Capstone 3. Product Recommender for online stores - Problem Statement -Shailaja

<table>
<tr><td colspan="2">Problem Statement Worksheet:
How online stores can help customers to buy the right product from their millions of products,so that online retailers can retain the customer from going elsewhere and see yearly increase in their average order value. Build a recommendation system so that customers can get the right product with personalized info.</td></tr>
<tr><td>Context:
Online stores show millions of products to the customer from their catalog. Choosing the correct product for their needs is becoming difficult because of so much information.Since customers are more likely to buy based on personalized recommendations, management has decided to go for a recommendation system so that they can retain the customer and hence increase yearly product sales.</td><td>Constrain within solution space:

Dataset has 7 M(7824481) rows and 9 columns,size of 1MB.
May need to limit 2 necessary 3 or 4 columns like,
userId : Every user identified with a unique id
productId : Every product identified with a unique id
Rating : Rating of the corresponding product by the corresponding user
timestamp : Time of the rating (ignore this column for this exercise)</td></tr>
<tr><td>Criteria for success:
In this project build a recommendation model for the electronics products of Amazon, based on the ratings.

Explore and visualize the dataset. Understand what customers are buying, and recommend at least 3 to 5 products based on the ratings.</td><td>Stakeholders to provide key insight:
-Business Leaders,
-Data science and Machine learning team</td></tr>
<tr><td>Scope of solution space:
Use machine learning and data mining techniques to bring Model-based collaborative filtering systems.

Source has the two different datasource 1.Ratings data 2. Review data. Try making use of review data by joining with reviewerid</td><td>Key data sources:
The dataset here is taken from the below website.

Source - Amazon Reviews data (http://jmcauley.ucsd.edu/data/amazon/links.html) The repository has several datasets. For this case study, I am using the Electronics dataset or Clothing, Shoes and Jewelry</td></tr>
</table>

Sample review dataset:
This dataset includes reviews (ratings, text, helpfulness votes), product metadata (descriptions, category information, price, brand, and image features), and links (also viewed/also bought graphs). And has 1689188 records.

"reviewerID": "A2SUAM1J3GNN3B",
"asin": "0000013714",
"reviewerName": "J. McDonald",
"helpful": [2, 3],
"reviewText": "I bought this for my husband who plays the piano. He is having a wonderful time playing these old hymns. The music is at times hard to read because we think the book was published for singing from more than playing from. Great purchase though!",
"overall": 5.0,
"summary": "Heavenly Highway Hymns",
"unixReviewTime": 1252800000,
"reviewTime": "09 13, 2009"

Deliverables

A GitHub repo containing the work you complete for each step of the project, including:

- A slide deck
- A project report

For My capstone 3 I will be using the Electronics dataset and with 4 columns.
"reviewerID": "A2SUAM1J3GNN3B",
"asin": "0000013714",
"helpful": [2, 3],
"overall": 5.0.

Model Selection :-

Surprise Package :http://surpriselib.com/

References

Documentation :https://surprise.readthedocs.io/en/latest/

Installation: http://surpriselib.com/
Git hub : https://github.com/NicolasHug/Surprise

There are a lot of different packages available to build a recommender system. For this one, I'm using the Surprise package. Surprise has many different algorithms built in.Itprovides

various ready-to-use prediction algorithms such as baseline algorithms, neighborhood methods, matrix factorization-based (SVD, PMF, SVD++, NMF), and many others. Also, various similarity measures (cosine, MSD, pearson…) are built-in.

In this case, I need to load in a custom dataset to use with Surprise. According to the documentation, we need to make sure our data frame has three columns: the user ids, the item ids, and the ratings. Additionally, we'll need to specify the rating scale. In our case, users has used the ratings discretely from 1 to 5.

I'm also going to split the data into training and testing data using the Surprise package

With the Surprise library, I use below algorithms

BaselineOnly: Algorithm predicting the baseline estimate for a given user and item.

KNNBaseline:A basic collaborative filtering algorithm taking into account a *baseline* rating.
NMF:A collaborative filtering algorithm based on Non-negative Matrix Factorization.
Co-clustering:A collaborative filtering algorithm based on co-clustering.
SVD:When baselines are not used, this is equivalent to Probabilistic Matrix Factorization, it is as popularized by Simon Funk during the Netflix Prize