Enhancing the Recommendation Engine of a Hungarian Bookstore

Public Capstone Project Summary

Zsófia Rebeka Katona
Central European University
Business Analytics, Master of Science
2024 June

Introduction

This document serves as a 3-page public summary of my Capstone project, focusing on the enhancement of the partner company's book recommendation engine. My partner company, a Hungarian SME-sized bookstore, focuses on specialty books. Their current algorithm creates weekly recommendations for each of their customers. The goal of this project was to improve their current recommendation engine, using text analysis. My objectives also included experimenting with various recommendation engine techniques and to ensure that my code and project are fully reproducible. My overall goal was to increase engagement by predicting more relevant suggestions. The methodology of the project contained data cleaning, exploratory data analysis, implementation, testing, comparing different techniques and documentation. I used various tools throughout the project, including Oracle Client, GitBash, DBeaver, JupyterLab and Power BI.

The Project

Exploratory Data Analysis: The company's database contained several hundred data tables and stored procedures, which served as the foundation for the recommendation engine. I considered developing the current engine too time-consuming, therefore I focused on the data and rebuilt the engine.

Item-based recommendation engine: I applied the Jaccard Similarity Score and the Nearest Neighbors (K-NN) techniques in this engine. Jaccard similarity calculation is a method that measures the number of common attributes divided by total combined attributes. While calculating the Jaccard Similarity score of two books was successful, computing all books' Jaccard similarity

scores and visualizing was computationally too intensive for my local computer. Therefore, I opted to implement the Nearest Neighbors method, which finds the closest items to a given item by comparing their features. As the dataset containing all books was too large, I conducted an analysis on a subset of 500 samples and created a heat map visualization. This chart provided minimal additional understanding, but showed high variability among the books due to the exclusion of non-numeric columns, such as authors and book abstracts.

Overall, I identified significant data quality issues that impacted the effectiveness of the recommendation system. The techniques provided limited insights due to data and computational constraints.

Customer-based recommendation engine: Me and the company selected three target customers with clear genre tastes to test the results of the engine. I used sklearn's cosine similarity package which measures the similarity of two items by comparing the angles of their vector representations, where smaller angles indicate greater similarity. As calculating the customer similarity with the full dataset was computationally too intensive for my machine, I opted to decrease its dimensionality, but the engine failed to make relevant predictions. In conclusion, I was unable to compare customers and make meaningful recommendations with this approach.

Vector database recommendation engine: I used Chroma DB, an AI-native open-source embedding database, to find and recommend similar items based on the book abstracts. I tested two kinds of trial queries for two of the target customers. The results of the first query were mixed, covering a wide range of topics from the same genre. The second query contained more detailed information about the other target customer's interest. Consequently, the results were more specific, but occasionally differed within the same genre. For instance, the query with keywords, like 'botanics', 'geography', and 'North America' yielded a reading about desert plants in India. I also experimented with the pairwise comparisons method, which compares each book with every other book to determine how different or similar they are to each other.

Overall, the vector database delivered both relevant and less relevant recommendations. More detailed queries and a complete dataset is likely to provide more accurate recommendations, however the computational challenges prevented full calculations in this project.

The company requested a separate analysis for a specific genre that they believe has significant market potential. I applied the same techniques and followed the same data cleaning steps as in the previous analyses and similarly, ChromaDB produced the most accurate results.

Benefits for the Company

Despite not building a complete recommendation algorithm, the partner was able to see a wide range of recommendation techniques that can be implemented in their current engine. They also have access to a private GitHub repository where all the codes, usage instructions and requirements can be found. Throughout the project, the importance of data entry and data processing also became evident, as a lot of observations were lost due to missing values. The project also showed the significance of a strong infrastructure as my local machine was unable to handle most data tables due to their sizes.

Lessons Learnt

Overall, I encountered several limitations, including challenges with unique keywords, sparse and large datasets, and inaccurate data entries affecting results. Moreover, I had to find alternative ways of implementing different techniques and aligning them with the data. The biggest limitations came from memory issues due to large datasets. This case highlighted the importance of proper data cleaning and exploratory data analysis. Furthermore, I gained an insight into various recommendation techniques and the iterative process of improvement based on feedback. In terms of project management, sharing partial results and initial code blocks earlier with the company could have resulted in more guidance throughout the case. This delay caused some miscommunication which could have been prevented with regular updates.