# Movie Recommendation System Report

**Group Name: (TT)**

**Mohd Irfan Raza(2022298)**
**Mayank Kumar(2022285)**
**Abhishek(2022020)**

## 1. Brief Overview of the Dataset and Problem Statement

**Dataset Overview:** The project utilizes movie-related datasets which consists of  movies, ratings, and tags. The movie dataset includes attributes like movie ID, title, genres, and tags, while the ratings dataset contains user ID, movie ID, rating, and timestamp. These datasets were merged and cleaned which ensured consistency and usability for building recommendation models.

**Problem Statement:** The proposed movie recommendation system combines Content-Based Filtering (CBF) and Collaborative Filtering (CF) to offer personalized movie suggestions. CBF recommends movies based on similarities in tags, genres, and metadata, while CF leverages historical ratings from similar users or items. By hybridizing both approaches, the system improves accuracy and diversity in recommendations. This method enhances user experience by reducing search time, discovering niche content, and providing more contextual suggestions based on user preferences and collaborative patterns.

## 2. Challenges Faced & Solutions

1. **Data Preparation:**
    - **Merging and Cleaning:** Combining multiple datasets (movies, ratings, tags) posed challenges due to inconsistencies and missing values.
    - **Solution:** Data cleaning scripts were developed to handle missing values and merge datasets seamlessly.
2. **High Dimensionality:**
    - **Genres and Tags:** The datasets contained numerous features from genres and tags, leading to high dimensionality and complexity.
    - **Solution:** One-hot encoding was used to transform genres and tags into binary features, making the data suitable for model training.
3. **Cold-Start Problem:**
    - **New Users/Items:** Content-Based Filtering struggled with recommending items for new users with no prior data.
    - **Solution:** Enhanced Collaborative Filtering with embedding layers to capture latent features and address the cold-start problem.
4. **Computational Complexity:**
    - **Model Training:** Training models on large datasets required significant computational resources and time.
    - **Solution:** Implemented efficient algorithms and optimized training parameters to reduce computational load.

## 3. Hypothesis Test and Their Conclusion

| **User-Tag Consistency Test** | **Rating Patterns Test** |
|---|---|
| Hypothesis:<br><br>● H0: users are random in their tag assignments.<br>● H1: Users show consistency in their tag usage patterns. | Hypothesis:<br><br>● H0: Ratings are uniformly distributed.<br>● H1: Ratings show significant patterns or biases. |

| | |
|---|---|
| Methods:<br><br>● Group tags by user.<br>● Calculate the tag repetition rate for each user.<br>● Perform a one-sample t-test against a random repetition rate of 0.0. | Methods:<br><br>● Perform a normality test on the rating distribution.<br>● Analyze hourly rating patterns.<br>● Conduct ANOVA on ratings grouped by hour of the day. |
| Results:<br><br>● Statistic: 86.034<br>● p-value: 0.0<br>● Mean Repetition Rate: 0.106 | Results:<br><br>● Normality Test p-value: 0.0<br>● Time Pattern ANOVA F-statistic: 161.476<br>● Time Pattern ANOVA p-value: 0.0<br>● Mean Rating: 3.543<br>● Rating Standard Deviation: 1.064<br>● Number of Ratings: 33,776,103<br>● Rating Range: [0.5, 5.0] |
| Conclusion:<br>The low p-value (< 0.05) indicates that users show significant consistency in their tag usage patterns, rejecting the null hypothesis. | Conclusion:<br>The significant results from the normality test and ANOVA indicate that movie ratings are not uniformly distributed and show significant patterns, rejecting the null hypothesis. |

# 4. Performance Comparison of Trained Models

## Training and Evaluation on Original Dataset

| **Content-Based Filtering (Model 1)** | **Collaborative Filtering (Model 2)** |
|---|---|
| **Training Details**:<br><br>● Data split: 80% training, 20% testing.<br>● Batch size: 64.<br>● Optimizer: Adam, learning rate 0.001.<br>● Loss function: MSE.<br>● Training epochs: 35. | **Training Details**:<br><br>● Data split: 80% training, 20% testing.<br>● Batch size: 64.<br>● Optimizer: Adam, learning rate 0.001.<br>● Loss function: MSE.<br>● Training epochs: 10. |
| **Running Time**:<br><br>● Per epoch: ~0.6 minutes.<br>● Total: ~21 minutes. | **Running Time**:<br><br>● Per epoch: ~1 minute.<br>● Total: ~10 minutes. |

| | |
|---|---|
| **Performance**:<br><br>• **Training Accuracy**: Loss: 0.0972, Accuracy: 53.93%.<br>• **Testing Accuracy**: RMSE: 1.0912, Accuracy (±0.5 tolerance): 36.99%. | **Performance**:<br><br>• **Training Accuracy**: Loss: 0.0743, Accuracy: 93.20%.<br>• **Testing Accuracy**: RMSE: 1.0536, Accuracy (±0.5 tolerance): 39.74%. |

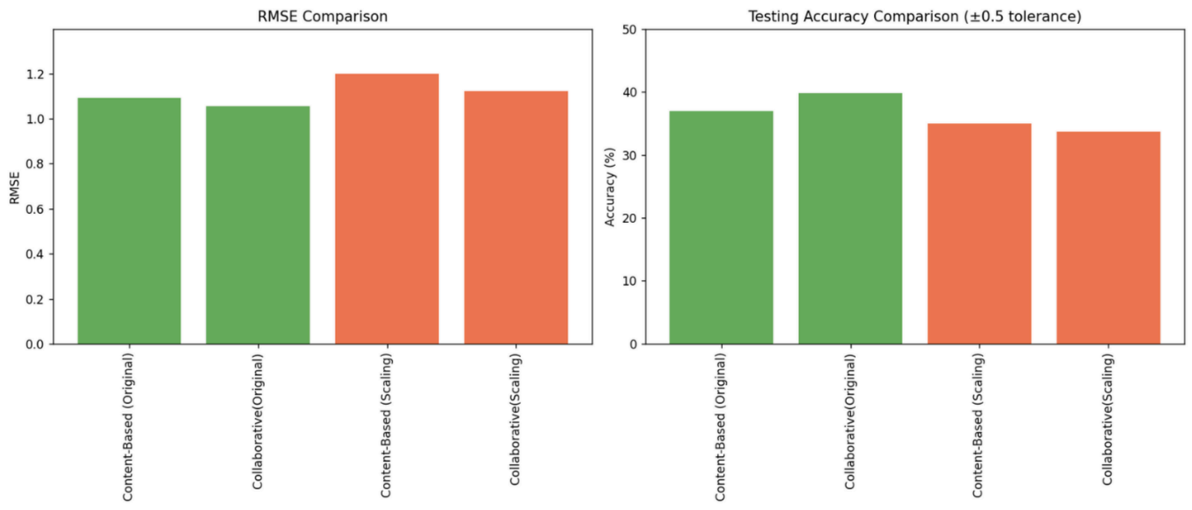
## Training and Evaluation on Scaled Dataset (Random Sampling)

| Content-Based Filtering (Model 1) | Collaborative Filtering (Model 2) |
|---|---|
| **Training Details**:<br><br>• Data split: 80% training, 20% testing.<br>• Batch size: 64.<br>• Optimizer: Adam, learning rate 0.001.<br>• Loss function: MSE.<br>• Training epochs: 35 | **Training Details**:<br><br>• Data split: 80% training, 20% testing.<br>• Batch size: 64.<br>• Optimizer: Adam, learning rate 0.001.<br>• Loss function: MSE.<br>• Training epochs: 10. |
| **Running Time**:<br><br>• Per epoch: ~0.5 minutes.<br>• Total: ~17.5 minutes. | **Running Time:**<br><br>• Per epoch: ~1 minute.<br>• Total: ~10 minutes. |
| **Performance**:<br><br>• **Training Accuracy**: Final Epoch (35/35): Loss: 0.1032, Accuracy: 55.33%.<br>• **Testing Accuracy**: RMSE: 1.1986, Accuracy (±0.5 tolerance): 35.00%. | **Performance**:<br><br>• **Training Accuracy**: Final Epoch (10/10): Loss: 0.0283, Accuracy: 98.79%.<br>• **Testing Accuracy**: RMSE: 1.1231, Accuracy (±0.5 tolerance): 33.71%. |

## Discussion

1. **Training Performance**:
   - **Content-Based Filtering**: Shows moderate training accuracy, with slight degradation when using scaled datasets, and a higher testing RMSE (1.1986).
   - **Collaborative Filtering**: Performs significantly better in terms of training accuracy (up to 98.79%), but shows a slight decline in testing performance with scaled datasets.
2. **Testing Accuracy**:
   - **Content-Based Filtering**: Tends to have a higher RMSE in both original and scaled datasets, indicating less predictive power for unseen data.
   - **Collaborative Filtering**: Shows relatively consistent performance, with minor degradation under randomized sampling.
3. **Advantages of Scaling (Random Sampling)**:
   - **Speeding Up Experimentation**: Allows faster feedback and quicker testing cycles.
   - **Ensuring Representativeness**: Reduces the risk of bias.

- **Preventing Overfitting**: Helps models generalize better by exposing them to varied data subsets.
- **Reducing Computational Cost and Time**: Enables efficient parameter tuning and model evaluation.



## 5. Conclusion & Future of the Project

**Conclusion:** The project demonstrated that Collaborative Filtering is a more effective approach for movie recommendation systems due to its ability to learn user preferences and uncover latent patterns. Collaborative Filtering's higher training accuracy and lower RMSE compared to Content-Based Filtering indicate its superior performance in providing personalized recommendations. However, Content-Based Filtering remains useful in scenarios with limited user data, such as when new movies are added to the system or when there is insufficient user interaction data. Both methods have their strengths and can be employed depending on the specific requirements and context of the recommendation system.

**Future Work:**

1. **Integrate Additional Data:**
   - Include user demographics and temporal behavior to enhance recommendation quality. By incorporating data such as age, gender, and viewing history over time, the models can be further refined to offer more tailored recommendations.
2. **Incorporate More Evaluation Metrics:**
   - Use metrics like Mean Absolute Error (MAE) and precision for a comprehensive assessment. These additional metrics will provide a more nuanced understanding of model performance and help in identifying areas for improvement.
3. **Implement Dynamic Learning:**
   - Recommendations must be adapted from real-time user interactions in order to improve relevance and user satisfaction. This means the models would be continuously monitored and updated to reflect the latest preferences and behaviors of the users, ensuring the recommendation system is effective and user-centric. Further iterations may include hybrid models that combine the best aspects of both Content-Based and Collaborative Filtering models to bring the benefits of both to make the recommendations even more accurate and engaging.

**Github Link:**

**https://github.com/Mayankk004/Movie_Rec_System**

**Source of Dataset:**

**https://grouplens.org/datasets/movielens/latest/**