**Thoughts on Comprehensive Guide to build a Recommendation Engine from scratch (in Python)**

Helping people to make decisions. The article first mentions how many choices today’s consumer is faced with, and how people try to come to decisions.

In our lifetime that’s certainly something that has grown and my husband and I often even discuss our indecision when faced with so many more choices than we historically have. It’s a joke between us about “what to have for dinner” (for a laugh I recommend if you haven’t already to listen to the Weird Al song “Stuck in the Drivethrough” though it’s a bit long it’s very relatable.

Therefore the goal of Recommendation Engines is to essentially learn what a person likes and make recommendations based on the history of what they have liked to recommend other things they may like to consumer (whether it be shows, food, books etc). They are now widely used in many day to day shopping or searching processes.

First step is to decide the problem you are wanting to work (their practice example is jokes so we will run with that). Initially when a user is new you won’t have any historical data to utilize to recommend so often the first step appears to be actually utilizing a popularity recommender to new users to build enough historical data on the user to begin instead using recommendation engines.

From there we need to decide what factors (features) will be most relevant to choosing well for the user – so for the example of a joke, does the person like more jokes about animals, people, work situations, personal situations etc (genre). What type of jokes next – silly, dry humour, dad jokes, crude jokes, pg 13? That could be another feature. How do we determine if they like or dislike a joke? How long they spend on the page for each joke perhaps or if there’s a rating, we would need to decide these factors to be able to establish the matrix to use for recommendations also.

Storage – this is something we haven’t really talked about in class much yet other than in a processing power way, however if we are building a dataset to work from this would be important to consider,and plan.

They go on to here regarding the math behind the scenes which I’m starting to get a grasp on in general but isn’t my personal strong suit, so I’ll summarize this a bit in that it’s how we are measuring what to recommend, such as distance or error or accuracy which each have their own calculations, to therefore recommend well.

When they describe the collaborative filtering, such as user-user collaborative filtering it’s almost like finding friends who like similar things, only within the dataset, and therefore recommending possible similar likes. So the predictions of what someone would like based on what other similar users like.

They go on to do a case studie on the MovieLens Dataset which I believe is actually the same one we use for our recommender editors and therefore feels fairly familiar, and to introduce matrix factorization which we just learned and makes sense overall to me so far. They also go on to discuss the different metrics that might be used or what else could be tried. It was very thoughtful and thought provoking.