Take-home exam (Jan 21-22)

Kevin Jacobs 3IMS

Contents

Intro

For our take-home exam, we had to write an algorithm that can recommend to customers who don't know what to buy for someone. We have to find our dataset ourselves and decide for ourselves which algorithm we are going to use. The algorithm I chose was K Nearest Neighbor..

The Algorithm

The algorithm I chose is K nearest neighbour(KNN), which was also on my mind before I started doing my research. The reason I chose this algorithm is simple. KNN looks for a result that comes closest to the input data. When you have multiple points on a graph and select a random point on that graph knn will return me the closest point it can find. That seemed interesting to me, and I've read several sources about it that say this is a good and possible solution. On the site madasmy.com, they write about several possible filters. They write about two different methods, Nearest Neighbour and Matrix factorization for Collaborative filtering.

K Nearest neighbour

User based collaborative filtering

KNN has two types of filters a “user-based” and “Item-based”. The “users based” finds the users who have the same taste in products as the current user. You can find a similarity based on what products they buy. Unfortunately, my dataset had no user data. Meaning I had to go for option two.

Item based collaborative filtering

Item-based filtering gives you products that are similar to what users previously bought. That sounded much better for me than user-based.

Matrix Factorization (MF)

Matrix Factorization was also an AI that I was eager to try and test out, but some issues kept me from getting started. For instance, I have no idea how this algorithm works, and I've never worked with UWP. I did, however, do some research about Matrix Factorization. A different problem is that MF is not a recommendation algorithm, but more of a review predictor. Not 100% the same, but close. This algorithm is used in Netflix and Amazon, among others.

Programming

Spelling

There were some difficulties I encountered with Verb/Noun/Subj. I thought long and hard about this and put too much time into it. My Dyslexia didn't help me with this. Fortunately, you can find list of Verb/noun/subj that other people have made for you and, they are free to use.

I use different word lists with thousands of words for each grammar notation. I put these in a separate variable for later use. The noun was a bit more difficult. You have so many nouns that it is almost impossible to list them here. Fortunately, I came up with an idea. A list of Hobbies from various sources. I added thousands of hobbies in a file, and now my program can break my grammar down 80% of the time.

Point system

I have worked with both KNN and a Point system. Meaning that when you enter something like “Girl likes books” it will search for everything that contains the word book in the name of the product as for the category. Based on these results, they already receive points. So here you get a list that always gives something relevant. I got this idea from a Youtuber CodeBullet seeing him always use a points system I thought why not use a similar method that he does? I realize that almost every algorithm has a point system (KNN tries to get the lowest points). But the point system I made is heavily focused on the points system.

Ethical problem

My algorithm is far from perfect. I am not happy with how it works at the moment. I can certainly mention a few ethical problems. One problem is that there is no age filter, meaning if you search for “Boy likes video games” you can get 18+ games or maybe show something worse. Fortunately, almost everything in my dataset is PG+16 (not 100% sure as there are 10,000 options). I have found a small solution for this, but this only applies to babies. I hardcoded this solution. I put "baby" in the nouns list if you say "baby likes trains".

End note

I have learned a lot from this project. I had a lot of fun programming this. The only thing I regret is not starting sooner. I know I can do this and, I was a bit disappointed with my results. Even though it works, my dataset is not built for KNN and, that resulted in me using the point system.

sources

info about datasources: https://cseweb.ucsd.edu/~jmcauley/datasets.html

Algorithmes: https://madasamy.medium.com/introduction-to-recommendation-systems-and-how-to-design-recommendation-system-that-resembling-the-9ac167e30e95, https://en.wikipedia.org/wiki/Recommender_system, https://xamlbrewer.wordpress.com/2019/04/05/machine-learning-with-ml-net-in-uwp-recommendation/,

API: https://elfsight.com/blog/2020/05/how-to-use-amazon-ecommerce-api-examples-and-pricing/,

Datasets: https://data.world/, https://en.wikipedia.org/wiki/List_of_hobbies,

Inspiration: https://www.youtube.com/c/CodeBullet,