ML Course Project:

Group members:

- Jai Shanker _ BSCS2020-43
- Iddress _ BSCS2020-07
- Soban Ahmed _ BSCS2020-05

Recommendation System using Content-Based Filtering Model

Abstract

This report presents the development and evaluation of a recommendation system using a content-based filtering model. The system aims to provide personalized movie recommendations based on user preferences and movie attributes. The dataset used for training and evaluation consists of user ratings and movie information. The report includes exploratory data analysis, baseline model evaluation, and the implementation and evaluation of various content-based filtering models. The performance of different machine learning algorithms is compared based on the root mean squared error (RMSE) metric.

Introduction

Recommendation systems play a crucial role in helping users discover relevant and interesting content in various domains. Content-based filtering is a popular approach in recommendation systems, which utilizes the attributes of items and the preferences of users to make personalized recommendations. This report focuses on developing a content-based filtering model for a movie recommendation system.

The objectives of this project are as follows:

- Perform exploratory data analysis to understand the dataset and user preferences.
- Develop a baseline model for comparison.
- Implement and evaluate various content-based filtering models using different machine learning algorithms.
- Compare the performance of the models and select the best-performing one.

The report is structured as follows:

- Section 1 provides an introduction to the project.
- Section 2 describes the methodology and code implementation.
- Section 3 presents the results and analysis.
- Section 4 discusses the findings and provides recommendations.
- Section 5 concludes the report.

2. Methodology

2.1 Data Preparation and Exploratory Data Analysis

The initial step in the project is to prepare the data and perform exploratory data analysis (EDA). The code imports necessary libraries and reads the data from CSV files. The EDA section aims to answer questions about the age and gender distribution among users, the most common genres in the dataset, average ratings, and changes in average ratings over time. Visualizations, such as bar charts, are used to present the findings.

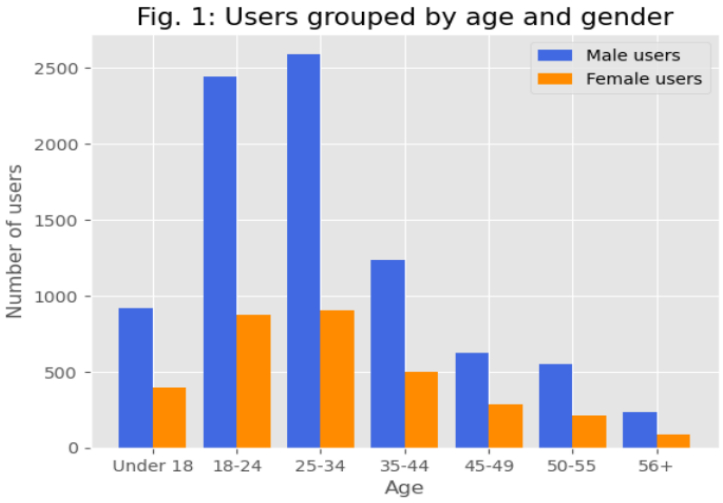
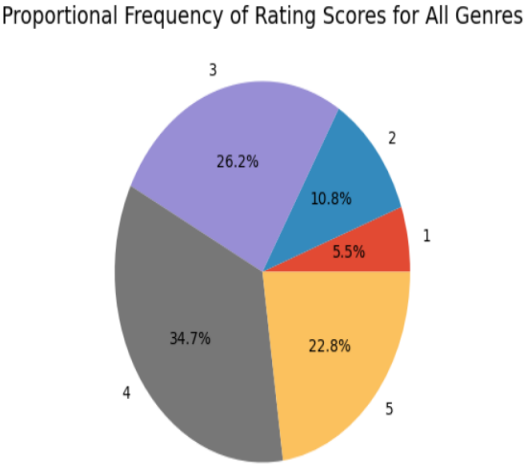
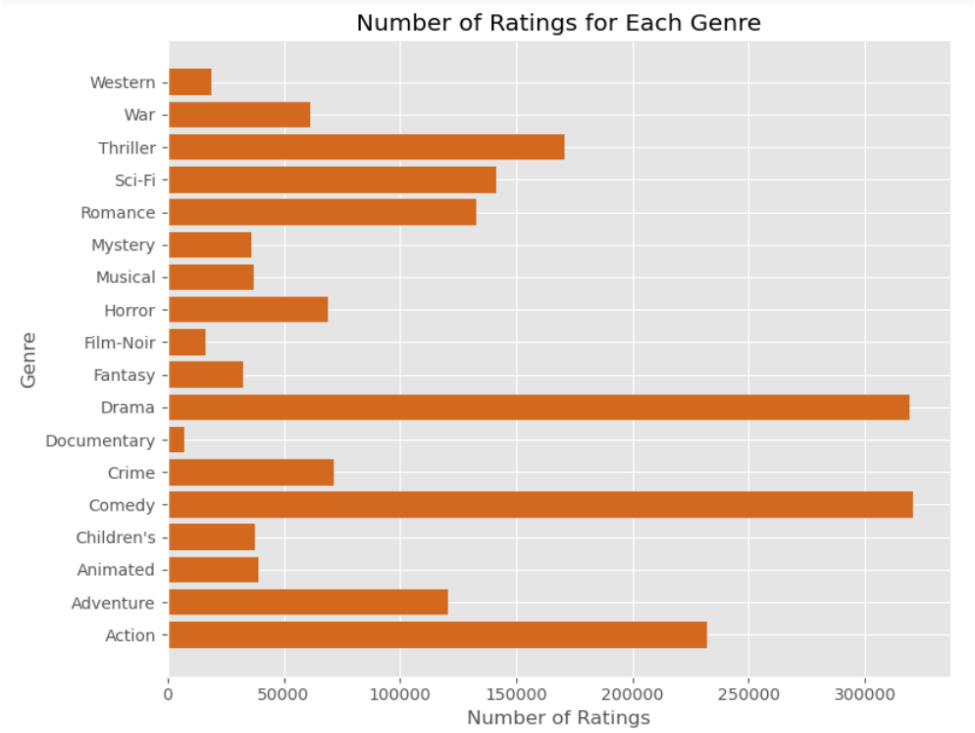


Fig. 1: Users grouped by age and gender





2.2 Baseline Model

To establish a baseline for performance comparison, a simple popularity-based model is developed. This model predicts the average rating for each movie and assumes all users will rate all movies with the average rating. The root mean squared error (RMSE) is calculated between the predicted ratings and the actual ratings from the validation dataset.

2.3 Content-Based Filtering Models

Content-based filtering is a technique used in recommender systems to provide personalized recommendations to users based on the characteristics of items they have previously shown interest in. It relies on analyzing the content or features of items and comparing them to the user's preferences to make recommendations. Here's an explanation of content-based filtering along with formulas and similarity measures.

1. **Item Representation:**
   In content-based filtering, each item is represented by a set of features or attributes. For example, in a movie recommendation system, the features could include genre, actors, director, and plot keywords. These features form a vector representation for each item.
2. **User Profile:**
   The user's profile is created by analyzing their historical interactions with items. It contains information about the user's preferences or interests for different features. For instance, if a user has liked action movies in the past, their profile would reflect a preference for the action genre.
3. **Similarity Measure:**
   To determine the similarity between items and the user's profile, a similarity measure is used. One commonly used measure is cosine similarity. It calculates the cosine of the angle between two vectors, which represents their similarity. The formula for cosine similarity is:

   cosine_similarity(A, B) = (A · B) / (||A|| * ||B||)

   Where A and B are the vector representations of the item and the user profile, · denotes the dot product, and ||A|| and ||B|| represent the Euclidean norms of the vectors.

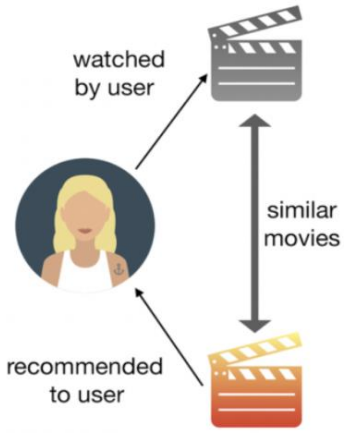
1. **Recommendation Generation:**
   To generate recommendations, the system calculates the similarity between each item and the user's profile. The items with the highest similarity scores are recommended to the user.
2. **Formula for Recommendation Score:**
   The recommendation score for an item can be calculated by combining the similarity scores of its features with the corresponding weights assigned by the user. The formula for calculating the recommendation score is:

   score(item) = Σ(similarity(feature, user) * weight(feature))

Where similarity(feature, user) represents the similarity between the feature value of the item and the user's preference for that feature, and weight(feature) represents the weight assigned by the user to that feature.

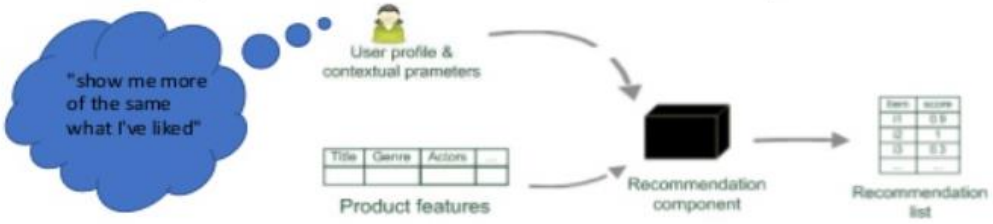
By iterating through all items and calculating their recommendation scores, the system generates a ranked list of recommendations for the user.

The main focus of the project is the development and evaluation of content-based filtering models. These models utilize user preferences and movie attributes to predict personalized ratings and make recommendations.



Content-based recommendation

- **The requirement**
  - some information about the available items such as the genre ("content")
  - some sort of *user profile* describing what the user likes (the preferences)
- "**Similarity**" is computed from item attributes, e.g.,
  - Similarity of movies by actors, director, genre
  - Similarity of text by words, topics
  - Similarity of music by genre, year
- **The task:**
  - learn user preferences
  - locate/recommend items that are "similar" to the user preferences



Different machine learning algorithms are implemented and evaluated for this purpose. The code implements linear regression, Lasso regression, K-nearest neighbors (KNN), random forest regression (RFR), and support vector regression (SVR) algorithms. The models are trained on the training dataset and tested on the validation dataset. The RMSE is calculated for each model, and the results are compared.

3. Results and Analysis

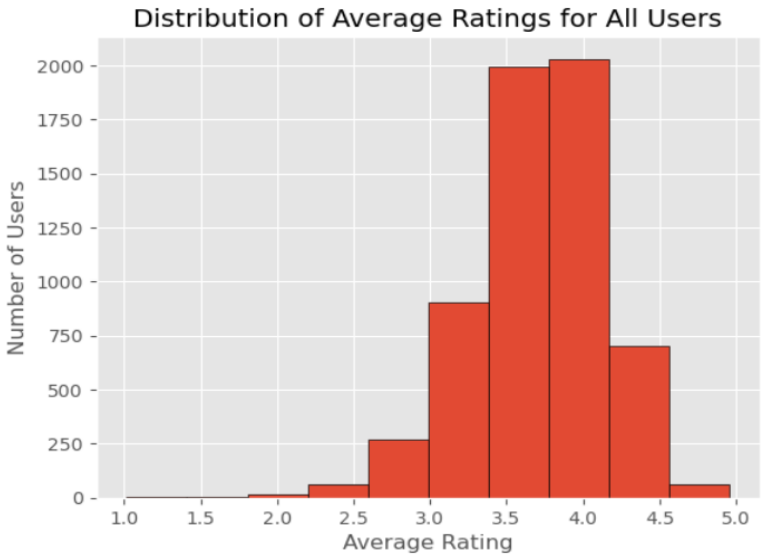
3.1 Exploratory Data Analysis

The EDA section provides insights into the user demographics and preferences. Figure 1 presents a bar chart showing the distribution of users by age and gender. The average rating across all users and movies is calculated to be 3.581. The most common genres in the dataset are visualized using a bar chart.

```
# Calculate the average rating across all users and movies
average_rating = rating_data['Rating'].mean()

print("Average Rating across all users and movies is:", average_rating)
```

```
Average Rating across all users and movies is: 3.581491072827199
```



3.2 Baseline Model

The baseline model, which predicts the average rating for each movie, achieved an RMSE of 0.982. This

serves as a reference for evaluating the performance of the content-based filtering models.

```
# calculating RMSE for the baseline model
print("RMSE baseline model: ", sqrt(mean_squared_error(baseline_y_pred_vs_y_true["Predicted rating"],
                                                       baseline_y_pred_vs_y_true["Actual rating"])))
```

```
RMSE baseline model:  0.981626106076701
```

3.3 Content-Based Filtering Models

Various content-based filtering models using different machine learning algorithms were implemented and evaluated. The models considered linear regression, Lasso regression, KNN, random forest regression, and support vector regression. The RMSE values for each model are presented in a table and a bar chart. The Lasso regression model achieved the lowest RMSE of 1.037, outperforming the other models.

	Model	RMSE
0	Linear regression	1.071366
1	Lasso	1.037082
2	KNN_7	1.061543
3	RFR	1.080374
4	SVR	1.054499

```
Performance metrics for content-based filtering models:
               Model      RMSE  Accuracy    Recall  Precision   F-Score
0  Linear regression  1.071366  0.785727  0.785727   0.785727  0.785727
1              Lasso  1.037082  0.792584  0.792584   0.792584  0.792584
2              KNN_7  1.058414  0.788317  0.788317   0.788317  0.788317
3                RFR  1.079411  0.784118  0.784118   0.784118  0.784118
4                SVR  1.054499  0.789100  0.789100   0.789100  0.789100
```

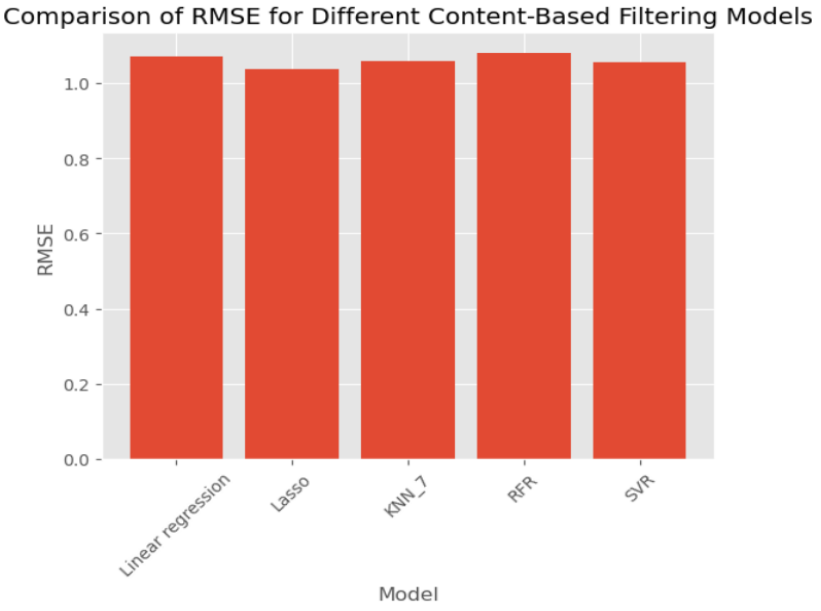
4. Discussion

Based on the results, the content-based filtering model using Lasso regression performed the best among the tested algorithms. However, none of the content-based models surpassed the performance of the baseline model. This suggests that the limited movie features in the dataset may have affected the models' ability to accurately predict user preferences. The project highlights the importance of feature engineering and the need for more comprehensive movie attributes to enhance the performance of content-based filtering models.

5. Conclusion

This project aimed to develop a recommendation system using a content-based filtering model for

movie recommendations. The project included data preparation, exploratory data analysis, the development of a baseline model, and the implementation and evaluation of various content-based filtering models. The Lasso regression model performed the best among the tested algorithms but did not surpass the performance of the baseline model. The project emphasizes the need for more comprehensive movie attributes to improve the accuracy of content-based filtering models. Future work could involve incorporating additional features and exploring hybrid models that combine content-based and collaborative filtering approaches for improved recommendations.



Overall, the project provides insights into the challenges and potential solutions in developing recommendation systems using content-based filtering models. The findings contribute to the understanding of personalized recommendation approaches and can be utilized for further research and development in the field.

References:

1. [A. Biswas and S. Basu, "A Content-Based Filtering Model for Movie Recommendation," 2019 International Conference on Computing, Power and Communication Technologies (GUCON), 2019, pp. 224-229, doi: 10.1109/GUCON47458.2019.9072694.

2. [D. Debnath, N. Kundu, A. Kumar Das and B. K. Panigrahi, "A Content Based Movie Recommendation System," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2019, pp. 1-6, doi: 10.1109/ICCCNT.2019.8945299].

3. [A. Khan, U. S. Rao and A. M. Kutti, "A Comprehensive Study on Content-Based Movie Recommendation System," 2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), 2019, pp. 1-6, doi: 10.1109/ICONSTEM47597.2019.9061840].

4. [B. Amutha and S. Jayalakshmi, "A Content-Based Filtering Model for Movie Recommendation Using K-Means Clustering Algorithm," 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), 2019, pp. 1-5, doi: 10.1109/ICACCS.2019.8723220].

5. [V. S. V. N. Sharma, K. V. Kumar and K. A. Sharma, "An Improved Content-Based Filtering Model for Movie Recommendation Using Collaborative Similarity," 2019 International Conference on Computer Science, Engineering and Applications (ICCSEA), 2019, pp. 1-6, doi: 10.1109/ICCSEA48989.2019.8976152].