RECOMMENDATION ENGINE PROPOSAL

-Aashesh Vontivillu Nareshchand

Objective: This recommendation engine provides E-commerce product recommendations based on the following approaches: -

1) **Content-based recommendation**: Finding products with similar product desciptions. This approach can be targeted for new customers or customers with low engagement (Some/No product rating provided).
2) **Model-based recommendation**: Finding similar products using item-item collaborative filtering. This approach can be targeted to recurring or loyal customers (High Product engagement/ Rating).

Dataset Source: Amazon product data (http://jmcauley.ucsd.edu/data/amazon/links.html)

Dataset 1: Product metadata files for categories – Beauty, Grocery and Gourmet Food, Baby, Musical Instruments, Patio-Lawn and Garden were chosen. These files were appended and used for content-based recommendation

Dataset 2: User Ratings/Reviews files for categories – Beauty, Grocery and Gourmet Food, Baby, Musical Instruments, Patio-Lawn and Garden were chosen. These files were appended and used for collaborative filtering

Note: Due to computational constraints, a sample of 10% data was used in both the approaches.

Machine Learning Algorithms Deployed:

1) **Neural Network Embedding Model**:
    - The goal of this model is to create 'Word Embedding'. In simple terms, words embedding are real-valued vector representations of a word. These embeddings are created by training product desciptions with their respective product categories.
    - The trained Word Embedding will have words represented as numerical vectors in a n-dimensional space such that, words closer to each other will be similar with respect to the product category.
    - In other words, cluster-like formation of words is generated with each cluster pertaining to a specific product category. Example: Words cocoa, truffle, pasta, will be closer to each other as they belong to the grocery category.
    - These trained Word Embedding were used to provide recommendations by finding products that have similar product descriptions.
2) **K-Nearest Neighbors**:
    - We attempt to understand the underlying distribution of the users and their item ratings
    - Product recommendations are provided by calculating the distance between the given product data point and the other data points.
    - The top N neighbors of the given product are listed as its recommended products, thereby achieving item-item collaborative filtering - "*Users who liked this item also liked*"

Advantages of Content-based recommendation

- **Enhancement of the existing model**: This approach can be appended with the existing algorithm used to provide recommendations in a website. It provides an additional similarity measure that calculates/tracks similarities between products using its product descriptions.

- **Search Engine Optimisation:** The trained Word Embedding can be utilised in search engine algorithms improving customer engagement to find relevant products even when the query given by the customer is not recognised and providing similar products that are relevant to the searched query. "*Misspelt product names are a source of amusement for your team, but they may also be a lost business opportunity*"

Trade Offs:

Content-based filtering:

- Given product descriptions are used to find similar products, it can be computationally expensive with the corpus of documents or the vocabulary of words is too large, eventually making the training process time-consuming
- This approach can never be used when the product descriptions are missing or not available.

Collaborative filtering:

- **New user/item cold start problem**: Since this approach deals with using a combination of both user and item data to provide recommendations, it is not wise to use when dealing with new users or items
- **Sparse Matrix**: This approach's preemtive assumption is that users have rated a product. When there is no rating provided by the user or in other words, when there is no user activity to the product, the sparse matrix can be too large to be handled.

Model Validation during Training:

Content-based filtering:

- Since Word Embedding are created and used in this process, the performance of the model would highly depend on how well the trained word embeddings are able to predict product category given the product description.
- Measures like recall score or F1-score can be used to evaluate this prediction performance.
- However, the product recommendations provided needs to be manually evaluated by randomly trying out various possible test cases.

Collaborative filtering: Evaluation based on how well the model can predict the user's rating for a product using metrics like RSME (Root-mean-square-error) or MAE (Mean Absolute Error).

A/B Testing: Model Validation after deployment:

Pre-requisites to be met while A/B Testing:

- Performed on a random set of users - 50% of the users will have the recommendation engine deployed over a period and the rest 50% will not have this solution deployed
- Website Traffic – There must atleast be a minimum number of customers visiting the website
- Increase in conversion rate - Atleast X% over the baseline solution (current conversion rate)

Evaluation metrics while A/B Testing:

- The customer behavioral attributes like Click-Through rate (CTR), Conversion rates and the duration of time spent on certain targeted pages can be tracked using Google Analytics Tracking code to make comparisons between the two groups of users.
- This information will help us in determining how well were we able to achieve the goal of increasing user-engagement and sales conversion.
- ROI - Consider the expenses of including the recommendation engine on a website with 100% traffic and the recommendations are not well received, the ROI can be low in this scenerio.

Deployment options:

- Content-based recommendation can be deployed along with the existing model such that after a user interacts with a product, its corresponding product description can be fetched to find products with similar product descriptions.
- Results from the collaborative filtering model can be shown under the “Users who liked this item also liked” section of the application front-end.
- As mentioned earlier that these approaches are targeted to different sets of people, the business model logic should have this as a consideration factor.

References:

Self reference: https://github.com/aasheshvn/Machine-Learning/blob/master/Word%20Embeddings%20and%20Movie%20Recommendations/Word%20Embeddings%20and%20Movie%20Recommendation.ipynb

https://heartbeat.fritz.ai/recommender-systems-with-python-part-ii-collaborative-filtering-k-nearest-neighbors-algorithm-c8dcd5fd89b2