Clothes Item-item based Recommendation system

By Kal Lemma and Georgiy Treyvus

Questions and Approach

Clothing Fit Dataset:

1. Started with a Customers Fit and Rating, a data-set that was collected from “ModCloth” and “RentTheRunWay” provided on Kaggle.
2. Dataset includes User’s measurements, items purchased, and ratings for purchased items.
3. Items did not include names, hence basing prediction for item ids.
4. Dataset is highly sparse, with most products and customers having only a single transaction.

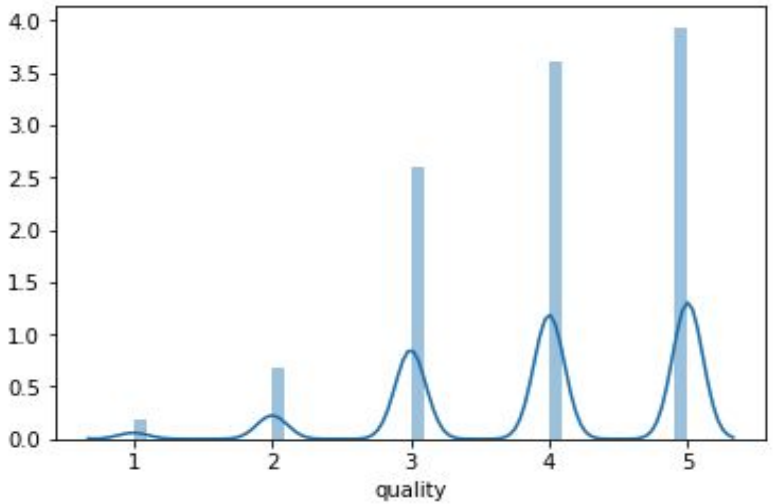
https://www.kaggle.com/rmisra/clothing-fit-dataset-for-size-recommendation

Exploratory Data Analysis

1. Dealing with highly sparse data originally, but primarily looking at at main three factors of usernames, item ids, and ratings
    a. Dropped everything else
2. Only had to take out 68 NaNs from our ratings column
3. Number of customers: 32,399 Number of products: 1,376 Number of transactions: 82,790
4. No need for Feature Engineering nor Selection

Better View of the Dataset

- Average amount of times an item is bought is around 60 times.
- Average amount of Items an individual bought was close to 3 items.
- Most customers seem to be happier with their purchases than being disappointed.



Recommendation System Plan

1. Memory-Based or Model-Based?
    a. Best Choice was a Model-Based Collaborative Filtering method using Surprise.
2. **Memory-Based**; Can Use multiple different similarity metrics to find out out which performs best: Pearson, Cosine, Jaccard.
3. **Model-Based**; Using Singular Value Decomposition (SVD) to decrease the dimensions of our utility matrix and extract latent factors.
    a. SVD essentially turning our Recommendation problem into an Optimization one.
4. Root Mean Square Error (**RMSE**) is our metric for performance.
5. Using Model-Based (Matrix factorization) rather than Memory-Based collaborative filtering to make faster predictions with less data than the original.

Attempting Memory-Based Models

- Memory-Based: Cosine similarity outperformed the Pearson method
  - Best Model performance using Neighbor-Based methods was KNN_Baseline with Cosine similarity

	RMSE	
KNN Basic Cosine	1.0351	
KNN Basic Pearson	1.0530	
KNN Means with Cosine	1.0017	
KNN Baseline with Cosine	0.9906	#WINNER#

Model-Based Approach with SVD

- GridSearch to find best parameters for SVD, changing our n_factors (10, 20, 30, 50, 100) and reg_all (0.2, 0.4, 0.6, 0.7)
- Using different Optimal Parameters from our gridsearch to train SVD

	n_factors	**reg_all**	**Predictions RMSE**
SVD A	20	0.4	0.9598
SVD B	10	0.2	0.9582
SVD C	20	0.2	0.9581 #WINNER#