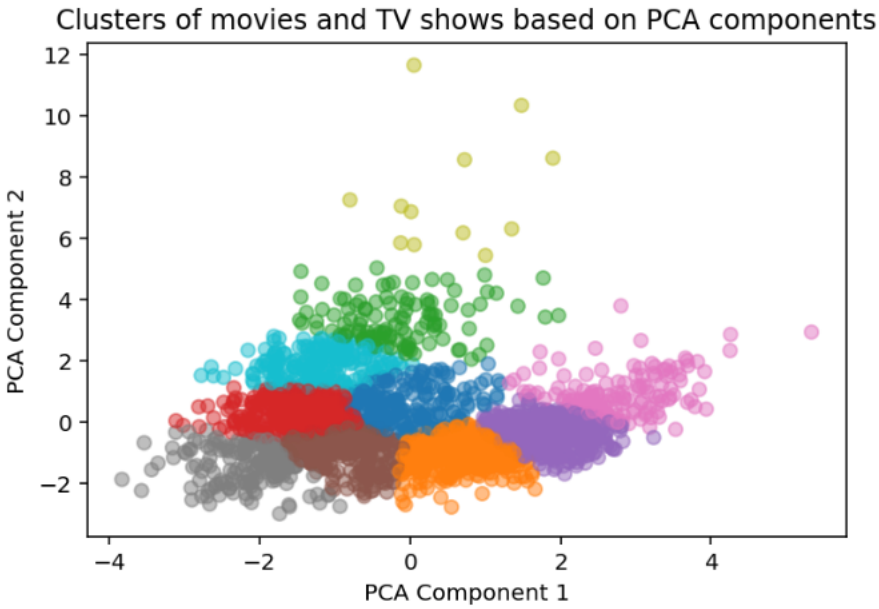
1. To obtain the best possible outcome, certain new columns have to be defined first.
2. Vectorization on a string including columns was then performed.
The dataset was separated into three sections: the train set, validation set, and test set.
4. KNeighborsClassifier was used.

then measured the Mean Square Error, Mean Absolute Error, and Root Mean Square Error for 4 regression models: KNN, Random Forest, Decision Tree, and Linear Regression.

Preliminary Results:

Recommendation System:
Kmeans:

The Kmeans result is displayed in the image below, along with clusters that collectively represent related TV series or motion pictures. Consequently, we can quickly suggest a film based on previously seen films.



Cosine Similarity:

The outcome of the suggested films based on the movies you have already watched is displayed in the image below.

```
film_recommend('Godzilla').head(5)
```

	Movie_Title	Score
0	Beyond Evil	0.342498
1	One Day	0.280386
2	Yellowbird	0.260208
3	Victor	0.259828
4	Kingpin	0.257248

IMDB Rating Prediction System:

Classification:

Random Forest classifier:

The accuracy for random forest classifier is 65%

```
Test Confusion Matrix:
[[ 39  90   0   0]
 [ 25 312   7   0]
 [  2  71  18   0]
 [  1  14   7   2]]

Test Classification report:
              precision    recall  f1-score   support

           0       0.58      0.30      0.40       129
           1       0.64      0.91      0.75       344
           2       0.56      0.20      0.29        91
           3       1.00      0.08      0.15        24

    accuracy                           0.63       588
   macro avg       0.70      0.37      0.40       588
weighted avg       0.63      0.63      0.58       588

Accuracy with Random Forest :0.6309523809523809
```

```
a_hyper_param = accuracy_score(diyte, grid_predss)
print(a_hyper_param)
```

```
0.8928571428571429
```

In order to improve accuracy, we then built a data frame including description, cast, and genre features that was built by using a counter vectorizer.
Accuracy is attained as:

```
print("Random Forest Accuracy:"+str(rf_acc_cv))
```

```
Random Forest Accuracy:0.9421768707482994
```

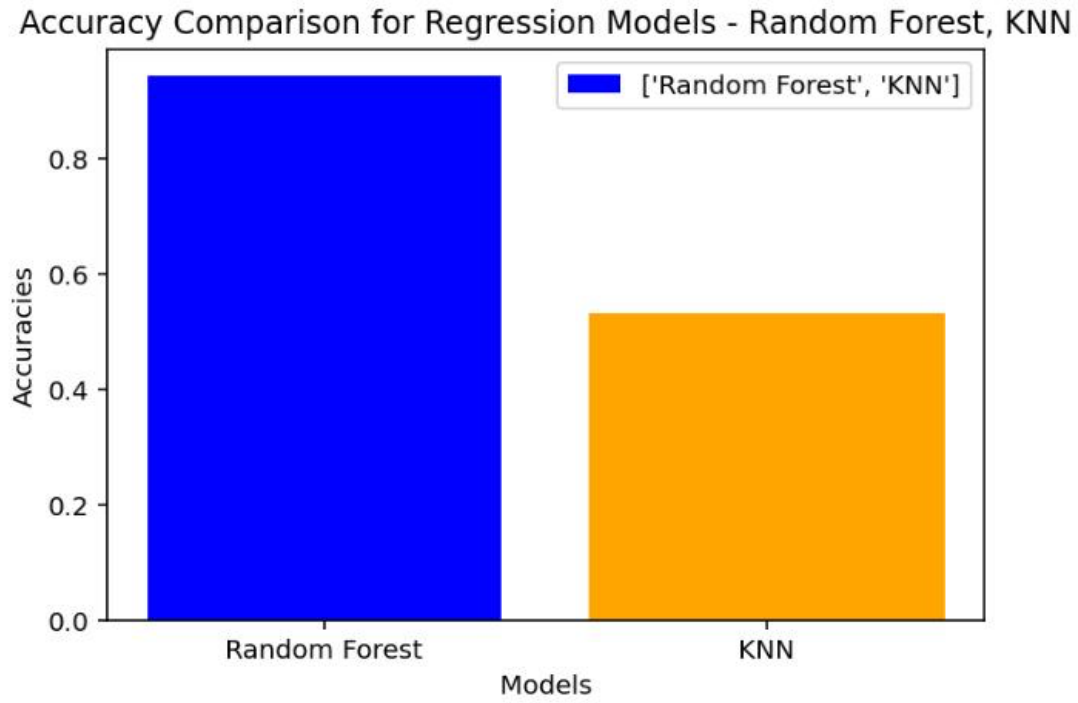
KNN classifier:

The accuracy for random forest classifier is 54%

```
knnModel = KNeighborsClassifier(43, metric="euclidean")
knnModel.fit(dxtr, dytr)
ypred = knnModel.predict(dxte)
knn_score = accuracy_score(dyte, ypred)
print(knn_score)
```

```
0.5446808510638298
```

After comparing the two classifiers, it is evident that the Random Forest Classifier outperforms the KNN Classifier.



After comparing the root mean square, mean absolute error, and mean square error of each of the four regressors, we found that linear regression outperforms the other three regressors.

Linear Regression:

```
For Linear Regression:
Mean Absolute Error(MAE): 1.2352127659574468
Mean Squared Error (MSE): 2.534547872340426
Root Mean Squared Error (RMSE): 1.5920263415975333
```

Random Forest Regression:

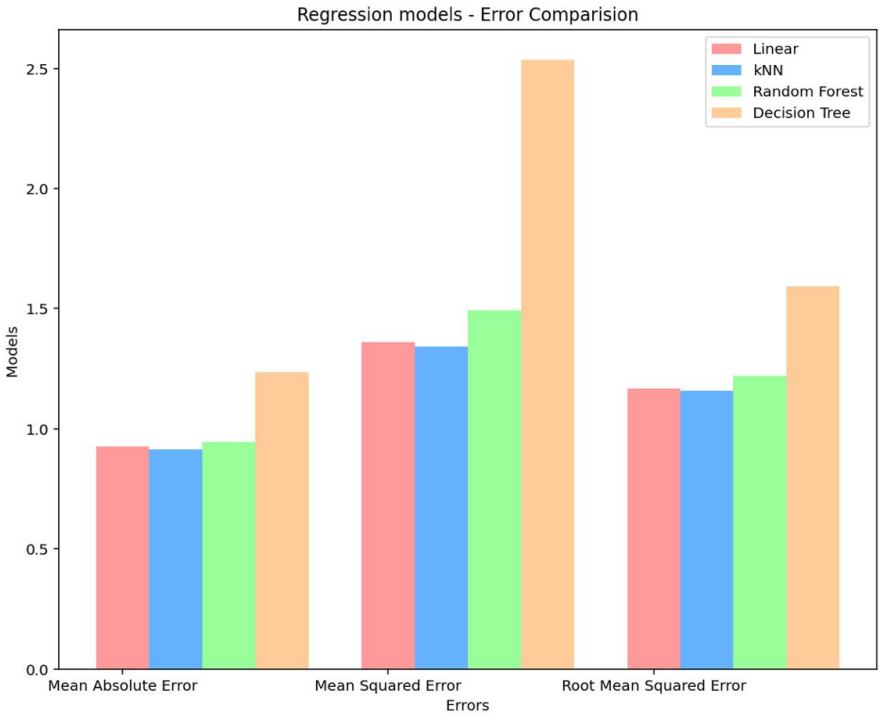
```
Random Forest Regression :
Mean Absolute Error (MAE): 0.9454749999999997
Mean Squared Error (MSE): 1.4912083407700942
Root Mean Squared Error (RMSE): 1.2211504169307295
```

Decision Tree Regression:

```
For Decision Tree:
Mean Absolute Error(MAE): 1.2352127659574468
Mean Squared Error(MSE): 2.534547872340426
Root Mean Squared Error(RMSE): 1.5920263415975333
```

KNN Regression:

```
For KNN:
Mean Absolute Error (MAE): 0.9136650652024707
Mean Squared Error (MSE): 1.3424513250824717
Root Mean Squared Error (RMSE): 1.1586420176579442
```



Conclusion:

Considering everything, In the classification phase of the IMDb prediction algorithms, KNN and RF initially offer similar accuracy values; however, upon applying the contour vectorizer, we observe that RF provides a greater accuracy value. When we compared the regression against KNN, RF, and Decision Tree regressions, we found that linear regression performed better. Additionally, cosine similarity outperformed k-means in our recommendation algorithms.

References:

[1] Augustine, A., & Pathak, M. (2008). User rating prediction for movies. Technical Report. University of Texas at Austin.

[2] Abarja, R. A., & Wibowo, A. (2020). Movie Rating Prediction usingConvolutionalNeural Network based on Historical Values. International Journal, 8(5).

[3] Dixit, P., Hussain, S., & Singh, G. (2020). Predicting the IMDB rating by using EDAand machine learning Algorithms.

[4] Sang-Ki Ko, Sang-Min Choi, Hae-Sung Eom , Jeong-Won Cha, Hyunchul Cho,Laehyum Kim, and Yo-Sub Han: A Smart Movie Recommendation System Content-based method uses item-to item similarity. If a user like B, we recommend A that is similar to B.