Data Report

Group 10

---

1. Business Understanding

The objective of this project was to build a movie recommender system capable of suggesting movies to users based on their preferences and historical ratings. The notebook focused on implementing and evaluating different recommendation approaches using the MovieLens dataset.

The main goal was to generate personalized recommendations that accurately predict user ratings. To achieve this, both collaborative filtering (using the Singular Value Decomposition, or SVD, algorithm) and content-based filtering (using movie genres) were developed, followed by a hybrid model that combines the two techniques.

The success of the project was measured through evaluation metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE), which quantify the accuracy of predicted ratings against actual user ratings.

---

2. Data Understanding

The data used in this project consisted of two main files:

- **`movies.csv` —** contained metadata for each movie, including the *movieId*, *title*, and *genres*.
- **`ratings.csv` —** included user-generated ratings with *userId*, *movieId*, *rating*, and *timestamp*.

After loading and merging the datasets, the following statistics were identified:

- Total ratings: 100,836
- Number of users: 610
- Number of movies: 9,742

The ratings ranged from 0.5 to 5.0, and each movie could belong to multiple genres (e.g., “Action|Comedy|Drama”).

To better understand trends in the data, the notebook computed the average rating per genre by splitting the genre strings into individual categories. A bar chart was generated to visualize the mean rating for each genre, showing that some genres tend to receive higher ratings than others.

Overall, the dataset represented a typical movie recommendation scenario with a large number of movies, many users, and sparse rating patterns.

---

3. Data Preparation

Data preparation involved several steps to ensure the data was suitable for modeling:

1. **Merging and Cleaning:**
   The ratings and movies datasets were merged on *movieId*, producing a unified dataset containing both rating and genre information.

2. **Genre Processing:**
   The *genres* column was split by the “|” separator, and the resulting list was expanded so that each movie–genre combination occupied a separate row. This transformation enabled the calculation of average ratings per genre and the use of genre text in the content-based model.

3. **Feature Engineering for Content-Based Filtering:**
   The notebook used the TF-IDF vectorizer to convert genre strings into numerical vectors. These vectors were then used to compute cosine similarity between movies, identifying which movies were most similar in genre composition.

4. **Train-Test Splitting:**
   The dataset was prepared for modeling by dividing the ratings into training and testing sets. This allowed for model performance evaluation on unseen data.

5. **Collaborative Filtering Setup:**
   Using the Surprise library, the data was converted into a format suitable for collaborative filtering, and the SVD algorithm was initialized for training.

---

4. Analysis

The notebook implemented and evaluated three main models: SVD collaborative filtering, content-based filtering, and a hybrid approach combining both methods.

4.1 Collaborative Filtering (SVD)

The SVD model was trained on the user–item rating matrix to learn latent user and movie features. A GridSearchCV was used to find the optimal hyperparameters by testing different values for the number of factors, learning rate, and regularization parameters.

The grid search identified the following best parameters:

- `n_factors`: **20**
- `lr_all`: **0.005**
- `reg_all`: **0.05**

The corresponding best RMSE from cross-validation was approximately 0.874.

Further evaluation using five-fold cross-validation yielded consistent results, with an average RMSE around 0.898 and an MAE around 0.695.

After retraining the model on the full training data and evaluating it on the test set, the final SVD model achieved an RMSE of 0.7138 and an MAE of 0.5626, indicating a strong performance in rating prediction.

4.2 Content-Based Filtering

The content-based filtering model relied on the similarity between movies' genres. Using TF-IDF vectorization on the genre text, cosine similarity scores were calculated for all movie pairs.

For a given movie, the system was able to identify and recommend other movies with similar genre compositions. This method was purely content-driven and did not rely on user behavior or historical ratings.

The notebook generated similarity matrices and sample recommendation lists to demonstrate how movies were matched based on genre similarity.

4.3 Hybrid Recommender System

The hybrid system combined the outputs of the SVD collaborative model and the content-based model. It introduced a `final_score` that represented a weighted combination of the two systems' predictions.

The hybrid approach was designed to leverage the advantages of both models:

- Collaborative filtering captures user preference patterns.
- Content-based filtering handles new or less-rated items.

The hybrid model was evaluated using the same metrics as the other models. The recorded performance on the test data showed:

- RMSE: 1.6956
- MAE: 1.6956

These results indicated that the hybrid's predicted ratings were less accurate numerically than the SVD model on this dataset. Nevertheless, the notebook demonstrated the process of combining the two approaches to create a unified recommendation framework.

---

5. Recommendations

Based on the notebook's workflow and outputs, the following conclusions can be drawn directly from the conducted analysis:

- The SVD collaborative filtering model performed best among the tested approaches, achieving the lowest RMSE and MAE scores.
- The content-based model successfully identified movie similarities using genre information and provided interpretable recommendations.
- The hybrid model integrated both approaches into a single framework, illustrating how multiple recommendation techniques can be combined in a real-world system.