Maithili Luktuke
CSE 272
5/26/23
HW 2 – Recommendation System

Introduction:
In this assignment we were asked to create a recommendation system based on Amazon user reviews which contained numerical ratings and user comments for each item given in the dataset. We were given a few choices for the dataset we could use and I ended up choosing the Movies and TV dataset which gave user reviews. As stated in the assignment description, each entry in the dataset consisted of the following fields:

- User ID: denoted as “reviewerID” in the dataset
- Product ID: denoted as "asin" in the dataset
- Rating: a 1-5 integer star rating that the user assigned to the product, denoted as "overall" in the dataset
- Review: a textual review of the product that the user provided, denoted as "reviewText" in the dataset
- Title: the title of the review, denoted as "summary" in the dataset
- Timestamp: the time that the user made the rating and review
- Helpfulness: contains two numbers, i.e., [#users that think this review is not helpful, #users that think this review is helpful]
- Image: for each product, the dataset provides the image of the product in the form of a 4096-dimensional vector learned by a CNN deep neural network. These vectors are provided in an independent dataset called "Visual Features," which is also available on the website
- Metadata: some metadata information for each product, including product title, price, image URL, brand, category, etc. It is also provided as an independent dataset called "Metadata," which is also available on the website

The Movies and TV dataset consists of a filed called style which conveys the format of the video, for example a VHS tape, DVD, or an Amazon video. For the purposes of this assignment, I only needed the rating, the user ID and the product ID so that would be the overall, reviewerID, and asin fields in the dataset.

Surprise:
For this assignment, I decided to use the surprise library in Python. The Surprise library in Python is a powerful and user-friendly recommendation system library. It provides a comprehensive set of tools and algorithms for building and evaluating recommendation systems. The library simplifies the process of developing recommendation models by offering a high-level interface and integration with popular datasets. The main purpose of the Surprise library is to facilitate the implementation and evaluation of collaborative filtering-based recommendation systems. Other important purposes include giving users control over their experiments, simplifying dataset handling, providing various prediction algorithms, facilitating the implementation of new algorithm ideas, and offering tools for evaluating and comparing

algorithm performance. It offers a variety of collaborative filtering algorithms, including matrix factorization-based methods, neighborhood-based approaches, and more. These algorithms are designed to predict user preferences or item ratings based on historical data, such as user-item interactions or explicit ratings. The Surprise library also provides functionalities for data preprocessing, cross-validation, hyperparameter tuning, and model evaluation. It supports common evaluation metrics like RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error) to assess the accuracy of recommendation models. The name "SurPRISE" stands for "Simple Python RecommendatIon System Engine."

Data Selection and Preprocessing:
I started by downloading the Amazon Movies and TV dataset. This dataset can in the form of a .json file. The entire file was around 5GB and thus was too large for Google Colab to handle. This is why I decided to only pick the first 500000 entries in the dataset. I read in the .json file using the pandas library in Python and created a dataframe. Then, I dropped the fields in the dataset that I didn't need so all the fields except for "overall", "reviewerID", and "asin" were dropped. Then I converted the .json file to .csv. Then loaded the dataset into Surprise from a dataframe using the Dataset class. The Reader class was also used. This class is used to parse a file containing ratings.

Train and Test Data:
Using the train_test_split from Surprise, split the dataset into a training set and a testing set. This split was 80% training and 20% testing.

Model Selection and Rating Predictions:
I chose 4 different models that could predict rating for the test dataset based on the training done with the training dataset. The models I chose were the following:

- BaselineOnly: BaselineOnly is a basic recommender model in Surprise that predicts ratings based on a baseline estimate, which considers the average rating for each user and item. It aims to capture the overall trends and biases in the data and provides a simple yet effective approach for recommendation.
- Matrix Factorization: Matrix Factorization, specifically Singular Value Decomposition (SVD), is a popular collaborative filtering technique used in recommender systems. It decomposes the user-item rating matrix into lower-dimensional representations, which capture latent factors or features. SVD-based models aim to reconstruct the original matrix by estimating missing ratings and predicting personalized recommendations.
- Item-based CF: Item-based Collaborative Filtering, specifically KNNwithMeans, is a neighborhood-based approach in recommender systems. It identifies similar items based on user ratings and predicts ratings for a target item by considering the ratings of its neighboring items. KNNwithMeans takes into account the mean ratings of users and adjusts predictions accordingly. It leverages item-item similarity to provide personalized recommendations based on user-item interactions.
- Slope One: Slope One is a memory-based collaborative filtering algorithm implemented in Surprise. It calculates the difference in ratings between items and uses this information to make predictions. Slope One is known for its simplicity and efficiency,

making it suitable for large datasets. It does not require any matrix factorization or extensive computation, making it a practical choice for certain scenarios.

Each of the models chosen has a different technique and approach for recommendation. For each of these models, I calculated the MAE and the RSME using the accuracy module of Surprise. These both are evaluation metrics used to measure the accuracy of predictions.

- Mean Absolute Error (MAE): MAE measures the average absolute difference between predicted and actual values. It calculates the average of the absolute differences between each predicted value and its corresponding actual value. MAE provides a measure of the average magnitude of errors without considering their direction.
- Root Mean Squared Error (RSME): RMSE is a variation of MAE that emphasizes larger errors. It calculates the square root of the average of the squared differences between predicted and actual values. RMSE provides a measure of the typical deviation of errors.

Ranking:

After this, I generated a recommendation list consisting of the top 10 items for each user in the test set. For this, I used ranking function from the Surprise documentation, the reference for which is given in the Google Colab notebook and at the end of this PDF.

Then, I evaluated the quality of the recommendation list using precision, recall, and f-measure.

- Precision: Precision is the proportion of correctly predicted positive instances out of all instances predicted as positive. It measures the accuracy of positive predictions and represents how well the model avoids false positives. Higher precision indicates fewer false positives and better precision.
- Recall: Recall is the proportion of correctly predicted positive instances out of all actual positive instances. It measures the ability of the model to identify all positive instances and represents how well the model avoids false negatives. Higher recall indicates fewer false negatives and better recall.
- F-measure: F-measure is a metric that combines precision and recall into a single value to provide a balanced evaluation of a model. It calculates the harmonic mean of precision and recall, giving equal importance to both metrics. The F-measure reaches its best value at 1 (perfect precision and recall) and worst at 0.

Results:

The results of all the models are summarized in the table below.

Model	MAE	RSME	Precision	Recall	F-measure
BaselineOnly	0.74577	1.0365	0.77858	0.77807	0.77832
Matrix Factorization	0.74189	1.04152	0.77139	0.76862	0.77001
Item-based CF	0.7906	1.09626	0.82424	0.82276	0.8235
Slope One	0.79687	1.12869	0.81274	0.81155	0.81214

From these results it can be seen that the MAE values and RSME values are pretty similar for BaselineOnly model and Matrix Factorization (SVD) model. The Item-based CF and SlopeOne had very similar MAE and RSME values. This indicates that the performance of BaselineOnly model and Matrix Factorization models was better.

Precision, recall, and f-measure values were higher for Item-based CF and SlopeOne models than they were for the BaselineOnly and Matrix Factorization models signifying that they had better accuracy than the other two models.

Github:
https://github.com/mluktuke/CSE-272_HW-2_Recommendation-System

References:
https://surprise.readthedocs.io/en/stable/FAQ.html