Lab3 for STAT 415

Xiaolong Liu

April 2024

1 Introduction

In this homework, we will continue our exploration of recommender systems, focusing on collaborative filtering and predictive modeling techniques. In particular, we will engage with Linear Modeling.

Also, we find that some numbers in the *Analysis Instructions* are not accurate. For example, in part 3, the dimension is not 63 but 68. In part 8, the number of coffee shops should be 4, not 3. This discrepancy may be attributed to a recent update of the dataset.

In addition to what we have addressed in the previous lab, we have identified inconsistencies in the values within the **Marital Status** column. Specifically, there are entries listed as **widow** and **widowed**, as well as **Single** and **SIngle**. These should be standardized to consistent values. We apologize for not catching these discrepancies sooner.

2 Collaborative Filtering

2.1 Create a user feature matrix

We collected users' demographic information from **reviews.csv** which is already processed partially (we have added a class **'Missing'** to replace **NaN** in the categorical variables and applied one-hot encoding to all categorical variables). Additionally, we have replaced the **NaN** values in the numerical variables **Birth Year**, **'Weight (lb)'** and **'Height (in)'**, because we believe it is reasonable to use the mean as the value, so that when calculating distances, this uses the distance at the 'center' (we cannot simply discard these variables because they contain some missing values). We decided to use the following demographic variables: **'Reviewer Name'**, **'Birth Year'**, **'Marital Status'**, **'Has Children?'**, **'Vegetarian?'**, **'Weight (lb)'**, **'Height (in)'**, **'Average Amount Spent'**, **'Preferred Mode of Transport'**, and **'Northwestern Student?'**. For the numerical variables **'Birth Year'**, **'Weight (lb)'**, and **'Height (in)'**, we have also standardized them.

We have a total of 1068 unique user vectors (which means 1068 different users), and for each vector, its dimension is 26 (3 numerical features and one-hot encoding for 8 categorical features).

2.2 Recommendation algorithm based on user feature matrix

We used the vectors created in the previous question to write a function that computes the cosine distance from one user to all others (recall that, in previous lab problems, we decided to use cosine instead of Euclidean distance). With this function, we build a recommendation algorithm that takes a user and outputs a recommendation made by the most similar user (i.e., the most similar user's favorite restaurant).

We select **'Timothy Mace'** as our user. His most similar user, according to our algorithm, is **'L S'**, and the recommendation provided is **Cross Rhodes** (only one recommendation).

Based on this example, our algorithm first calculates the cosine distance between **'Timothy Mace'** and all other users respectively, based on the users' vectors from the first problem in this report. **'L S'** is identified as the closest user. Then, the algorithm checks all the restaurants **'L S'** has rated and chooses the restaurant with the highest rating as the recommendation, **Cross Rhodes**.

This algorithm does not suggest more than one recommendation for every user in the dataset because some users only have one restaurant with the highest rating. If we want the algorithm to give more than one recommendation, here is the way to get them: 1. We can find all the restaurants the closest user has rated and recommend the top n restaurants. 2. If, after selecting the closest user, they only generate one recommendation, the algorithm will look for the next closest user to generate more recommendations.

3 Construct similar reviews vectors

Rather than finding users that are similar in terms of demographics, we aim to find users who gave similar reviews. To identify users who have given similar reviews, for each user j, we form a vector where entry i represents user j's review of restaurant i.

There are 68 different restaurants in **reviews.csv** and 64 different restaurants in **restaurants.csv**. Since we are utilizing the information in **reviews.csv** only, we will construct a 68-dimensional vector.

The vector has many blank entries. To fill in these blanks, we can use KNN (recall that we have performed clustering based on the categorical variables in a previous lab; here, we will use clustering based on all the demographic data) to cluster the users and use the mean of the clusters to fill in the blanks.

DBSCAN treats all the data into one cluster, which is not what we want to see. Both Kmeans and AgglomerativeClustering have shown similar results; hence, we will use Kmeans here.

First, we temporarily impute the mean value for the **NaN**s. Then, we should choose a suitable number of clusters. According to the inertia plot, we observe that the inertia decreases rapidly at first, and then the curve becomes flat, indicating that the optimal number of clusters is $k = 6$.

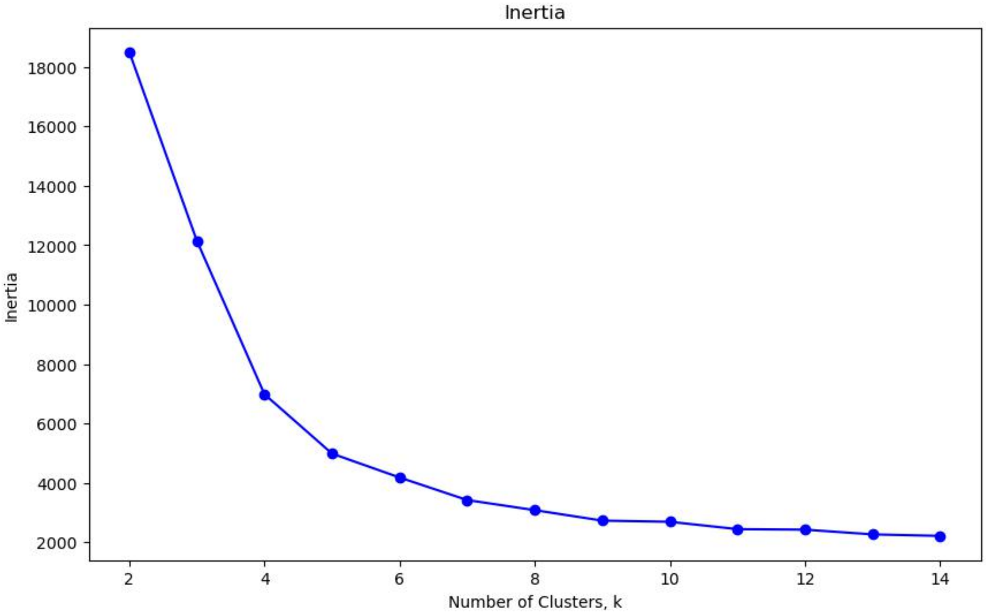


Figure 1: Inertia plot for Kmeans showing the selection of $k = 6$.

From the t-SNE plot, we can observe that the clustering result is satisfactory.

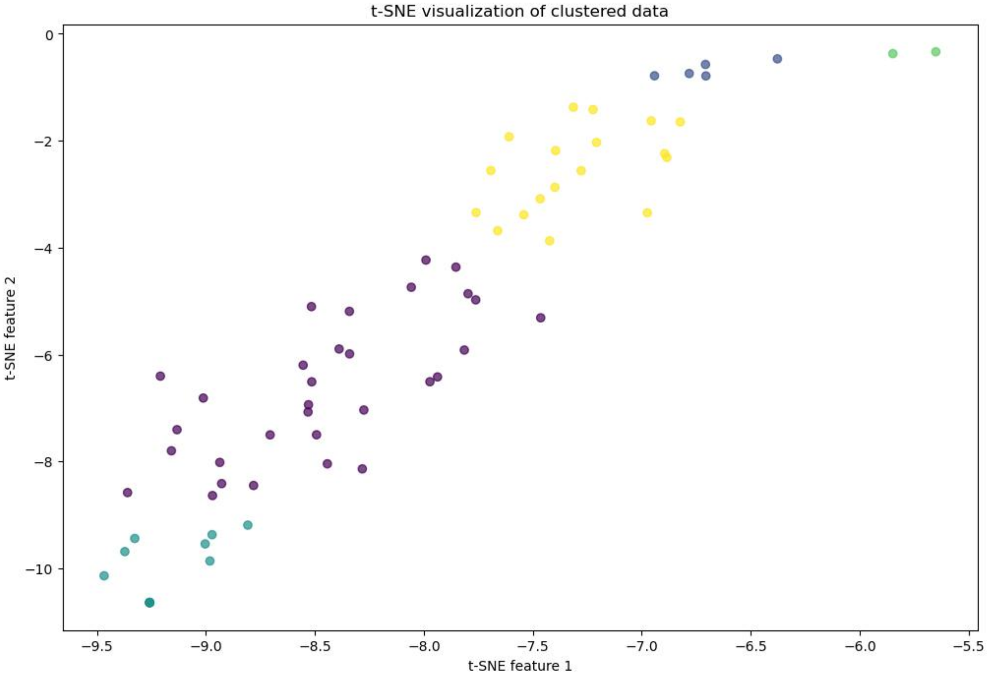


Figure 2: t-SNE plot demonstrating the effectiveness of the clustering.

Finally, we use the clusters' mean values to impute the **NaN**s as the final values.

3.1 Recommendation algorithm based on similar reviews vectors

Similar to part 2, we write a function that computes the distance from one user to all others based on the vectors created in step 3.

We select **'Sarah Belle'** as our user. Her most similar user, according to our algorithm, is **'Johnny Mcginnis'**, and the recommendation provided is **Mumbai Indian Grill** (only one recommendation).

Based on this example, our algorithm first calculates the cosine distance between **'Sarah Belle'** and all other users respectively, based on the similar reviews vectors from Part 3. **'Johnny Mcginnis'** is identified as the closest user. Then, the algorithm checks all the restaurants **'Johnny Mcginnis'** has rated and chooses the restaurant with the highest rating as the recommendation, **Mumbai Indian Grill**.

4 Predictive Modeling

4.1 Linear Regression

We develop a linear regression model that integrates demographic data and the cuisine type of restaurants to accurately predict a restaurant's rating. We find that some restaurants in the **reviews.csv** file are not present in the **restaurants.csv** file. We assign the class **'missing'** as their cuisine type and use one-hot encoding to represent different cuisine types.

We then use the demographic data and encoded cuisine types to predict the rating scores. The detailed procedure will be discussed in the next section.

4.2 Evaluating the linear regression

To be concrete, we split the data into two parts: 20% of the data as test data and the remaining as training data.

We then fit the linear regression model using the demographic data and cuisine type to predict the rating scores.

The Mean Squared Error (MSE) on the training dataset is approximately 1.797, and on the test dataset, it is around 2.078.

To illustrate its performance, we randomly select one reviewer, **'NU Student 4'**, rating the restaurant **'Brothers K Coffeehouse'**. The actual rating score is 5, and the model's prediction is 4.125, which we consider to be reasonable.

However, we also calculate the standard deviation of the dataset, which is around 1.45. This value is much lower than the MSE for both the training and test datasets, suggesting that this model is not very effective.

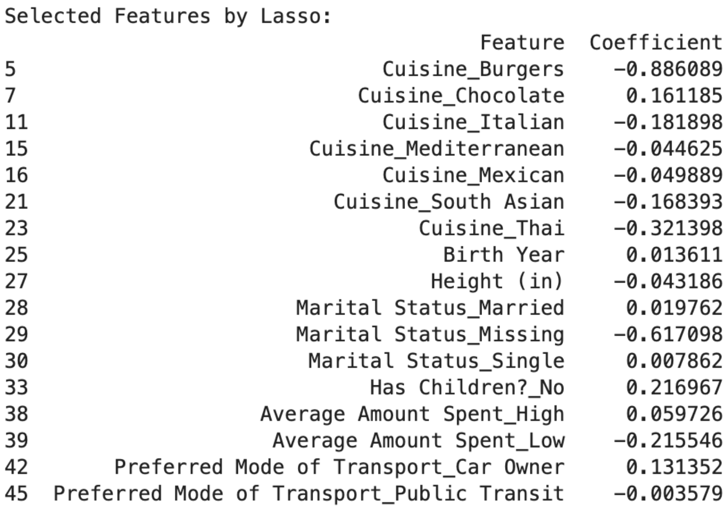
4.3 Lasso regularization

For this part, we introduce Lasso regularization into the linear regression model. Since Lasso incorporates a hyper-parameter, α, which must be determined before fitting the model, we now divide our entire dataset into three parts: 60% for training, 20% for validation, and 20% for testing.

We utilize a grid search to identify the best hyper-parameter α based on the Mean Squared Error (MSE). Specifically, for each value of α, we fit a Lasso model on the training dataset, evaluate it on the validation dataset, and select the α that yields the best performance on the validation dataset. Afterwards, we assess this model on the test dataset.

The grid of α values tested is $[0.001, 0.01, 0.1, 1, 10]$. Ultimately, we choose $\alpha = 0.01$ with MSE on the training dataset as 1.82%, MSE on the validation dataset as 2.07, and MSE on the test dataset as 2.09. These results are similar to those in the previous problem, noting that our training dataset has become smaller but the size of the test dataset has not changed.

Regarding the features selected by Lasso, we find that almost all the cuisine types selected carry negative weights, except for **chocolate**, which has a positive weight. **Birth Year** and **Preferred Mode of Transport_Public Transit** have quite small weights. **Height (in)**, **Marital Status_Missing**, and **Average Amount Spent_Low** have negative weights, while the remaining variables have positive weights. **Cuisine_Burgers** shows the most negative weight and **Has Children?_No** the most positive. Thus, variables such as **Northwestern Student?**, **Preferred Mode of Transport_On Foot**, **Preferred Mode of Transport_Missing**, **Average Amount Spent_Medium**, **Average Amount Spent_Missing**, and several types of cuisine like **Cuisine_American**, **Cuisine_BBQ**, **Cuisine_Breakfast**, **Cuisine_Brewery**, among others, are deemed non-contributive according to the Lasso.



```
Selected Features by Lasso:
                                       Feature  Coefficient
5                              Cuisine_Burgers    -0.886089
7                            Cuisine_Chocolate     0.161185
11                             Cuisine_Italian    -0.181898
15                       Cuisine_Mediterranean    -0.044625
16                             Cuisine_Mexican    -0.049889
21                         Cuisine_South Asian    -0.168393
23                                Cuisine_Thai    -0.321398
25                                  Birth Year     0.013611
27                                 Height (in)    -0.043186
28                      Marital Status_Married     0.019762
29                      Marital Status_Missing    -0.617098
30                       Marital Status_Single     0.007862
33                             Has Children?_No     0.216967
38                  Average Amount Spent_High     0.059726
39                   Average Amount Spent_Low    -0.215546
42        Preferred Mode of Transport_Car Owner     0.131352
45   Preferred Mode of Transport_Public Transit    -0.003579
```

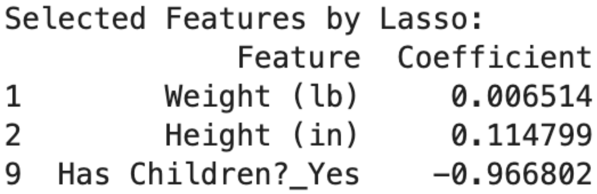
4.4 Linear model for coffee restaurants only

We investigate the utility of demographic features for predicting coffee scores, focusing on 4 coffee shops within our dataset. For these specific restaurants, we employ a linear model using demographic data to predict the scores, following the same methodology as previously discussed.

We select $\alpha = 0.1$ for the Lasso regularization in our model. The Mean Squared Error (MSE) outcomes for this setup are as follows: MSE on the training dataset is 1.21%, MSE on the validation dataset is 2.06, and MSE on the test dataset is 0.5. The notably lower MSE on the test dataset may be attributed to the small size of this dataset, suggesting that while the model performs well, the results might be influenced by the limited data volume.

4.5 Examine the weights

We then examine the weights produced by the linear model developed in the previous section. The Lasso regularization selected only three variables: **Weight (lb)** and **Height (in)**, with the latter having the highest weight, and **Has Children?_Yes** with the lowest weight.



```
Selected Features by Lasso:
              Feature  Coefficient
1         Weight (lb)     0.006514
2         Height (in)     0.114799
9   Has Children?_Yes    -0.966802
```

Figure 3: Selected features by Lasso for predicting coffee scores.

Based on these results, we might speculate that individuals who do not have children are more likely to frequent coffee shops. This interpretation suggests a potential lifestyle or availability factor influencing coffee shop visits.

5 Text Embedding

We now incorporate the **'Review Text'** column into our analysis by embedding the review texts into vectors using Sentence Transformers. These embeddings serve as features to train a linear model aimed at predicting the review scores.

The performance of this model is quantitatively assessed through the Mean Squared Error (MSE) on different datasets: MSE on the training dataset is remarkably low at 0.56%, MSE on the validation dataset is 0.98, and MSE on the test dataset is 0.73. These results are exceptionally good and represent the best performance among all the regression models we have fitted so far.

We attribute this success to the **'Review Text'** containing more explicit attitudes towards the restaurant, which significantly enhances the accuracy of the fitted scores.

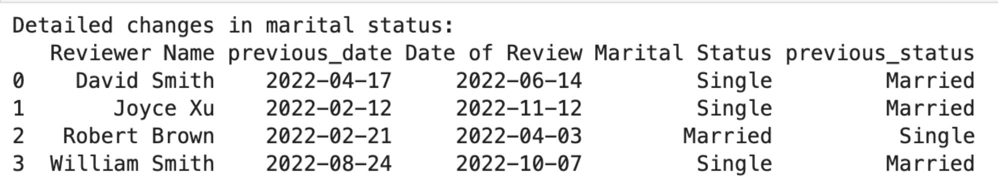
6 Final

At last, in this dataset, we uncover some intriguing findings:

1. The reviewer **Castor Z** rated the restaurant **Evanston Chicken Shack** a staggering 15 times, consistently giving a score of 1! All of his review texts are negative. Interestingly, this is the only restaurant he has rated, which suggests a particularly strong aversion to this establishment.

2. Apart from **Castor Z**, **Steven Rusert** has given every restaurant he reviewed a perfect score of 5 (13 times in total, with some repeats).

3. There are 3 individuals who have changed their **Marital Status**: **David Smith**, **Joyce Xu**, and **Robert Brown**, as well as **William Smith**. Notably, **Robert Brown** has gotten married, while the others have gotten divorced.

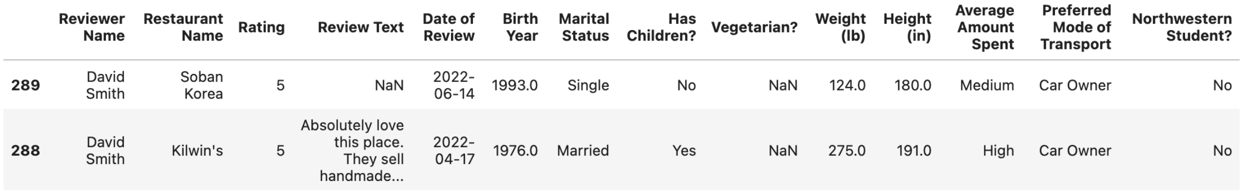


```
Detailed changes in marital status:
   Reviewer Name previous_date Date of Review Marital Status previous_status
0    David Smith    2022-04-17     2022-06-14         Single         Married
1       Joyce Xu    2022-02-12     2022-11-12         Single         Married
2   Robert Brown    2022-02-21     2022-04-03        Married          Single
3  William Smith    2022-08-24     2022-10-07         Single         Married
```

Figure 4: Marital Status Changes

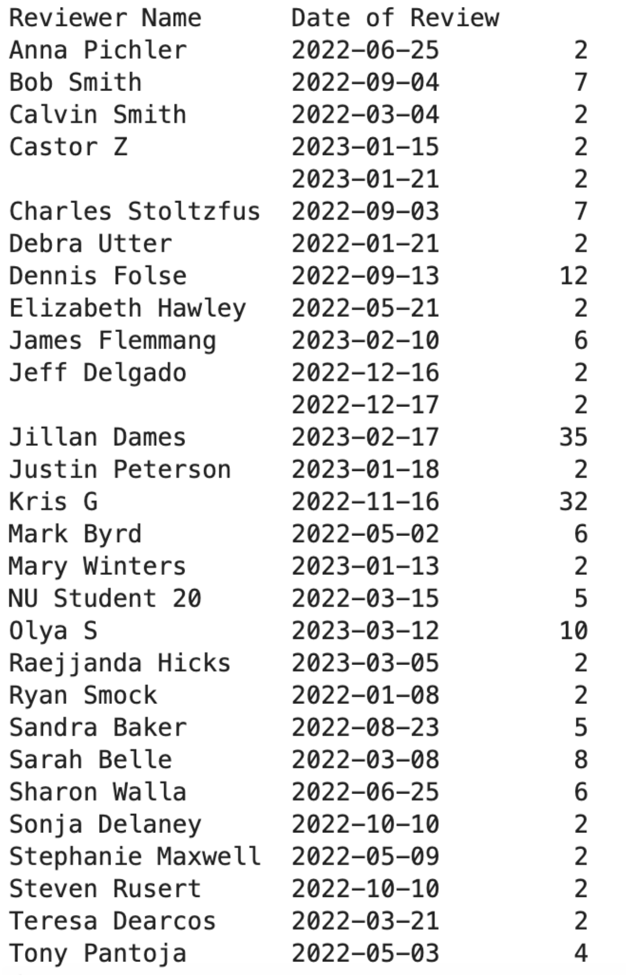
4. Eight people have experienced changes in their **Has Children?** status: **Bethany Carlson**, **David Smith**, **Jeff Delgado**, **Jillian Dames**, **Joyce Xu**, **Max Tomillo**, **Sandra Baker**, and **Solomon M**.

5. Further we find that there are two people called **David Smith** with exact different demographic data, which maybe they are two people or the reviewer are randomly enter his information.



	Reviewer Name	Restaurant Name	Rating	Review Text	Date of Review	Birth Year	Marital Status	Has Children?	Vegetarian?	Weight (lb)	Height (in)	Average Amount Spent	Preferred Mode of Transport	Northwestern Student?
289	David Smith	Soban Korea	5	NaN	2022-06-14	1993.0	Single	No	NaN	124.0	180.0	Medium	Car Owner	No
288	David Smith	Kilwin's	5	Absolutely love this place. They sell handmade...	2022-04-17	1976.0	Married	Yes	NaN	275.0	191.0	High	Car Owner	No

6. Also we find that **Jillan Dames** rate for 35 restaurant at **2023-02-17** and **Kris G** rate for 32 restaurant at **2022-11-16**.



```
Reviewer Name      Date of Review
Anna Pichler       2022-06-25         2
Bob Smith          2022-09-04         7
Calvin Smith       2022-03-04         2
Castor Z           2023-01-15         2
                   2023-01-21         2
Charles Stoltzfus  2022-09-03         7
Debra Utter        2022-01-21         2
Dennis Folse       2022-09-13        12
Elizabeth Hawley   2022-05-21         2
James Flemmang     2023-02-10         6
Jeff Delgado       2022-12-16         2
                   2022-12-17         2
Jillan Dames       2023-02-17        35
Justin Peterson    2023-01-18         2
Kris G             2022-11-16        32
Mark Byrd          2022-05-02         6
Mary Winters       2023-01-13         2
NU Student 20      2022-03-15         5
Olya S             2023-03-12        10
Raejjanda Hicks    2023-03-05         2
Ryan Smock         2022-01-08         2
Sandra Baker       2022-08-23         5
Sarah Belle        2022-03-08         8
Sharon Walla       2022-06-25         6
Sonja Delaney      2022-10-10         2
Stephanie Maxwell  2022-05-09         2
Steven Rusert      2022-10-10         2
Teresa Dearcos     2022-03-21         2
Tony Pantoja       2022-05-03         4
```