Jed Isom

Data Mining Capstone

October 3, 2015

## Overview

The University of Illinois – Urbana Champaign Data Mining Specialization is utilizing a real world data set from Yelp! to teach it’s students how to analyze data. This data has information about the businesses, the reviews for the businesses, the users who rate businesses, etc. In tasks 4 and 5, we were asked to take a list of dishes previously mined from the dataset in prior tasks to create recommendations for (task 4) other dishes to try within a given cuisine and (task 5) restaurants to try if you want to try a specific dish in a cuisine

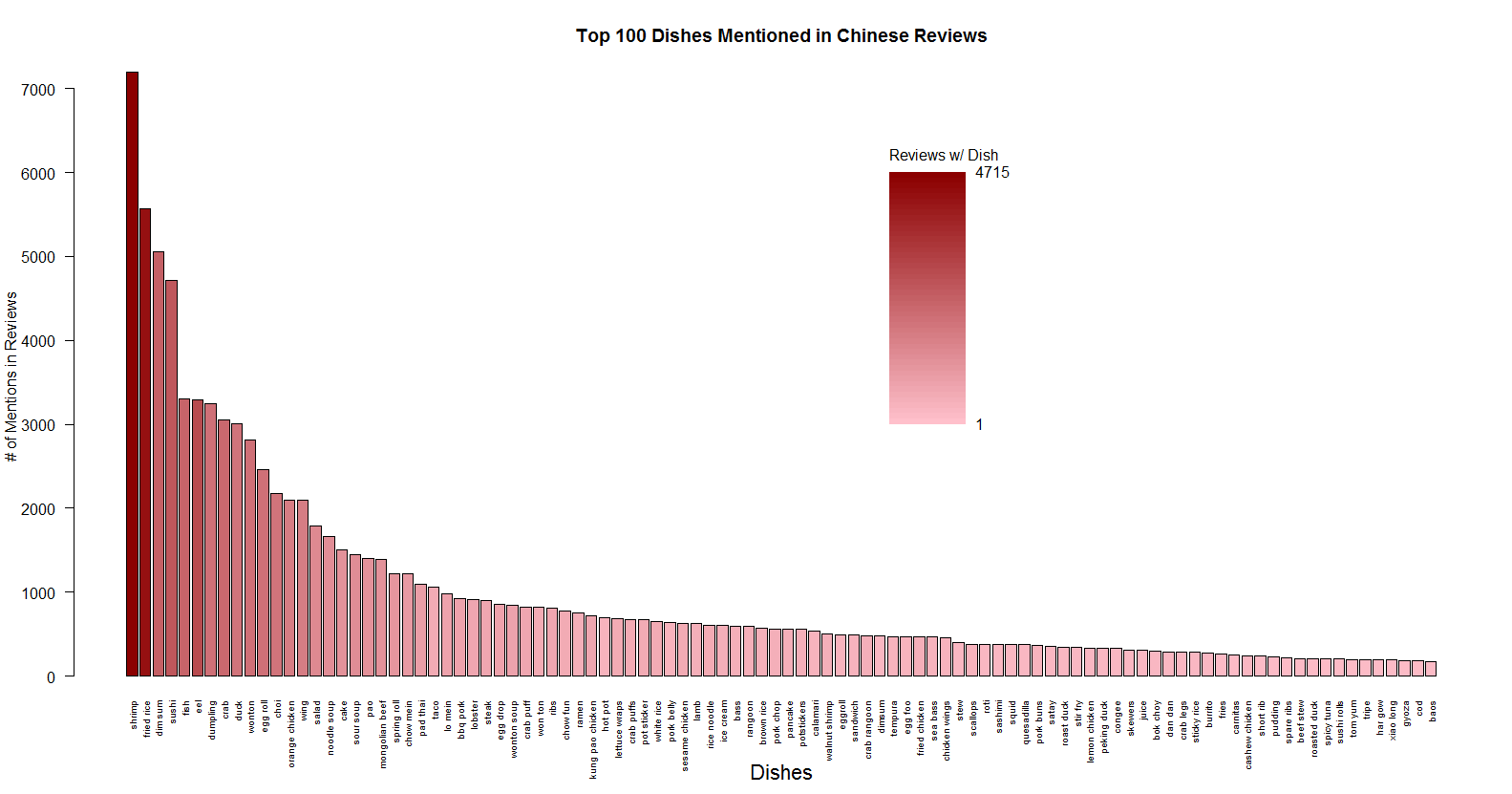
## Task 4

As mentioned above, task 4 was all about creating a “ranking of the dishes for a Yelp cuisine of [my] choice”. Since I worked with the Chinese cuisine in the prior task, I decided to continue to work with it here. To prepare the dataset to create a recommended dish in Chinese, I took the following steps in the R scripting language (see appendix I for the entire code for task 4 and 5):

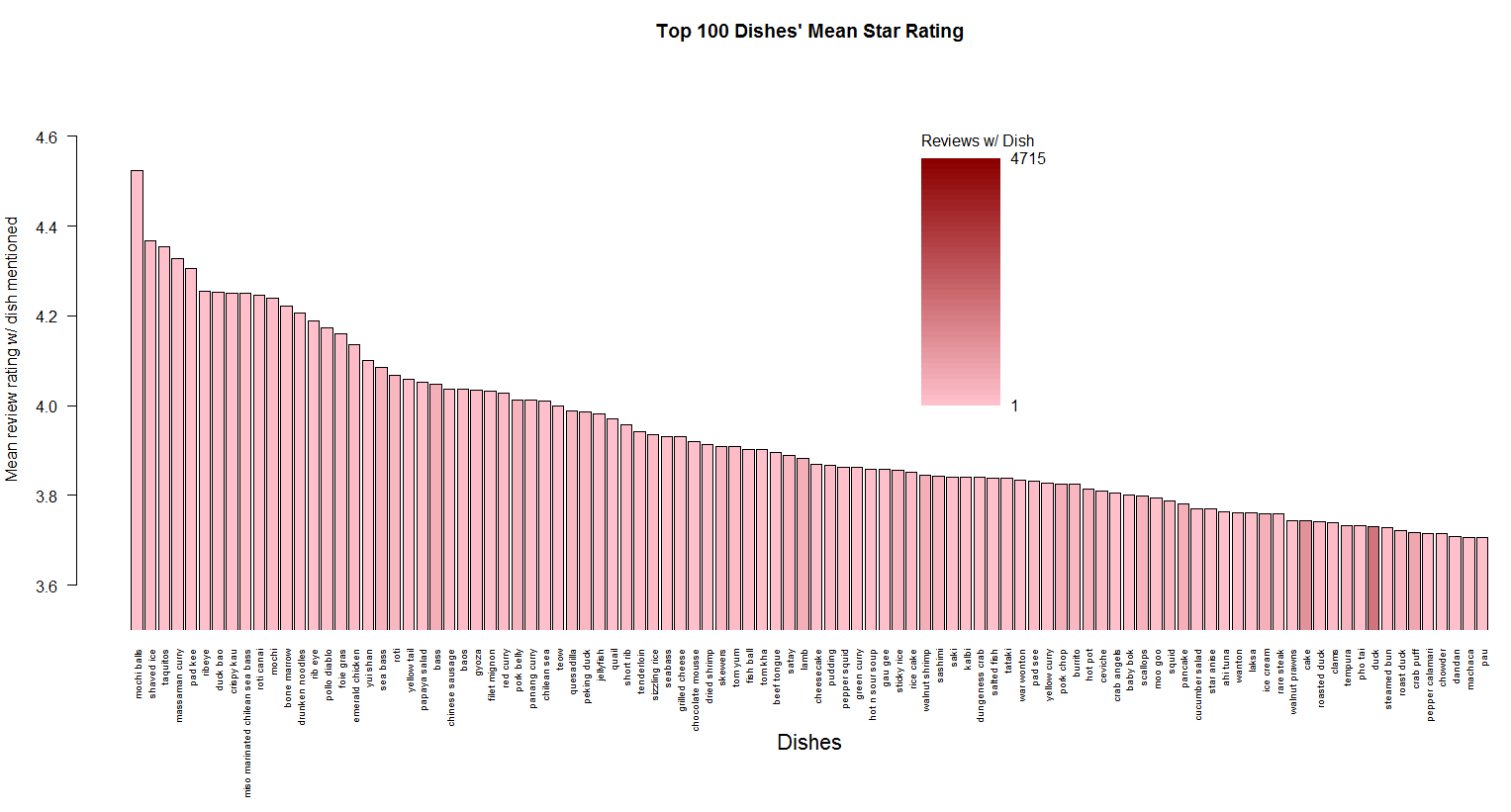
*Create table with scores for dishes and businesses in the Chinese cuisine*

1. Read in the JSON file into a variable called “review” and “business”
2. Filter the business table by the “category” variable to only get restaurants that included “Chinese” as a cuisine.
3. Create a list Chinese cuisine reviews by filtering the review table by those businesses included in the “Chinese” cuisine list
4. Turn this list of Chinese reviews into a “corpus” in R. This makes it easier to clean up the text and do analysis in the future
   1. Remove numbers from the review text(s)
   2. Turn all text to lower case
   3. Remove punctuation
   4. Remove any extra whitespace
   5. (I did not remove stop words because they might be part of a dishes name, or use stemming because then a common dish name might not be found)
5. Read in the list of Chinese dishes from the previous task. In the previous task I allowed the algorithm to suggest dishes, but before I used this list in this task I manually reviewed the list to make sure everything in the list was actually a Chinese dish
6. Loop through all of the reviews in the Chinese cuisine list
   1. Count how many times all of the dishes appear in that review using a custom built function I created in R
   2. Determine which dishes were actually found in the review
   3. Tabulate this information for all of the dishes found
      1. Dish name
      2. The review ID
      3. The business ID
      4. The number of times the dish was mentioned in the review
      5. The star rating of the review
7. Finally I aggregated this data by dish so I got a table that included the number of reviews for each dish, average star rating for each dish and total number of mentions of each dish.

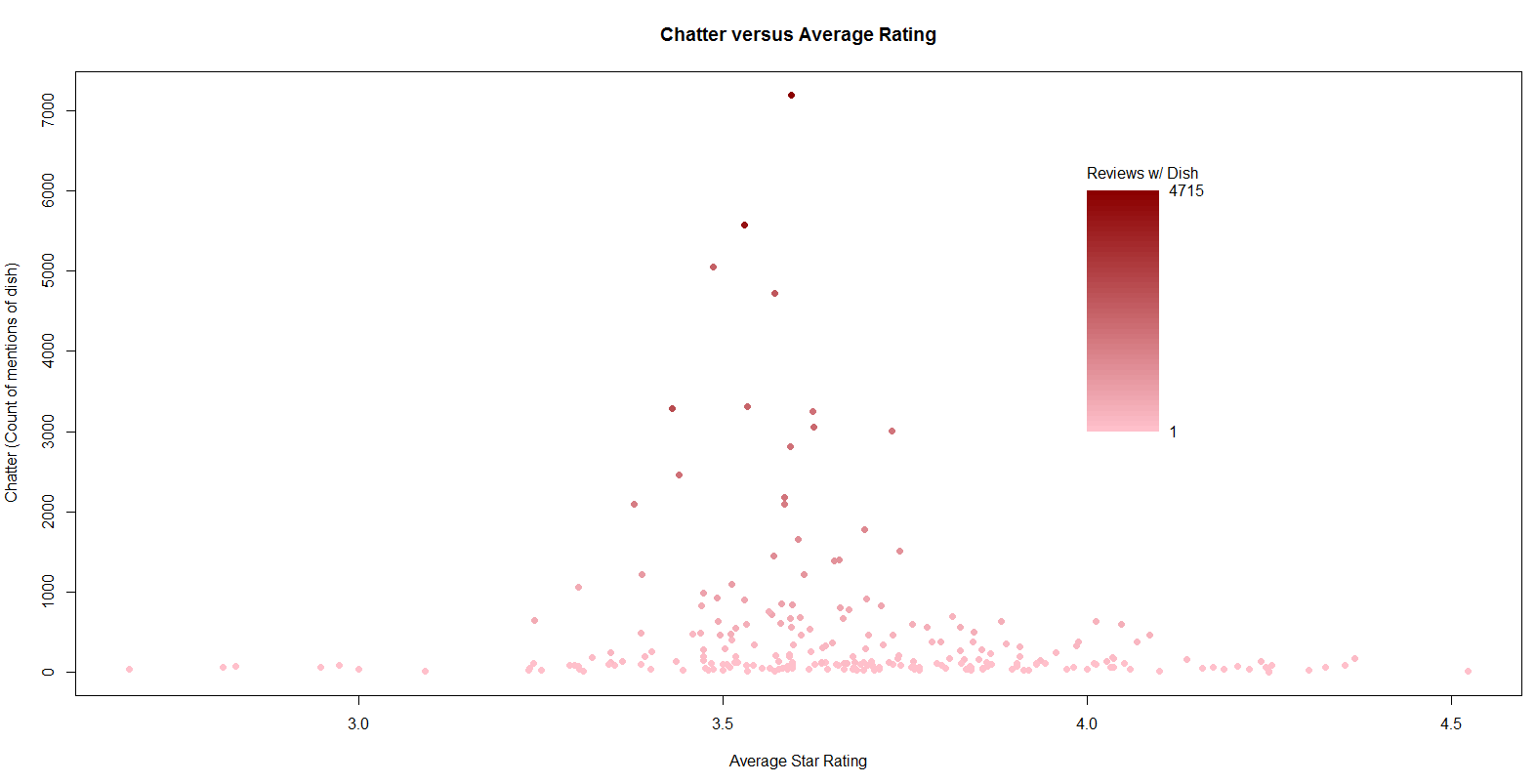
The simplest way to visualize this data would have been to just rank the dishes by how many reviews included them, but I decided to start with how many times the dish was mentioned in reviews and show the review count by using a color pallet as shown below (you may have to zoom in to see this chart effectively):



Just by looking at the visualization, it appears that the total number of mentions seems to be correlated with the number of reviews that contain each dish. This makes sense, and I’ll prove it to be true later. This doesn’t seem to be very helpful though in telling us which dish is the best tasting, just most talked about. That ‘talk’ could be positive or negative. So I also created a visualization for dishes based on the average review rating for reviews were each dish is mentioned. That one looks like this:



There does not appear to be a very strong correlation here between the average rating and the number of reviews that mention the dish. To confirm this, I created a scatter plot to visualize all of the dishes against these 3 dimensions



Not surprisingly (based on the central limit theorem of statistics) this plot looks a lot like a bell curve. I was curious about whether there was any correlation between these different ways of measuring the dishes so I used R to calculate the correlation coefficients as well as their p-values, which helped me determine if the correlation coefficients were statistically significant.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Review Count | “Chatter”  (total mentions of dish) | Average Star Rating |
| Review Count | 1 | 0.9830206 | -0.1460474 |
| “Chatter”  (total mentions of dish) | 0.9830206 | 1 |  |
| Average Star Rating | -0.1460474 | -0.1356369 | 1 |

*Rating Correlation Coefficients*

|  |  |  |  |
| --- | --- | --- | --- |
|  | Review Count | “Chatter”  (total mentions of dish) | Average Star Rating |
| Review Count |  | 0.000 | 0.0239 |
| “Chatter”  (total mentions of dish) | 0.000 |  | 0.0361 |
| Average Star Rating | 0.0239 | 0.0361 |  |

*P-values of Rating Correlation Coefficients*

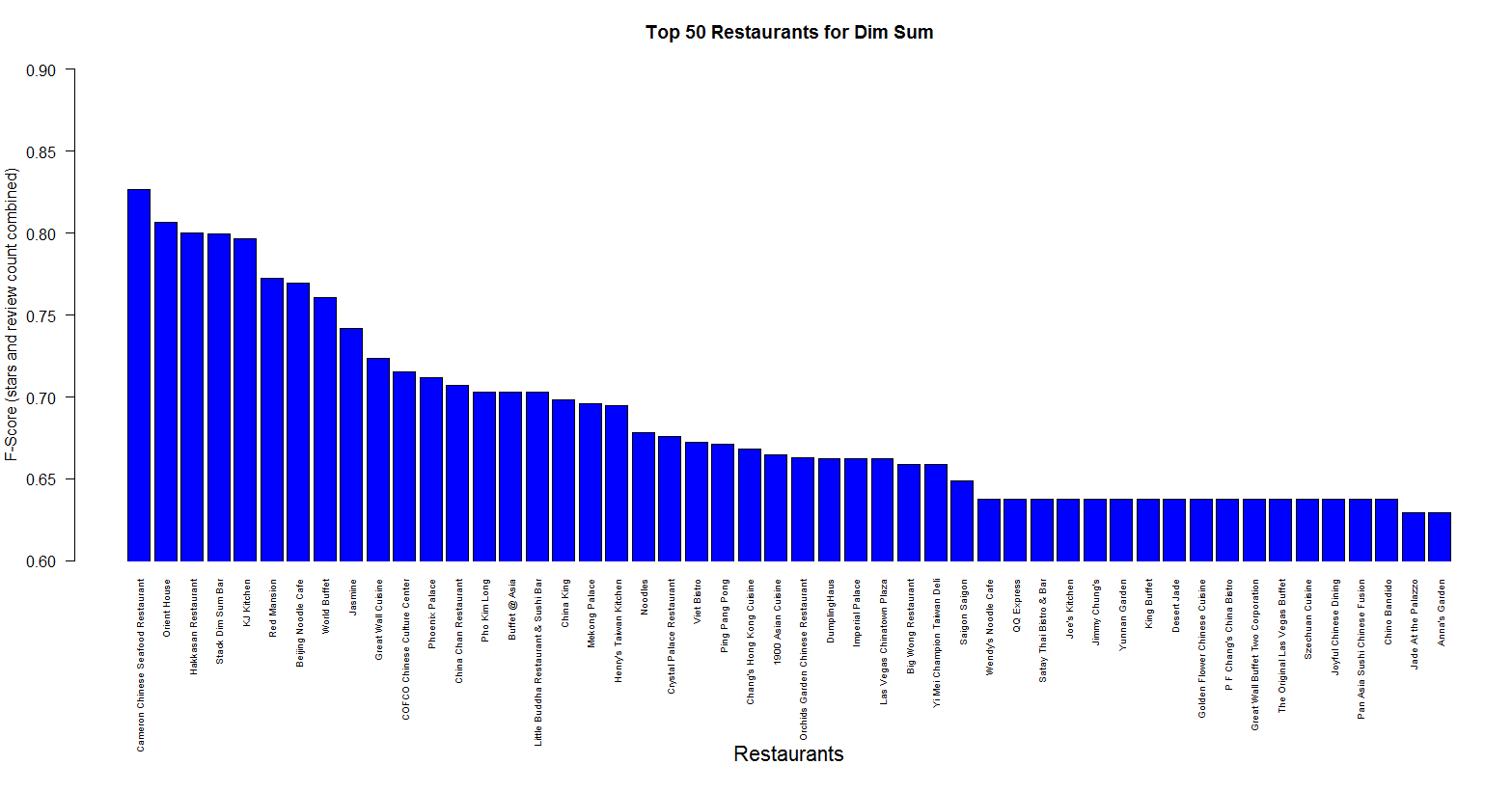
These 2 tables showed me that the review count and “chatter” rating methods were basically the same because of the strong correlation between the 2. The average star rating shows a slight negative correlation to the other 2 which is also significant. That means that using a combination of average star rating and simpler “review count”, might make for a good combined measurement system (“how popular?” AND “how good?”). I could have pursued this more in task 4, but because I knew I had task 5 to work on I decided to use this new knowledge there.

## Task 5

Task 5 the “goal [was] to recommend good restaurants to those who would like to try one or more dishes in a cuisine.” In the Chinese cuisine, I chose “dim sum” as the dish that I would study because it was included in a lot of reviews and would probably yield some interesting data. Plus, I haven’t really eaten dim sum, and thought I might learn something about where I might get some good dim sum.

With the knowledge I gained in task 4 about the review count and the average star rating I came up with the idea of using an F-score to balance them (similar to the one used in text retrieval applications to balance precision and recall). To do this I would need to normalize the review counts and average ratings to be values between 0 and 1. However, as I started working with this data though, I noticed that the counts of reviews for businesses, or dishes in this data set seem to be very stratified. Either there weren’t many reviews, or there were lots of reviews. If I were to plot this variable it would likely look exponential. To make the review count score more linear, I took the logarithm of this count first, and then normalized it. I did this because I didn’t want restaurants that might not be well known to be overly penalized. Maybe their new restaurants and they’re really great right? So in the end my review count score got normalized like this

The normalization of the average star rating was done the same way, except without the logarithm. I started by trying the standard F1 score that basically equally weighs the 2 metrics. I did this in Excel so I could play around with it quickly and figure out what worked. I quickly realized that the review count was still too influential in the final score. This was based on my judgement when looking at how the rankings turned out. I had to ask myself, would I prefer a restaurant with a lot of ratings and a lower rating, our a restaurant with less ratings but higher ratings? To fix what I say I used the more general form of F-score and found that when I set β = 3.0, the scores seemed to give good rankings. Here’s the final visualization I came up with for task 5



This chart looks very simple, but it was the result of a lot of trial and error and thinking about how a user would want these restaurants ranked if they wanted a recommendation for where to try a new cuisine like “dim sum”.