Everett Johnson

Recommender System

January 6, 2022

Background

Recommender systems have become increasingly popular over the last decade or so. More and more companies are starting to utilize data to make recommendations to customers to ultimately increase their bottom line. Recommender systems can also help improve the customer experience by making buying recommendations, either based on previous purchases or purchases made by people with similar demographic characteristics. With these types of systems, there are a number of considerations. One being the data available for the type of system being used. Is the data readily available from internal or external sources? Is the data accurate? All questions that must be answered prior to implementation. Another consideration is the type of system being used. Will the system use content from related books sources? Or will the data be based on users previous reading history or perhaps ratings of other books? Depending on the type of system and the data available will determine the best approach for which type of system to use. Collaborative filters use different data and models to perform their functions versus a Content based filter. This whitepaper is based on a collaborative filter with generic ratings data. The data has not been vetted for accuracy and contains anonymized data.

Business Problem

The business problem that can be solved by this type of system is that of making recommendations to customers who shop online for items. As opposed to providing generic recommendations, the recommendation system has the ability to personalize recommendations that more accurately align with the shoppers tastes and preferences. This type of approach can help improve upsells and overall revenue. Amazon is a good example of how these systems could be implemented. Amazon uses previous purchase history as well as search history to provide recommendations to customers for future purchases. It has the ability to recommend products that perhaps customers didn’t know existed.

Approach

For this model, I used an item to item, collaborative filter to see how a user may rate a certain book. This system takes the user’s ratings of a certain book and based their ratings, predicts how another reader may rate that particular book, there by making a recommendation for that user. The first step in this process is to find quality data that can be used for this application. I used data found at the following location on Kaggle, https://www.kaggle.com/arashnic/book-recommendation-dataset. Three files were used: one file with user information, one file with ratings information by user, and one file with book information and descriptions.

Cleaning/EDA

All three files were used but the primary file used for this application was the ratings file, Ratings1.csv. The ratings file has 3 columns, the User ID, the ISBN, and the Rating. It also has 15185 rows of data. The first step was to clean the data and format it into a pandas dataframe. This is for readability and to start the cleaning process. After putting the files into a pandas df, I explored how much data may be missing or inaccurate by using the .isna method. As with most data being used in recommender systems, not all users will rate all products or services being used. This is to be expected and accounted for. In my process, I accounted for this by replacing the NA values with 0’s and then later on replacing it with the mean values of all ratings. This will show in the “user\_ratings\_table\_normal” dataframe. There are other approaches to handling this type of problem, but I have found this approach may work best for this type of system here. I calculated the rankings for the ratings and showed that 20% of the books have higher than an 8 rating.

Model/Method

The model that I used for this system is the K-Nearest Neighbors model in SciKit-Learn. This model allows us to find the users that are most similar (or closest) to our target user and average how they rated the book or ISBN that is presented. This can give us a reference of how a reader may rate a book, even though they may not have rated it before.

Conclusions

For the example input, I used user “276744” and the ISBN “3442413508” which is a crime novel, “Well-Schooled in Murder” written by Elizabeth George. The model scored the likely rating as a zero, which could indicate that the user does not read or like crime novels. Another run in the code with user 276747 and ISBN 899197698 (a children’s book – 5 Little Monkey’s Jumping on a Bed”) gave a 0 rating as well. Because of the repeating of the 0 prediction, I tweaked the n\_neighbors parameter in the model and went from 5 to 1000. Because of the sparse matrix that comes with this type of data, I will look to more “neighbors” to see if there is a different prediction that may occur. With this tweak, the prediction goes to .0053 for the book in question. From here, more testing on different datasets may be needed to see how well the model is prediction and to ensure it is not overfitting.

Challenges

One of the major challenges for this is the number of ratings that is being provided. I accounted for having many books that are not rated but the number of actual ratings proved challenging. I used the average of the ratings to normalize the data but there may be other methods to explore in order to be able to improve the results.

Assumption

The major assumption for this type of system is that the user preferences and ratings would stay the same through time. This may not always be accurate. Tastes may evolve and people may explore new types of genres for reading material. This must be accounted for when considering recommendation systems.

Limitations

The limitation for this system is primarily the data. For more testing of results, similar data must be obtained to test outcomes. Testing data on another group of individuals with a same rating system could be the next steps in the process to test whether these results are accurate and valid.

Implementation Plan

After further testing and validation, this type of system could be implemented on a bookseller’s website. This would be part of the field of MLOps, but would include building a training pipeline for the data, having a storage facility for the data, such as a SQL database or other type of data warehouse, and have a constant flow of data that is monitored closely for accuracy.

Ethical Considerations

When dealing with personal data, ethical considerations are always present. In this situation, the data used did not include personally identifiable information. If this model was deployed in real time, this more than likely would not be the case. In working with data, the company or team must be very careful in how the data is used. The person must be aware that the data is being collected and should be advised of how it is being used. The company or team could provide a consent for opt in as well to ensure the customer is aware of this process. The data should not be sold or otherwise used without first gaining consent from the user.

References

BANIK, R. O. U. N. A. K. (2018). *Hands-on recommendation systems with Python: Start building powerful and personalized recommendation engines with python*. PACKT Publishing Limited.

*Deploying a recommender system in production*. Modulai. (2021, August 31). Retrieved January 7, 2022, from https://modulai.io/mlops-at-ahlsell-deploying-a-recommender-system-in-a-production-environment/

*Overview*. Recommendation Systems :: General Collaborative Filtering Algorithm Ideas. (n.d.). Retrieved January 7, 2022, from https://www.cs.carleton.edu/cs\_comps/0607/recommend/recommender/collaborativefiltering.html