Lab 2 Recommender System

Duruo Li

May 2, 2023

1 Introduction

This report is about recommender systems. I divide the report into two parts: Part 1. Data Pre-processing and Part 2. Build Recommendation Engines.

In Part 2, there will be three subsections which describe 3 different recommending strategies, i.e., popularity matching, content-based filtering and collaborative filtering. In these subsections, I will show the process of building engines and analyze their performances.

2 Data Pre-processing

2.1 Missing value

Question: 1

We should pay attention to missing values.

As for the table of reviews, we need to use customers' demographic and review information to build distance matrix, or else it's hard to define the "similarity" between users. According to the missing percentage table 1, that several columns contain missing values. However, only two columns, namely "vegetarian?" and "review text", have missing percentages exceeding 10%, i.e., 93% and 38%, respectively. Given that there are other more complete demographic data available, "vegetarian?" is dropped from the analysis. On the other hand, "review text" is retained as it provides unique and valuable information.

As for the table of restaurants, according to the missing percentage table 2, there is no missing problem.

Column	Missing_Percent
Reviewer Name	0.0000
Restaurant Name	0.0000
Rating	0.0000
Review Text	0.3809
Date of Review	0.0000
Birth Year	0.0014
Marital Status	0.0242
Has Children?	0.0263
Vegetarian?	0.9349
Weight (lb)	0.0672
Height (in)	0.0374
Average Amount Spent	0.0014
Preferred Mode of Transport	0.0048
Northwestern Student?	0.0007

Table 1: Missing percentage of customers' information

Column	Missing_Percent
Restaurant Name	0.0000
Cuisine	0.0000
Latitude	0.0000
Longitude	0.0000
Average Cost	0.0000
Open After 8pm?	0.0000
Brief Description	0.0000

Table 2: Missing percentage of restaurants' information

2.2 Data exploration: imbalance problem

Question: 2

To have a better undertanding of the data, we visualize the distributions of each features, and analyze the imbalance problems.

As for the table of reviews: see histograms 1. There are several interesting patterns:

- Marital Status: there is a strange alternative "widow", and there are quite a few. However, I was wondering if we set an alternative as "widower", will there also be a number of them?
- Birth Year: it seems that with age increasing, the number is decreasing, i.e., in this dataset, there exists imbalance between different age groups. Perhaps it's because younger people are more likely to be involved in such investigations.
- Northwestern Student?: it's imbalanced but reasonable, since residents in Evanston couldn't be mostly students. However, if we want to study the preference of Northwestern students, this training data isn't so reliable.
- Height: there exists an disproportionate number of people with a height between 155 and 158 cm

When it comes to the 'Cuisine' attribute of restaurants data, as shown in histogram 1, we observe that there are obvious issues of imbalance. Specifically, American cuisine has a much higher number of samples (8 individuals) compared to any other types which mostly only have 1 or 2 samples. Moreover, since the whole dataset is quite small (n=63), it's hard to oversample from the original data. Therefore, the cuisines which have only 1 or 2 samples may result in unrobust outcomes.

2.3 Data exploration: potential clusters

Questions: 3,4

To find potential patterns of customers, I use K-Means method to cluster their demographic data. According to silhouette score plot 2, Cluster_n=5 is observed to be the best, which has a silhouette score of about 0.43. However, since 0.43 is closer to 0 rather than 1, the points in the high-dimensional space are more likely to be overlapping rather than well separated.

Thus, there are no obvious clusters of users.

To be more specific, the three two-dimensional plots 3 suggest that there is no clear clustering pattern in people's height or weight. However, the last two plots show that most individuals in cluster 2 are relatively young, whereas the age distribution of individuals in the other clusters is more dispersed. Therefore, there are clusters based on people's age.

Examining the average review scores across the five clusters as shown in Table 3, it can be observed that there is a trend. The average review score increases from cluster 0 to cluster 2, reaches the peak in cluster 2, and then decreases in clusters 3 and 4. Interestingly, cluster 2 has a younger average age compared to the other clusters, suggesting that younger people may rate restaurants more favorably.

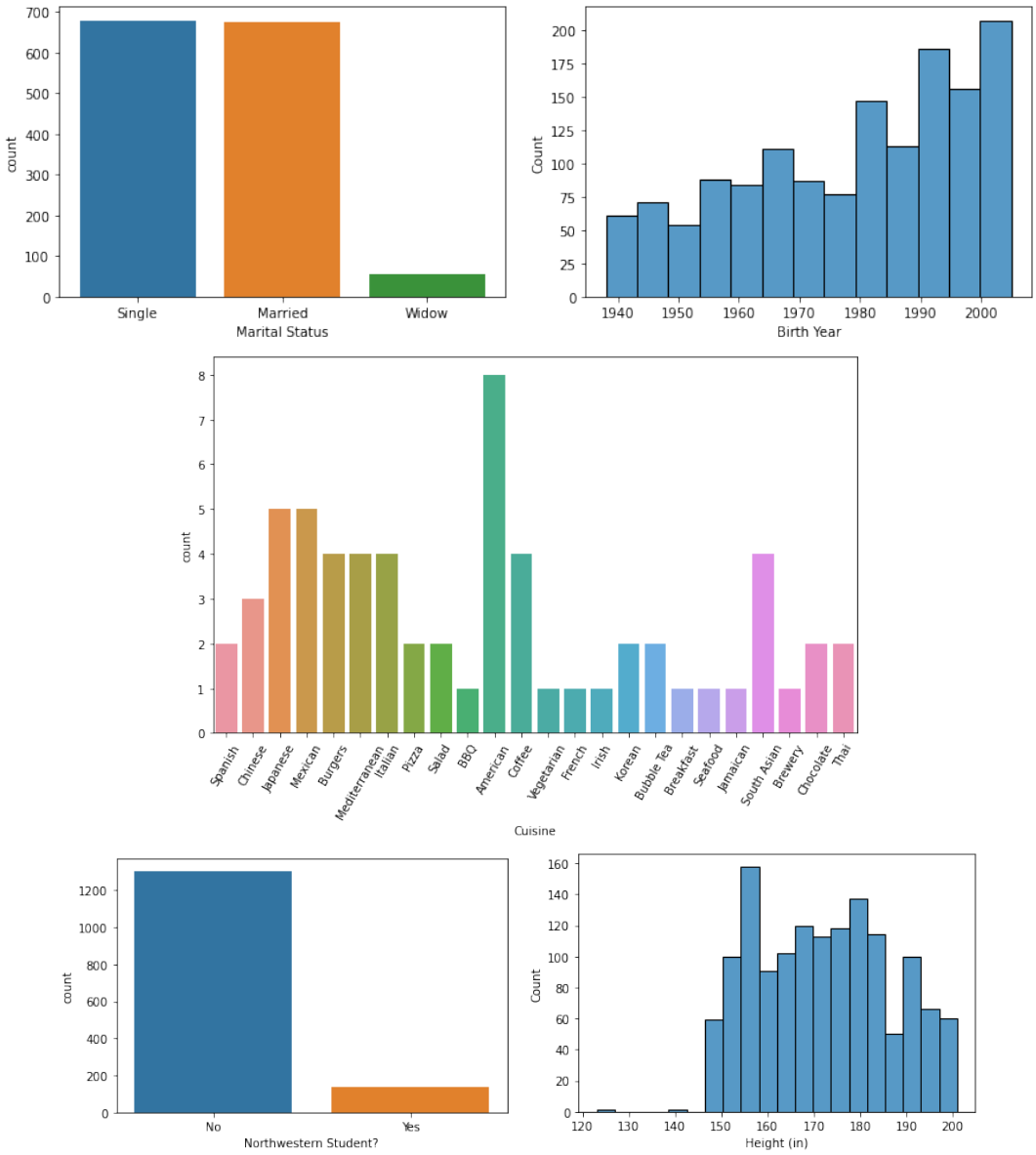

Figure 1: Data exploration: distribution histograms

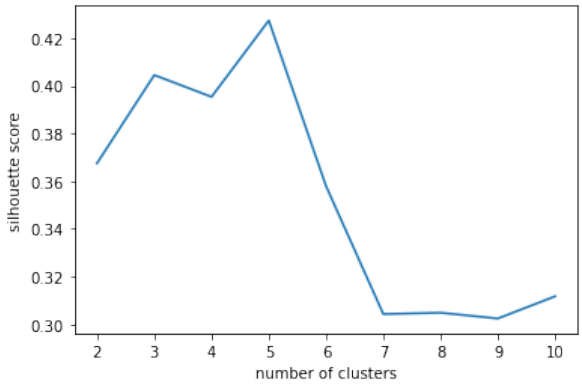

Figure 2
Silhouette score w.r.t number of clusters

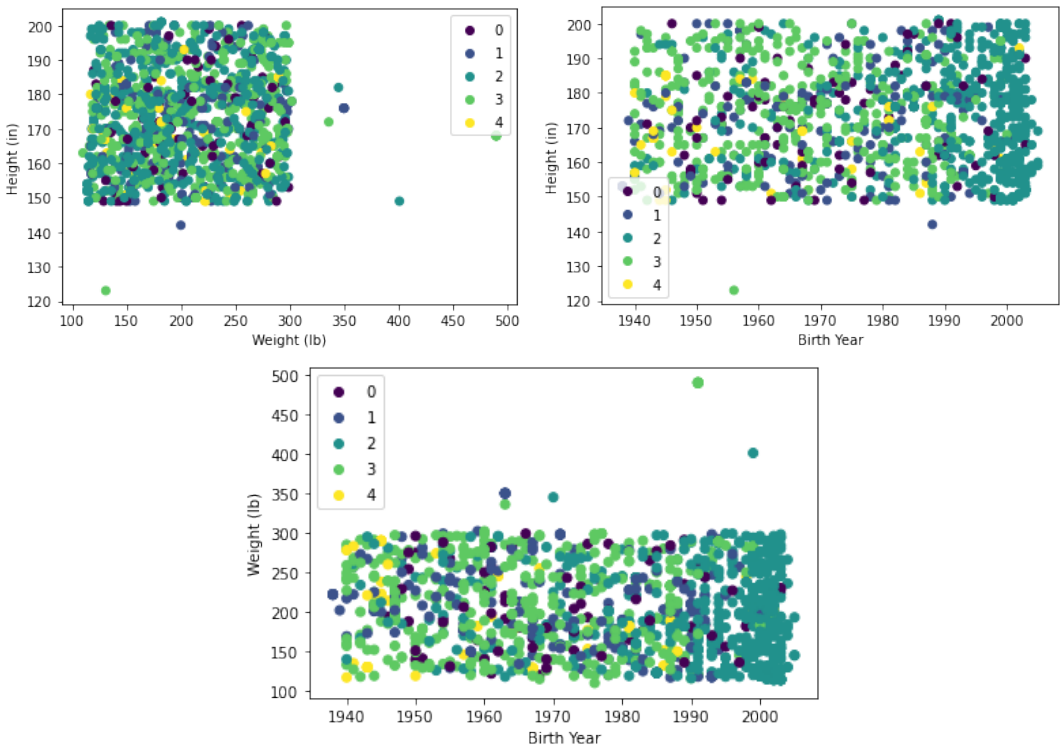

Figure 3: Two dimensional cluster distributions

Cluster	Average Review Score
0	3.789474
1	3.833333
2	3.867470
3	3.621247
4	3.423077

Table 3: Average review score of each cluster

count	67.000
mean	3.800
std	0.752
min	1.000
25%	3.439
50%	3.920
75%	4.215
max	5.000

Table 4: Descriptive table of review scores

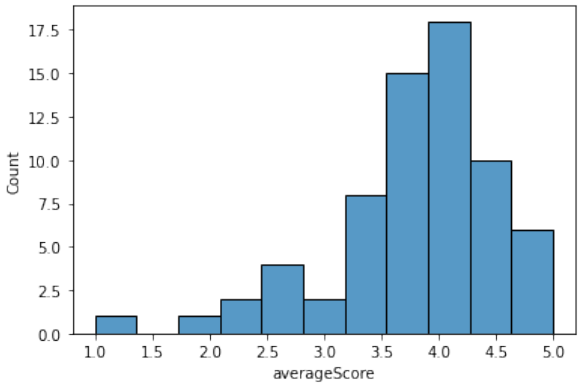

Figure 4
Distribution of average review scores

3 Build Recommendation Engines

3.1 Method 1: Popularity matching

Questions: 5,6,7,8

Popularity matching is a straightforward method for making recommendations, where the highest-rated item of a certain type is recommended to users. However, it's crucial to consider both the quantity and quality of ratings when defining "highest-rated" since a highly rated item may have received only a few ratings. And this is where "shrinkage" comes in.

According to some basic explorations in "Review Score", i.e., the average rating and the number of ratings for each restaurant, it's observed that:

- Average rating (table 4, figure 4): The majority of the average scores fall between 3.5 and 4.5, and the highest-rated restaurants are Evanston Games & Cafe, Fonda Cantina, La Principal, LeTour, and World Market, all with an average score of 5.
- Number of ratings: the median of the number of reviews is 23, with Campagnola having received the highest quantity at 48.

Using the raw review scores to build a simple recommendation engine, it gives some example recommendations as table 5.

However, there exists some risks, e.g., Evanston Games & Cafe has only one review recording despite its highest average rating. Therefore, shrinkage towards the mean score is needed. I choose to use $\alpha = min(\frac{N_i}{N}, 1)$ as the shrinkage parameter, N_i is the number of scores of restaurant i, and N is the overall number of scores.

After the scaling process, the overall changes in review scores are shown in figure 5, and the top 5 positive and negative changes are shown in figure 6. It's observed that the scaling procedure mainly

Cuisine Type	Restaurant Name
Spanish food	Tapas Barcelona
Chinese food	Joy Yee Noodle
Mexican food	Fonda Cantina
Coffee	Evanston Games & Cafe

Table 5: Example recommendations (popularity score)

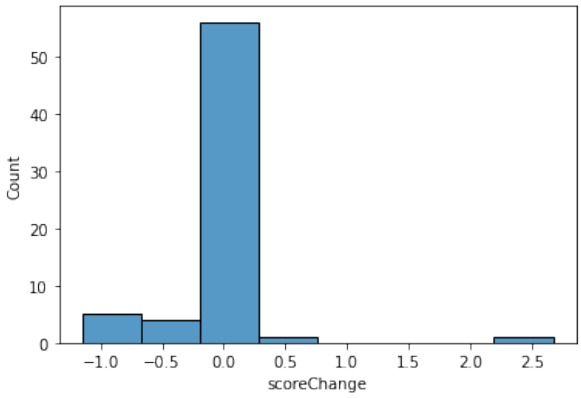

Figure 5
Changes in review scores (Overall)

affects restaurants with a small number of reviews. Those with low review scores (below the mean score) and a small number of reviews (below the mean review number) benefit the most, while those with high scores and a small number of reviews are hurt the most.

Among the restaurants, Clare's Korner benefits the most, with its score increasing from 1 to 3.67 (an increase of 2.67). On the other hand, Evanston Games & Cafe is hurt the most, with its score decreasing from 5 to 3.856 (a decrease of 1.144).

3.2 Method 2: Content based filtering

Questions: 9-18

I think the core of recommender systems is to find the best way to define "similarity", either between items or users. From this perspective, the content-based filtering is a way focusing on the

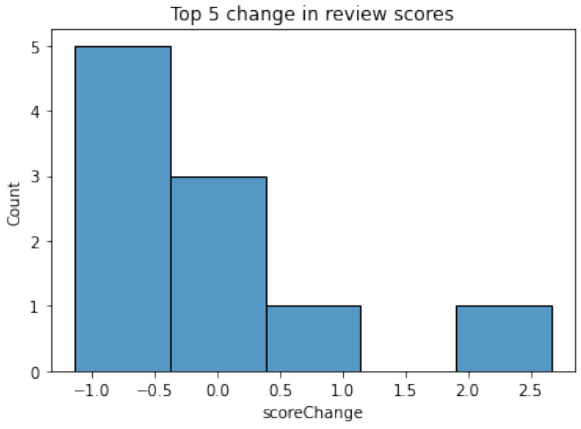

Figure 6
Changes in review scores (Top 5 positive and negative)

Top Recommend	Restaurant Name
1	Trattoria DOC
2	Dave's Italian Kitchen
3	Peppercorns Kitchen
4	Sweet Green
5	Taste of Nepal

Table 6: Example recommendations (user: Kim Hamilton)

side of items, i.e., recommend the "closest" items to the items already being selected. In our case, the primary goal is to find a best way to define the "distance" between restaurants based on their background information.

3.2.1 Numeric and categorical data

One traditional way is to transform the numeric and categorical data of each restaurant into multi-dimensional "coordinates" respectively, and then use metrics such as Euclidean or cosine distance to form a distance matrix. The recommending strategy is then simply to recommend the closest restaurants to the user's highest-rated one.

As an example, for a given user: Kim Hamilton and a chosen distance: euclidean distance, the recommendation engine gives top 5 recommendations as following table 6.

3.2.2 Text data

If taking text data into account, e.g., the brief description of the restaurant, the core is the transformation from natural language to numerical "coordinates" (vectors). There are several ways such as Jaccard distance(one-to-one; intuitive), TF-IDF(representative words; local and total), BERT/Word2Vector (embedding; black box), etc.

After augementing the description by attaching the restaurant's cusine type, I build three distance matrix based on three methods:

- Jaccard distance: $Jaccarddistance = 1 - Jaccard$ similarity. This method doesn't need the first step to transform natural language into vectors.
- TF-IDF: first build a function to calculate a word's TF-IDF score for each augmented description, e.g., restaurant Taste of Napel has the highest TF-IDF score for word 'cozy' at 0.277, and Lao Sze Chuan has the highest score for 'Chinese' at 0.251. Then transform each augmented description into a 100-dimensional coordinate based on the 100 most popular words in all descriptions. And use these coordinates to build distance matrix using either cosine or euclidean metric.
- BERT: each augmented description is embedded to a 384-unit vector by using a pre-trained large language model. Then, use embedded matrix to compute embedding-distance matrix (euclidean or cosine)

And then, use these three distance matrices to make recommendations.

Model comparison:

To compare the performance of each engine based on different distance matrix, the most reasonable way to evaluate a recommendation is to see whether the user will really accept it. However, it's impossible in this case since we cannot get future data.

Another intuitive way is to use available data to analyze people's reviews on top "recommendations"—those should be recommended to them. There are two ways to leverage reviews: (1) focus on concrete scores, which is not practical as it's hard to define the "be-supposed" score; (2) focus on the "occurrence" and treat each review as an indicator, i.e., if people have been to the restaurant, it should be labeled as "been favored and should be recommended" regardless of the score.

My method is based on idea (2). To evaluate the method's performance, I will compare the top 5/10 recommendations from the distance matrix with restaurants that customers have actually

Method	Hit Rate
Jaccard distance	0.237
TF-IDF score	0.181
BERT	0.302

Table 7: "Hit Rate" of three distances

Similar user	Distance
Bridgett Colley2001	0.746355
Jay Colston1985	0.830256
Sanford Persaud1999	0.940802
Patricia Beattie1994	1.190599
Dana Thomas1981	1.326136

Table 8: Distance between target user and "collaborative" users

reviewed. The **evaluation metric** will be the "Hit Rate": average of the number of "intersections" (i.e., restaurants recommended and also reviewed by customers) divided by the number of real reviews for each customer.

The comparison outcomes are as table 7. BERT distance performs the best according this evaluating method.

P.S. note that only 88 of all the 1047 users have more than 1 reviews, and 3 of whose reviews are for the same restaurant.

3.3 Method 3: Collaborative filtering

Questions: 19-23

The essence of collaborative filtering is to make use of "collaborative information," i.e., information from other users.

Considering the idea of "similarity" , we are now focusing on the side of users.

3.3.1 Demographic data

After exploring the data, I discovered that "Reviewer Name" is insufficient to identify individual users due to the presence of "namesake". Thus, I create a new feature "cName-birth" by combining "Reviewer Name" and "Birth Year" to be the unique label for each user.

Form a demographic vector for each user by encoding the categorical data. To prevent certain features from dominating the distance calculation, I scale the entire matrix. To make recommendations, I have two ideas:

- Select K+ nearest users, recommend their highest-rated restaurants (overlapping may occur)
- Select multiple nearest users and recommend all their selections until K recommendations have been made

It depends on which we prefer: highly-rated restaurants or closer users. Since recommending highest-rated restaurants makes more sense to me, I choose the first idea.

To demonstrate the system, I take "Sarah Hardy1994" as our target user. The system made 5 recommendations: '5411 Empanadas', 'Epic Burger', 'Chipotle', "Hecky's BBQ", 'Picnic'. The distance between the user and the user used to make recommendations is shown in table 8

3.3.2 Review score vector

After filtering all the users with at least 4 reviews, I build a 66-dimensional(weird, not 64-dimensional) score vector for each user. However, the score matrix for users is quite sparse, so we need to fill in the blanks (at least some of them).

There are three possible ways:

Method	Hit Rate
Demographic	0.163
Review score	0.206

Table 9: "Hit Rate" of demographic vector model and review score vector model

- Use distance matrix of restaurants, closer =¿ similar scores
- Clustering: mutate the missing score by the average score of the restaurant in the cluster to which the user belongs
- Matrix factorization

I choose to use the second method. To make use of more data, I do the clustering based on the whole dataset instead of those with at least 4 reviews.

While doing the mutation, I encountered a problem where only certain restaurants have average scores from the clustered data, i.e., each cluster excludes certain restaurants. As a result, some of the missing restaurants' scores could not be filled up. However, the resulting matrix after filtering is much less sparse than before.

Then, we use the updated score vector as the "coordinate" for users and calculate the distance matrix which are used to make recommendations afterwards.

p.s. There is a potential problem: the distances between individuals largely depend on the clusters they belong to, as the members within each cluster tend to have similar feature vectors, with most of the values (the blanks being filled) being the average ratings of the cluster.

3.3.3 Model comparison

I use the same comparison method: the metric of "hit rate", i.e., compare the hit rate of recommendations with respect to the actual review records (before mutation). The outcomes are as table 9. It seems that the review score model performs better.

3.4 Model 3 Plus: Predictive models

Questions: 24-30

Roughly speaking, predictive modeling is also a collaborative method since it takes other users' information into account.

3.4.1 Model 0: demographics+cuisine

Standard model:

I use user demographics and cusine type to build a prediction model. To evaluate its performance, I split the data into 80% train data and 20% test data. The MSE of the test data is 1.967. And by taking the 155th record in the data, I do a single prediction test. The predicted score is 4.625 while the real review score is 3. Therefore, the prediction isn't very accurate.

Lasso regularized model:

To do regularization, I add an L1 penalty to the standard linear model. By applying 5-fold cross-validation, the best hyperparameter α is found to be 0.005465. To compare the performance of two models:

standard Linear Regression MSE = 1.967

Lasso Regression MSE = 1.946

I.e., lasso regression model performs better.

The coefficients tables of two models are shown as table 10. It's observed that:

- Positive weights: the coefficient of 'Cuisine_Chocolate' is the largest, i.e., 0.424, 'Has Children?_No', 'Cuisine_Salad' and 'Cuisine_Japanese' also have relatively large weights, i.e., greater

than 0.1. On the other hand, the coefficient of 'Northwestern Student?_Yes' is almost 0, i.e., 10^{-16}

- Negative weights: the coefficient of 'Cuisine_Burgers' is the smallest (largest magnitude), i.e., -1.033, and 'Cuisine_American', 'Cuisine_Burgers', 'Cuisine_Chinese', 'Cuisine_Italian', 'Cuisine_Mediterranean', 'Cuisine_Mexican', 'Cuisine_South Asian', 'Cuisine_Thai', 'Average Amount Spent_Low', 'Northwestern Student?_No' also have large magnitude of weights i.e., greater than 0.1. 'Has Children?_Yes"s effect can almost be neglected, i.e., 10^{-16}
- Zero weights: 'Cuisine_BBQ', 'Cuisine_Breakfast', 'Cuisine_Bubble Tea', 'Cuisine_French', 'Cuisine_Irish', 'Cuisine_Jamaican', 'Cuisine_Korean', 'Cuisine_Pizza', 'Cuisine_Seafood', 'Cuisine_Spanish', 'Cuisine_Vegetarian', 'Marital Status_Married', 'Average Amount Spent_High', 'Average Amount Spent_Medium', 'Preferred Mode of Transport_On Foot' Many cuisine type indicator features have been shrunk to 0. Except for thes features, others are those being selected by the penalty term.

To conclude, the demographic features whose coefficients have a larger magnitude are more predictive of score. E.g., 'Has Children?'(don't have children), 'Preferred Mode of Transport'(car owner), 'Marital Status'(the indicator of widow),' Average Amount Spent'(low average spent), 'Northwestern Student?'(no), etc.

3.4.2 Model 0.5: embedded review text

I use BERT method to transform 'Review Text' into embedded vectors, and the model error (80%/20% split) is: MSE=1.887

3.4.3 Model 1: demographics+embedded review text+cuisine

Based on Model 0, I add embedded review text into predictors to build Model 1.

To compare the performances of two models, Model 0 and Model 1, I use metrics:**MSE, R-squared, adjusted R-squared, AIC,** and **BIC** . See table 11, according to all the metrics, Model 1 performs better (especially in R-squared and adjusted R-squared), i.e., the embedded review text indeed improves the predictive power.

3.4.4 Model 2: a specific model (coffee)

If we are interesting in a special type of cuisine, e.g., coffee, we can build a specific linear model for it—using demographic data to predict the review score for coffee.

However, in the standard linear model, none of the predictors are statistical significant according to p-value, see table 13.

Feature selection:

If we regard coefficients of the standard model as the representative of feature importance, 7 features are selected: 'Has Children?_No', 'Has Children?_Yes', 'Average Amount Spent_Low', 'Preferred Mode of Transport_On Foot', 'Preferred Mode of Transport_Public Transit', 'Northwestern Student?_No', and 'Northwestern Student?_Yes'.

Then we build a new model based on the 7 selected features. First, we compare the performances of the raw model and model after feature selection using metrics as before, see table 12. Except for R-squared, all the other 4 metrics show that updated model performs better, i.e., dropping those variables doesn't hurt, or even improves the model.

Moreover, the coefficient and p-value table, we can see that the first 5 most important features (large magnitude of coefficients) are statistically significant (at a level of 0.05).

The coefficients suggest that: people who do not have children and are Northwestern students have a stronger preference for coffee. However, people who have children or are not Northwestern students also like coffee, but to a lesser extent according to the coefficients. On the other hand, those with low average spending and those who prefer public transit have a negative attitude towards coffee.

Feature	Standard	Lasso
Cuisine_Chocolate	0.5780	0.4238
Has Children?_No	0.1623	0.2745
Cuisine_Salad	0.6934	0.1948
Cuisine_Japanese	0.2393	0.1373
Cuisine_Coffee	0.2112	0.0435
Preferred Mode of Transport_Car Owner	0.1005	0.0333
Cuisine_Brewery	0.3294	0.0122
Weight (lb)	0.0008	0.0007
Marital Status_Single	0.2242	0.0005
Northwestern Student?_Yes	0.1198	0.0000
Marital Status_Married	0.2200	0.0000
Average Amount Spent_High	0.0600	0.0000
Average Amount Spent_Medium	0.0466	0.0000
Cuisine_Vegetarian	-0.0829	-0.0000
Cuisine_Spanish	0.1508	0.0000
Cuisine_Seafood	-0.1601	-0.0000
Preferred Mode of Transport_On Foot	0.0262	-0.0000
Cuisine_Pizza	0.2314	0.0000
Cuisine_French	0.9983	0.0000
Cuisine_Irish	0.0559	0.0000
Cuisine_Breakfast	0.3751	0.0000
Cuisine_Bubble Tea	0.1111	0.0000
Cuisine_BBQ	-0.3029	-0.0000
Cuisine_Jamaican	0.2925	0.0000
Cuisine_Korean	0.0808	0.0000
Has Children?_Yes	-0.1623	-0.0000
Birth Year	-0.0030	-0.0010
Height (in)	-0.0031	-0.0031
Cuisine_Chinese	-0.1067	-0.0608
Preferred Mode of Transport_Public Transit	-0.1266	-0.0898
Northwestern Student?_No	-0.1198	-0.1216
Average Amount Spent_Low	-0.1066	-0.1374
Cuisine_South Asian	-0.2859	-0.2285
Cuisine_Mexican	-0.2920	-0.2567
Cuisine_Italian	-0.3233	-0.2777
Cuisine_Mediterranean	-0.3955	-0.3566
Marital Status_Widow	-0.4442	-0.4062
Cuisine_American	-0.5668	-0.5635
Cuisine_Thai	-0.7704	-0.6647
Cuisine_Burgers	-1.0608	-1.0330

Table 10: Coefficients comparison: standard linear vs Lasso (descending order)

Metric	Model 0	Model 1(+embedded review text)
MSE(train/test split)	1.967	1.927
R^2	0.106	0.850
Adjusted-R^2	0.082	0.709
AIC	4672	2363
BIC	4854	4346

Table 11: Model comparisons: Model 0 versus Model 1(+embedded review text)

Metric	Model 2	Model 2_1 (feature selection)
MSE(train/test split)	3.008	2.434
R^2	0.241	0.196
Adjusted-R^2	0.051	0.107
AIC	177.3	170.2
BIC	198.5	181.8

Table 12: Model comparisons: Model 2 versus Model 2_1 (feature selection)

	coefficient	p-value
const	3.1696	0.7569
Birth Year	-0.0038	0.8139
Weight (lb)	0.0045	0.2314
Height (in)	0.0065	0.6273
Marital Status_Married	1.6771	0.7427
Marital Status_Single	1.4925	0.7720
Has Children?_No	2.1815	0.6799
Has Children?_Yes	0.9882	0.8425
Average Amount Spent_High	1.2231	0.7208
Average Amount Spent_Low	0.7143	0.8339
Average Amount Spent_Medium	1.2323	0.7219
Preferred Mode of Transport_Car Owner	0.9565	0.7776
Preferred Mode of Transport_On Foot	0.7785	0.8187
Preferred Mode of Transport_Public Transit	1.4346	0.6837
Northwestern Student?_No	1.1667	0.8167
Northwestern Student?_Yes	2.0030	0.7013

Table 13: Coefficient and p-value for Model 2 (coffee)

	coefficient	p-value
const	2.301919	0.000000*
Has Children?_No	1.571067	0.000000*
Has Children?_Yes	0.730852	0.002857*
Average Amount Spent_Low	-0.457226	0.333601
Preferred Mode of Transport_On Foot	-0.269300	0.509246
Preferred Mode of Transport_Public Transit	0.534560	0.450978
Northwestern Student?_No	0.719788	0.001077*
Northwestern Student?_Yes	1.582131	0.000012*

Table 14: Coefficient and p-value for Model 2_1 (coffee; feature selection)

4 Interesting Finding

Question: 31

1) Namesake

Upon analyzing the data, I noticed that there are some individuals with the same reviewer name but with different demographic features.

In some cases, it turns out that they are the same person, but some life events have occurred, such as having children! Examples of such individuals include Edward Smith, Raymond Wickstrom, Barbara Mcelroy, and Paris Hancock.

On the other hand, in some cases, different people happen to share the same name. Examples of such individuals are Maria Reading, Robt Ortiz, Julia Turner (one is much taller than the other, 185 vs. 152), and Jodi Bougie. I also found it surprising that both Richard Shelton individuals are widows. Another interesting case is John Holm, where two men with the same name are of the same age.

Additionally, I came across a peculiar case with Jodee Cryderman, whose birth year changed from 2003 to 1999, while all the other features remained the same. I suspect that this is a typo and corrected the birth year to 2003 since 2/3 of the records had that year.

In terms of clustering, I decided to treat individuals with different demographic features as distinct individuals since they may be in different life phases. However, when it comes to making recommendations, for individuals who belong to multiple clusters at different life phases, I assign them randomly to one of the clusters.

2) Exclude certain restaurants

Each cluster seems to exclude certain restaurants. For example, cluster 0 only includes 31 out of 66 restaurants, cluster 1 only includes 54 out of 66 restaurants, while cluster 2 includes 65 out of 66 restaurants. Although there are no obvious demographic features that distinguish the clusters, this exclusion pattern could be utilized to make restaurant recommendations. Specifically, certain restaurants could be excluded from recommendations for certain groups of customers.

5 Conclusion

Be careful about defining "similarity". There may be underlying patterns that only algorithms can identify and understand, such as matrix factorization. This can be both fascinating and intimidating, as computers may know more about us than we know about ourselves. Perhaps this is because they can perceive us in a higher-dimensional view than our brains can comprehend.