Efficient Hotel Recommendations

Jackson Barkstrom, Rohan Dalvi, Eric Liu, Alex Miao

Motivation

Driving Questions:

- **What did people who viewed this also view?**
- **What hotels are similar based on attributes?**
- What did similar people view?’

Objective:

- Answer the question: “Given that a user viewed this hotel, what is he/she most likely to look at?”

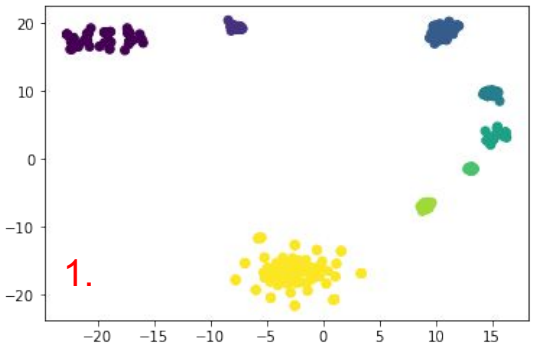
Similarity of Users' Hotel Preferences

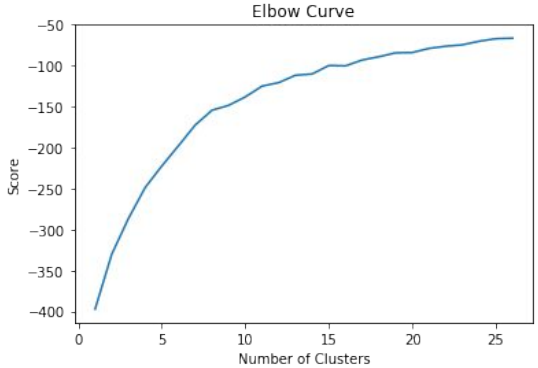
Construct a **correlation matrix** with hotels on both axes, where each cell describes the percent of viewers of hotel A who have viewed both hotel A and hotel B. This is very efficient, as this is almost all the data we need to give recommendations.

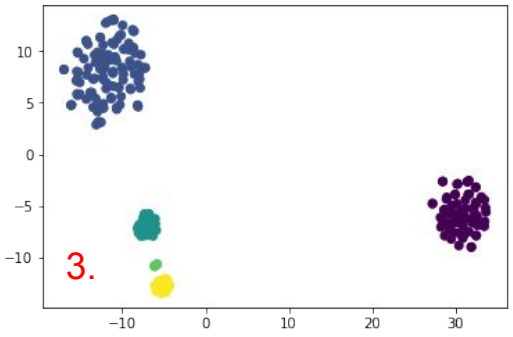
hotel_id	1858923.0	12297961.0	2079052.0	3235844.0	601762.0	99387.0	99302.0	208455.0	9335
75688.0	0.004182	0.005297	0.106496	0.024812	0.009479	0.024254	0.026485	0.011430	0.0328
75711.0	0.004900	0.002614	0.074812	0.020255	0.011761	0.011107	0.008821	0.002123	0.0406
80075.0	0.003430	0.002668	0.067073	0.034299	0.016006	0.007622	0.031250	0.001905	0.0472
80110.0	0.005719	0.003119	0.070445	0.013517	0.007538	0.053028	0.014557	0.019756	0.0135
93333.0	0.007928	0.007550	0.060778	0.021895	0.018875	0.030955	0.016233	0.001133	0.0351
93334.0	0.005159	0.004690	0.062852	0.015478	0.012664	0.013602	0.017355	0.005159	0.0276
93335.0	0.001080	0.003240	0.098092	0.014039	0.006299	0.060655	0.023398	0.040497	0.0097
93338.0	0.001132	0.003585	0.073962	0.017925	0.013208	0.019623	0.018679	0.012075	0.0194
93339.0	0.010259	0.005386	0.068992	0.031547	0.007694	0.015132	0.019492	0.002821	0.0679
93340.0	0.001212	0.002424	0.059394	0.010909	0.008485	0.009293	0.013737	0.040808	0.0157
93344.0	0.014666	0.005641	0.096119	0.028881	0.049639	0.016020	0.026963	0.006882	0.0766
93345.0	0.003209	0.001728	0.074549	0.016786	0.013330	0.014071	0.014317	0.004443	0.0266
93346.0	0.004521	0.003875	0.082984	0.023248	0.032935	0.014207	0.011947	0.005166	0.0309
93352.0	0.034412	0.005770	0.060581	0.008448	0.008448	0.008242	0.006594	0.002885	0.0175
93358.0	0.011491	0.006819	0.166667	0.128061	0.057936	0.011115	0.022981	0.007947	1.0000
93376.0	0.002833	0.003777	0.097891	0.015738	0.009443	0.007240	0.012590	0.002518	0.0358
93382.0	0.001751	0.008319	0.090630	0.029335	0.028021	0.017951	0.020578	0.040280	0.0323
93390.0	0.006143	0.004203	0.094730	0.026511	0.038151	0.018429	0.026188	0.003556	0.0375
93396.0	0.000342	0.008200	0.065938	0.014349	0.003075	0.020157	0.009224	0.031090	0.0092

(Note: for more efficiency in storage we could create a sparse matrix by changing ~0 values to 0. This would not meaningfully reduce our performance.

Justifying our Model with Clustering







1. Primary clustering on pure hotel attribute data
    a. Results: saw some good clustering, but we can do better
2. Created a custom metric called “hotel cross-over” **(our correlation matrix)** for more informative data
3. Final clustering with attribute data and hotel cross-over data
    a. Results: we do much better, justifying our model

Results

Given a hotel, we recommend the top 5 most least distant hotels based on the question “what did people who viewed this also view?”

```
[[34, 60, 58, 46, 131],
 [127, 15, 163, 33, 103],
 [103, 1, 119, 4, 106],
 [229, 224, 219, 51, 100],
 [43, 103, 106, 10, 11],
 [17, 54, 7, 55, 103],
 [104, 185, 18, 220, 88],
 [54, 55, 5, 83, 17],
 [103, 76, 106, 1, 2],
 [105, 78, 70, 104, 99],
 [95, 57, 84, 103, 39],
 [10, 103, 4, 57, 39],
 [57, 45, 58, 141, 10],
 [226, 91, 31, 215, 113],
 [95, 170, 125, 234, 10],
 [1, 163, 127, 33, 21],
 [156, 202, 237, 209, 138],
 [5, 54, 55, 7, 10],
 [6, 185, 104, 190, 78],
```

Utilizing Individual User History

Divide user's history into sub-histories (if necessary), and generate recommendations for each sub-history by finding the least-distant hotels to the hotels the user has interacted with

Sub-history separation

- Cluster the hotels into *n* clusters for various *n*
- Elbow Curve: Find the optimal number of clusters by finding the *n* where there is the greatest reduction

Hotel similarity

- Find the manhattan distances between every pair of hotels in each cluster

Recommendation Engine

- Each hotel h_j has a “match value” M_j calculated off of user sub-history $\{h_i\}$, indexed in order of recency

$$M_j = \Sigma \text{ similarity } (h_i, h_j) * w_i$$

- Where w_i is the weight assigned to each hotel h_i, such that more recent hotels are weighted higher.
- Similarity is inverse distance
- The hotels with the highest match value are recommended