DSCI 351 Recommender Systems
Final Project: Amazon Recommendation System
Team 3: Yara Musad, Jason Le, Kelsey Chong

1. **Background of the field your system would make recommendations.**

    A. Background:

Online shopping has become an integral part of our lives, offering a convenient alternative to traditional brick-and-mortar stores. Amazon, a leading e-commerce platform, provides a wide variety of products, including electronics. However, the abundance of choices can often be overwhelming for consumers. That's where recommendation systems come into play. They filter product options and personalize the shopping experience based on individual customer behavior, thereby helping both consumers in decision-making and businesses in enhancing customer engagement.

    B. Problem Statement:

While there are existing recommendation systems in place, many of them operate either on a purely content-based or collaborative filtering basis. Each approach has its own set of limitations—content-based methods often suffer from the 'filter bubble' phenomenon, while collaborative filtering methods struggle with the 'cold start' problem. Additionally, the vast and sparse datasets generated by Amazon's extensive user and product base present challenges that are computational as well as algorithmic.

    C. Goal:

Conduct exploratory data analysis (EDA) to understand the characteristics of Amazon's electronic product reviews and ratings. Implement a model-based collaborative filtering recommendation system using machine learning algorithms. Evaluate the model's performance using metrics like RMSE (Root Mean Square Error) and MAE (Mean Absolute Error). Provide insights into how this model can be deployed and scaled effectively to improve user experience and engagement.

    D. Methodology:

To accomplish the objectives, this project will leverage Python's data science ecosystem, including libraries like Pandas for data manipulation, Matplotlib and Seaborn for data visualization, and Scikit-Surprise for building the recommendation algorithm. We will leverage User-based, Item-based and Model-based Collaborative Filtering methods.

The dataset consists of Amazon electronic product reviews, capturing various attributes such as user ID, product ID, ratings, and timestamps.

2. **Provide a description of the dataset used.**
This dataset was obtained from Kaggle, in particular, it contains data from Amazon products.

The dataset contains electronic product reviews on amazon.
Source - Amazon Reviews data (http://jmcauley.ucsd.edu/data/amazon/)
Attribute Information:
userId : Every user identified with a unique id (First Column)
productId : Every product identified with a unique id(Second Column)
Rating : Rating of the corresponding product by the corresponding user(Third Column)
timestamp : Time of the rating (Fourth Column)

3. **What is/are the method(s)/model(s) utilized in your system and why?**
   Above, we mentioned the methodology we will be using.

   We have decided to use the following approach:

   Collaborative Filtering
   In general, Collaborative filtering (CF) is more commonly used than content-based systems because it usually gives better results and is relatively easy to understand (from an overall implementation perspective). CF is based on the idea that the best recommendations come from people who have similar tastes. In other words, it uses historical item ratings of like-minded people to predict how someone would rate an item. Collaborative filtering recommends the set of items based on what is called the user-item interaction matrix.
   In particular, we have adopted user-based CF, item-based CF, model-based CF.

   First, we filtered the rating matrix, since there are vast amount of products and many users don't rate enough, we can only retain 515 users out of 167080 with a filter of 0.05% total products rated. We then get user mean and product mean.

   For user-based CF, we first calculate the rating matrix adjusted by taking the filtered matrix and subtracting the user mean as a way to normalise the data. Then we calculated the similarity score based on Pearson correlation of each other users to our targeted user. After that, we returned the list of all the similarity scores. We created predictions of a targeted user's rating on a targeted product by calculating ratings of such product from users who are most similar to the targeted user.

   For item-based CF, we in return found out products that are most similar to the targeted products, then predicted the targeted products' ratings based on a particular user's rating of similar products.

   Model-Based
   These methods are based on machine learning and data mining techniques. The goal is to train models to be able to make predictions. For example, we could use existing user-item interactions to train a model to predict the top-5 items that a user might like the most. One advantage of these methods is that they are able to recommend a larger number of items to a larger number of users, compared to other methods like memory based approach. They have large coverage, even when working with large sparse matrices.

4. **How to interpret/explain your system's recommended results (and/or the comparison of different results from different methods)?**

User-Based CF: The similarity scores identify users who are most similar to the target user ("A105TOJ6LTVMBG"). Users with higher similarity scores are considered more similar in their rating behaviors.

The user's ratings distribution suggests that most of their ratings are close to their average rating, indicating that they might not have rated item 3 yet.

The prediction result (-0.008691551271196928) suggests that, based on the ratings of similar users, the target user is expected to give item 3 a rating close to their average rating. However, the prediction might be less confident since only 17 similar users were considered.

These results show the inner workings of the user-based collaborative filtering (CF) method and, more importantly, underscore its capacity to produce highly personalized recommendations for the target user ('A105TOJ6LTVMBG'). User-based collaborative filtering is inherently designed to tailor its recommendations to individual users by leveraging the collective wisdom of like-minded users within the dataset

Item-Based CF: The similarity calculation identifies products that are highly similar to the target product based on user ratings. In this case, the system has found products with a high degree of similarity to "1400501776."

Mean-centering user ratings ensures that the predictions are relative to the user's rating tendencies. This step is crucial for capturing individual user preferences accurately.

The prediction process leverages the similarity scores and adjusted ratings to estimate how the target user would rate the target product. The prediction result (-0.00944835025852769) suggests that, based on the behavior of 2 similar products and the user's ratings, the system expects the user to rate "1400501776" with a rating close to their own average rating.

Item-based collaborative filtering excels in providing personalized recommendations based on the similarities between products and is a valuable technique for enhancing user engagement and satisfaction.

Model-Based CF: We leveraged both Popularity Measure & SVD in surprise techniques when applying model-based CF.

For popularity measure, the CF methoth employing Truncated SVD and correlation analysis, has generated personalized recommendations for a specific customer based on their purchase history of the product "B00000K135."

We initiated the recommendation process by decomposing the original user-product rating matrix using Truncated SVD. This mathematical technique allowed us to reduce the dimensionality of the utility matrix, effectively capturing latent features that characterize both products and users.

Following matrix decomposition, we constructed a correlation matrix. Each entry in this matrix represents the correlation between pairs of products, utilizing the latent features

obtained during decomposition. In essence, it quantifies the similarity or dissimilarity between products based on user preferences.

Product Selection: We specifically selected the product "B00000K135" as the customer's starting point for recommendations. This product represents the customer's historical purchase, providing a valuable reference for tailoring suggestions.

We calculated the correlation scores between "B00000K135" and all other products in our dataset. These scores reflect how closely related each product is to the one the customer has already bought. Higher correlation scores signify greater similarity in user preference.

To create recommendations, we applied a correlation threshold (0.65) to identify products with substantial similarity to the customer's purchased product. These similar products are considered candidates for recommendation. Additionally, we took care to exclude products the customer has already purchased to ensure fresh recommendations.

Finally, we present the top 10 recommended products for the customer. These recommendations are derived from products that share significant correlations with "B00000K135" in terms of user behavior. By offering these tailored suggestions, we aim to enhance the customer's shopping experience and encourage further engagement with our platform.

As for the **SVD in surprise**, we started by running a 5-fold cross-validation to obtain the RMSE. Then we loaded our data and used to it train the SVD model and make predictions.

To illustrate how the model generates recommendations, we performed a specific prediction for a user with the userId 'A17HMM1M7T9PJ1' and the productId '0970407998.' The model predicts a rating of approximately 4.15 for this user on the specified product.

The final output of our model is user-centric recommendations. By analyzing user-item interactions and latent factors derived from SVD, the model identifies products that align with a user's preferences. These recommendations are tailored to individual users, aiming to enhance their shopping experience.

5. **How to evaluate your system's performance?**

For model-based, we conducted 5-fold cross-validation to assess the performance of the SVD model.

The results indicate two key evaluation metrics: Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). These metrics help gauge the accuracy and precision of our recommendation system.

RMSE: RMSE measures the average prediction error of the model, with lower values indicating better accuracy. In our case, the average RMSE across all folds is approximately 1.2694.

MAE: MAE represents the mean absolute difference between predicted and actual ratings. A lower MAE signifies higher precision. Our model achieves an average MAE of around 0.9938.

The SVD-based model delivers recommendations with pretty good accuracy and precision, as demonstrated by the RMSE and MAE scores. It efficiently leverages matrix factorization to capture underlying user-product interactions. The model's predictions empower us to offer personalized recommendations, exemplified by the specific prediction for 'A17HMM1M7T9PJ1' and '0970407998.' These how we could derive our system's performance to improve user engagement and satisfaction through a data-driven recommendation system.

6. **<u>What is the limitation and future improvement of your system?</u>**
   A. Limitations:
      The scope of this project is limited to Amazon's electronic products category. Although the model can be generalized to other categories, the algorithms and parameters may need to be fine-tuned accordingly. The project will also address the computational challenges posed by large and sparse datasets.

   B. Future Work:
      In general, content-based focuses on the attributes of items and provides you with recommendations based on the similarities between them. We can use this approach to find similarities between each type of product item and rank the similarity score from highest to lowest and select product sets based on the number of recommendations we want to offer.

      However, our datastet only contains information about the ratings, users, and products, without additional features about the products. A possible workaorund is to assume that a high rating signifies likeness for a certain type of product and aply the approach based on user preferrences.

      Limitations: The recommendation will be limited to what users liked, watched, interacted with before. It doesn't give users a chance to explore a new area they've never been to before. Also, all users who like item X will receive the same recommendation set.

      Content Based Filtering
      While the present study concentrates on model-based collaborative filtering, future research could expand into the realm of content-based filtering. This technique places emphasis on the attributes of items, generating recommendations based on the similarity metrics between them. Such an approach would involve ranking items according to their similarity scores, from which a predetermined number of top-rated recommendations could be made.

      However, our existing dataset predominantly contains user IDs, product IDs, and their corresponding ratings. It lacks additional attributes or features that could be employed

in content-based filtering. One potential workaround for this limitation is to leverage high ratings as a proxy for product preference, thereby applying the content-based methodology based on inferred user preferences.

Limitations: This approach would inherently be constrained by a user's prior interactions and ratings. It wouldn't offer users opportunities to discover new domains or product categories. Moreover, the recommendations would lack diversification as all users who have rated item 'X' highly would receive identical suggestions.

Hybrid Recommendation Systems

In the future, the hybrid recommendation system could be extended to incorporate content-based filtering techniques. Content-based filtering emphasizes item attributes and leverages similarity metrics to make recommendations. However, this approach necessitates additional item features beyond user ratings, which are currently the primary data source. One potential approach to address this limitation is to infer user preferences based on high ratings and apply content-based filtering accordingly.

Limitations:
This extension would still be contingent on users' past interactions and preferences, limiting its ability to introduce users to entirely new product domains. It might result in recommending similar products to users who have rated a particular item highly, potentially lacking diversification in recommendations.

Knowledge Based Recommendation Systems

Future endeavors in knowledge-based recommendation systems could involve augmenting the knowledge base with additional product attributes, specifications, and domain-specific information. This expansion would enable the system to make more informed recommendations across a broader spectrum of product categories. Additionally, integrating natural language processing (NLP) techniques to analyze unstructured data, such as product reviews and descriptions, could enhance the system's knowledge-driven capabilities.

Limitations:
Despite its potential for knowledge-driven recommendations, this approach is bound by the availability and quality of structured data. The system may encounter challenges when dealing with niche or less-documented product categories, where structured information is limited. Furthermore, the system may not capture nuanced user preferences that could be discerned from unstructured textual data.

7. **References**

Source - Amazon Reviews data (http://jmcauley.ucsd.edu/data/amazon/)
Sources:
https://www.kaggle.com/code/farizhaykal/recommendation-system-for-amazon-products

https://www.kaggle.com/code/saurav9786/recommender-system-using-amazon-reviews/notebook