# task04&05-rank-dishes-and-restaurants

October 6, 2018

loganjtravis@gmail.com (Logan Travis)

## 1 Summary

From course page Week 4 > Task 4 and 5 Information > Task 4 and 5 Overview:

> The general goal of Tasks 4 and 5 is to leverage recognized dish names to further help people making dining decisions. Specifically, Task 4 is to mine popular dishes in a cuisine that are liked by people; this can be very useful for people who would be interested in trying a cuisine that they might not be familiar with. Task 5 is to recommend restaurants to people who would like to have a particular dish or a certain type of dishes. This is directly useful to help people choose where to dine.
>
> ...
>
> **Instructions** Some questions to consider when working on Tasks 4 & 5:
>
> 1. Given a cuisine and a set of candidate dish names of the cuisine, how do we quantify the popularity of a dish? How can we discover the popular dishes that are liked by many reviewers? What kind of dishes should be ranked higher in general if we are to recommend dishes of a cuisine for people to try? Would the number of times a dish is mentioned in all the reviews be a better indicator of a popular dish than the number of restaurants whose reviews mentioned the dish?
> 2. For people who are interested in a particular dish or a certain type of dishes, which restaurants should be recommended? How can we design a ranking function based on the reviews of the restaurants that mention the particular dish(es)? Should a restaurant with more dish name occurrences be ranked higher than one with more unique dish names?
> 3. How can you visualize the recommended dishes for a cuisine and the recommended restaurants for particular dishes to make them as useful as possible to users? How can the visualization be incorporated into a usable system? For example, you can imagine using the algorithms you developed for Tasks 4 and 5 to construct a system that allows a user to select a cuisine to see the favorite/popular dishes of the cuisine and further recommends the best restaurants if a user selects a particular dish or a set of dishes that are interesting to him/her.

## 2 A Note on This Report

I hid much of my code displaying only chunks that clarified my process. My previous reports exceeded 15 pages, mostly Python code. Reviewers suggested replacing code with written descriptions for clarity.

## 3 Task 04: Mine Popular Dishes

I chose to investigate popular dishes for Mexican cuisine. I use the list of Mexican dishes generated for task 3. It includes both dish names mined from frequent phrases and a list of dishes obtained from Wikipedia - List of Mexican dishes.

I make only one change to that previous list: Stip accents from characters. Other task 3 reports I read took that extra step for cuisines with non-English characters like Mexican and Chinese. It improved their frequent phrase mining and I anticipate it will help me to investigate popular dishes.

### 3.1 Get Dishes for Mexican Cuisine

```
In [6]: # Get dishes for Mexican cuisine
        dfMexDishes = pd.read_csv(PATH_SOURCE_MEXICAN_LABELS, names=["dish"])
```

```
In [7]: # Strip accents from dish names
        dfMexDishes.dish = dfMexDishes.dish.transform(unidecode.unidecode)
```

### 3.2 Rank Dishes by Frequency in Reviews (simple)

I first rank dish popularity by the frequency of their appearance in Yelp reviews for Mexican restaurants. This method - though simple - provides an excellent base line for subsequent improvements.

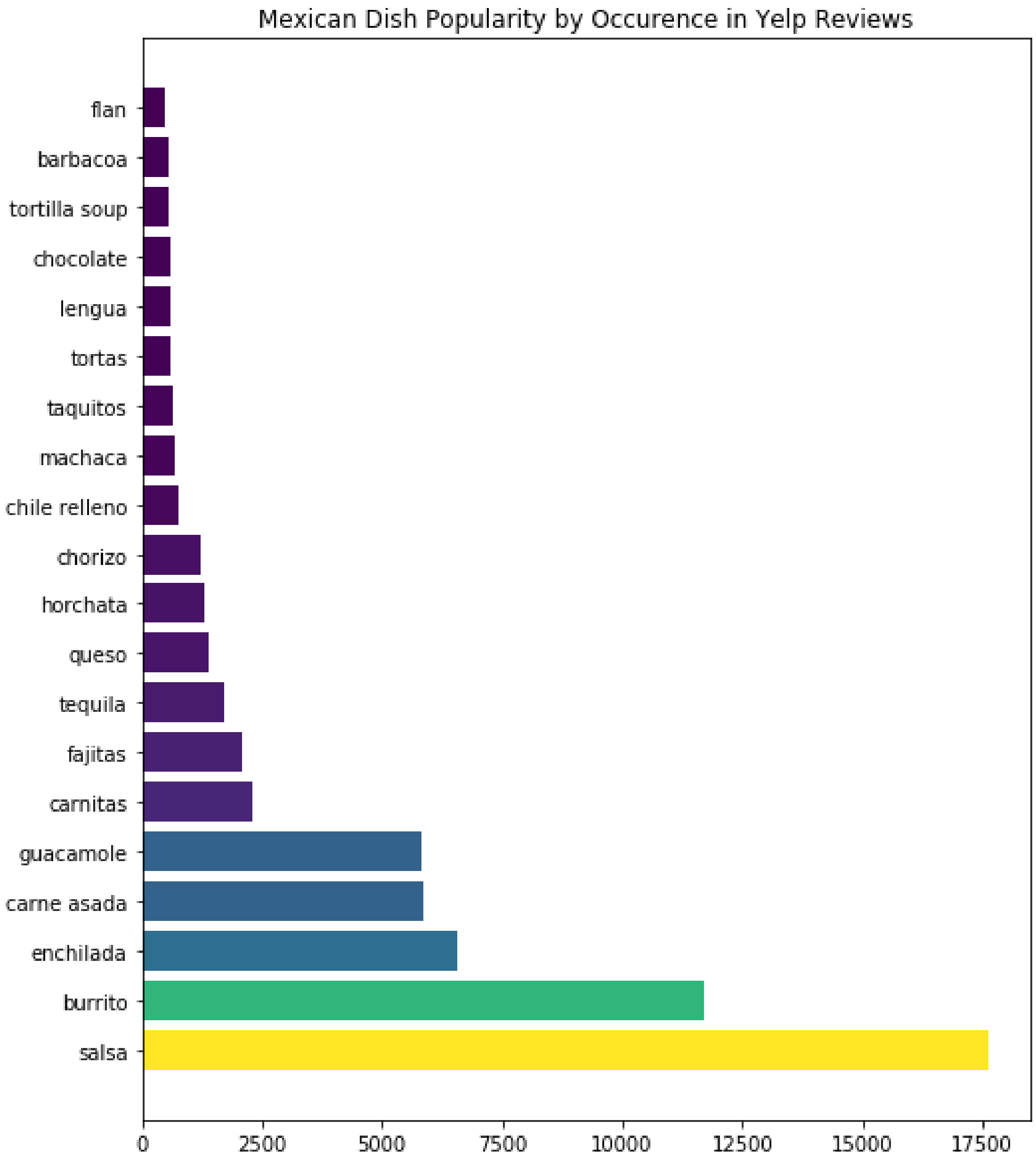
Note: I chose to count only the first appearance of a dish in each review (essentially a binary is/is not present in review). Counting all the apperances of a dish would introduce more reviewer sentiment than I want for this initial ranking.

```
In [15]: %%time

         # Calculate dish frequencies
         dishInReview = dishFreq.fit_transform(dfYelpReviews.text)

CPU times: user 1min 32s, sys: 156 ms, total: 1min 33s
Wall time: 1min 34s
```

```
In [19]: # Show plot
         _, _ = myPlot()
         plot.show()
```



The chart above displays the top 20 Mexican dishes (out of 242) with popularity measured by counting the number of reviews in which a dish appears. Staple dishes like “salsa”, “burrito”, and “enchilada” appear most frequently. However, I appreciate seeing less well known but more representive dishes like “chille relleno”, “tortas”, and “lengua”. They rank lower than the top-ranked “salsa” by an order of magnitude. I would still include them on a list of top dishes for those seeking to experience Mexican cuisine.

The simple count of appearance in reviews has one potential flaw: It follows a roughly exponential curve with an extremely long tail. I suspect counting dish appearance *without* adjusting for positive/negative review over represents common dishes.

### 3.3 Rank Dishes by Review Stars

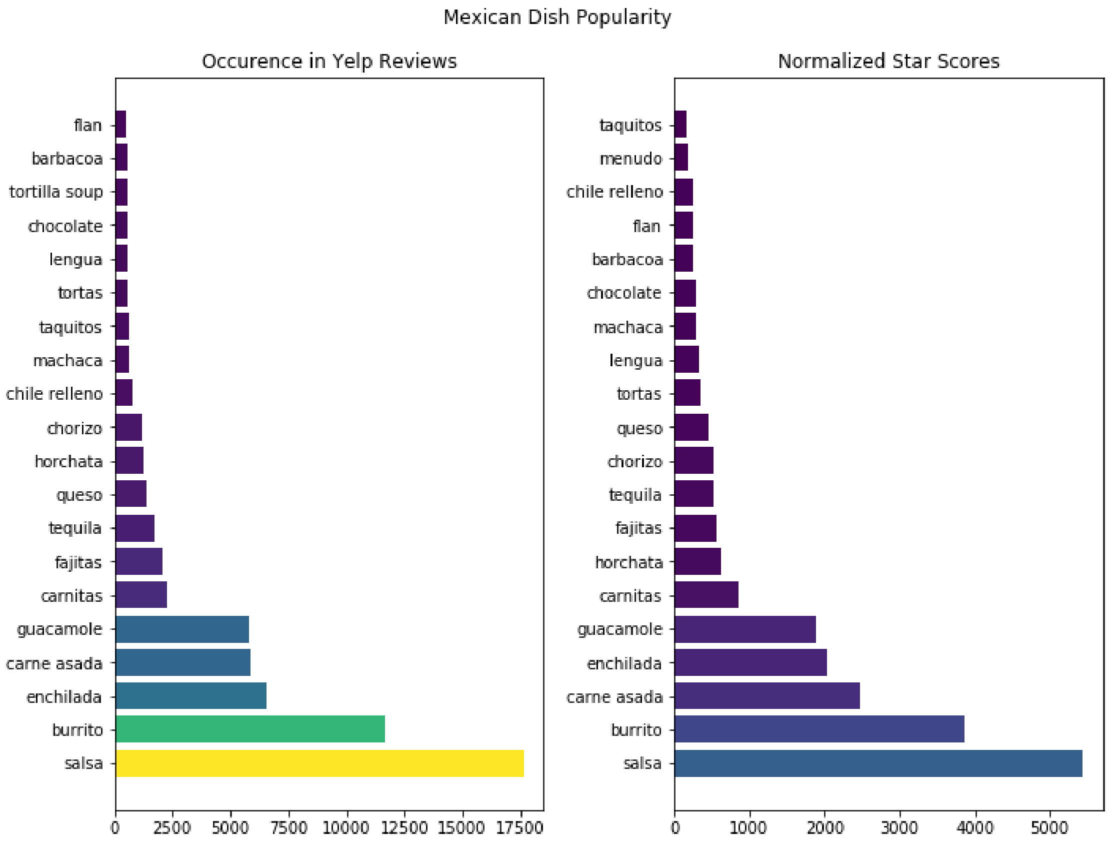
Adjusting the popularity measure for review “stars” may improve on the simple count of dish appearance in reviews. It has its own potential shortcoming. A reviewer might only award one star for poor service despite excellent food. However, hypothesizing that most reviewers weight their appreciation of the food higher than other factors will provide a useful comparsion to the simple count.

Note: I changed the Yelp [1, 5] star scale to a [-1, 1] scale. I wanted negative and positive reviews to offset eachother as way to reduce representation of common dishes.

```
In [20]: %%capture --no-stdout

         # Normalize star scores; Note: Captures `DataConversionWarning`
         # from int64 to float64
         minMax = MinMaxScaler((-1, 1))
         normedStars = minMax.fit_transform(dfYelpReviews.stars.values.reshape(-1, 1))
```

```
In [25]: # Show plot
         _, _ = myPlot()
         plot.show()
```



Side by side comparions of the simple count versus the [-1, 1] star score shows a couple improvements:

- Though still exponential, the star score graph grows more slowly. I would feel comfortable recommended dishes below the top three as “popular”.
- The maximum range dropped by half. Clearly not all mentions of a dish indicate appreciation.

### 3.4 Conclusion

Though I did not have time for this assignment, refining the postive/negative sentiment for the *dish* would improve the popularity measure further. The star score mixes many metrics including food, service quality, ambiance, location. I would start replace the star score with a sentiment measure for the entire review. I would then narrow the sentiment analysis to “word window” around the dish in each review.

## 4 Task 05: Restaurant Recommendation

Recommending popular dishes to someone wanting to experience Mexican cuisine is just a first step. They will want to taste *good* versions of those dishes. A bad burrito - speaking from experience - will turn anyone off Mexican cuisine!

### 4.1 Rank Restaurants by Dish Frequency In Reviews

I began with a simple restaurant rank by dish repesentation in each restaurant’s reviews. Pick a dish - I chose “burrito” as an example - and the list will display restaurants with a higher proportion of reviews mentioning that dish. Note “higher proportion”. I normalized the number of times a dish appeared in a review against the total number of reviews for a given restaurant. That enabled comparison between restaurants. It also introduced a problem I resolve in my next iteration: Restaurants with disproportionatly few reviews.

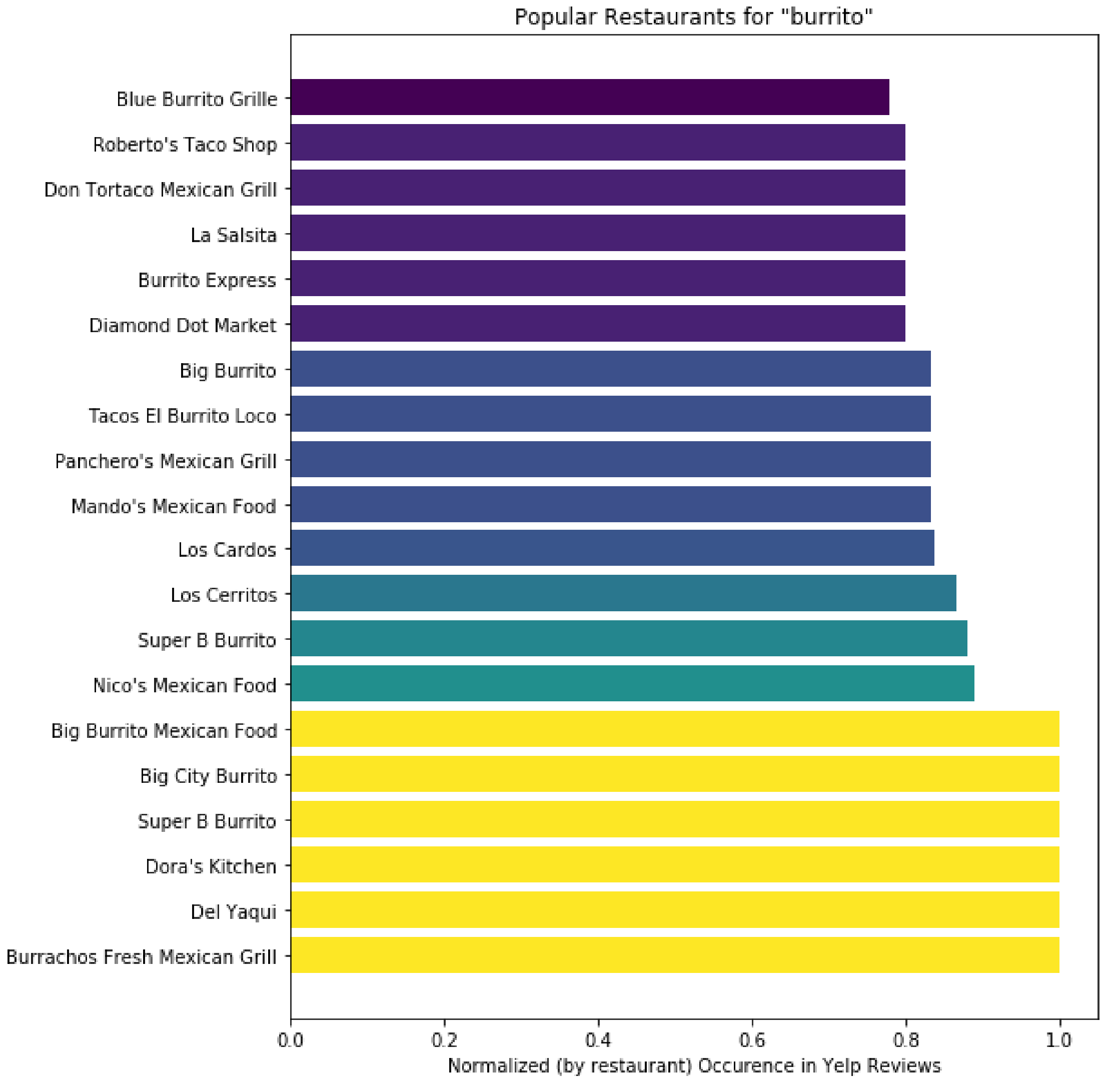
```
In [29]: %%time

         # Sum dish appearances in reviews across restaurants
         dishCountRestaurant = sp.sparse.coo_matrix((0, dishInReview.shape[1]))
         for b in dfYelpBusinesses.business_id:
             idxs = dfYelpReviews[dfYelpReviews.business_id == b].index
             dishCountRestaurant = sp.sparse.vstack([dishCountRestaurant, \
                                                     dishInReview[idxs, ].sum(axis=0)])
```

```
CPU times: user 10.5 s, sys: 0 ns, total: 10.5 s
Wall time: 10.6 s
```

```
In [31]: # Normalize by number of reviews for each resaurant
         dishReviewRatioRestaurant = dishCountRestaurant.multiply(\
                 1.0 / dfYelpBusinesses.my_review_count.values[:, np.newaxis])
```

```
In [34]: # Show plot
         _, _ = myPlot()
         plot.show()
```



Immediately the low review problem appears. Six restaurants mention "burrito" in *every* review. Several of those - "Super B Burrito" and "Big City Burrito" as examples - likely serve only burritos; every review should mention that dish. They could make great burritos. However, I cannot confidently use this simple count to recommend restaurants to someone wanting to try his/her first burrito.

## 4.2 Rank Restaurants by Dish Review Stars

Ranking restaurants by their review star scores should improve the recommendation. I used the same [-1, 1] scale from ranking dishes by their reviews. Note that some of the issues for dishes do not impair using star scores for restaurants: I want to recommend restaurants not just for their food but also for their ambience, service quality, etc. Those seeking an introduction to Mexican cuisine want a good *overall* experience not just great food.

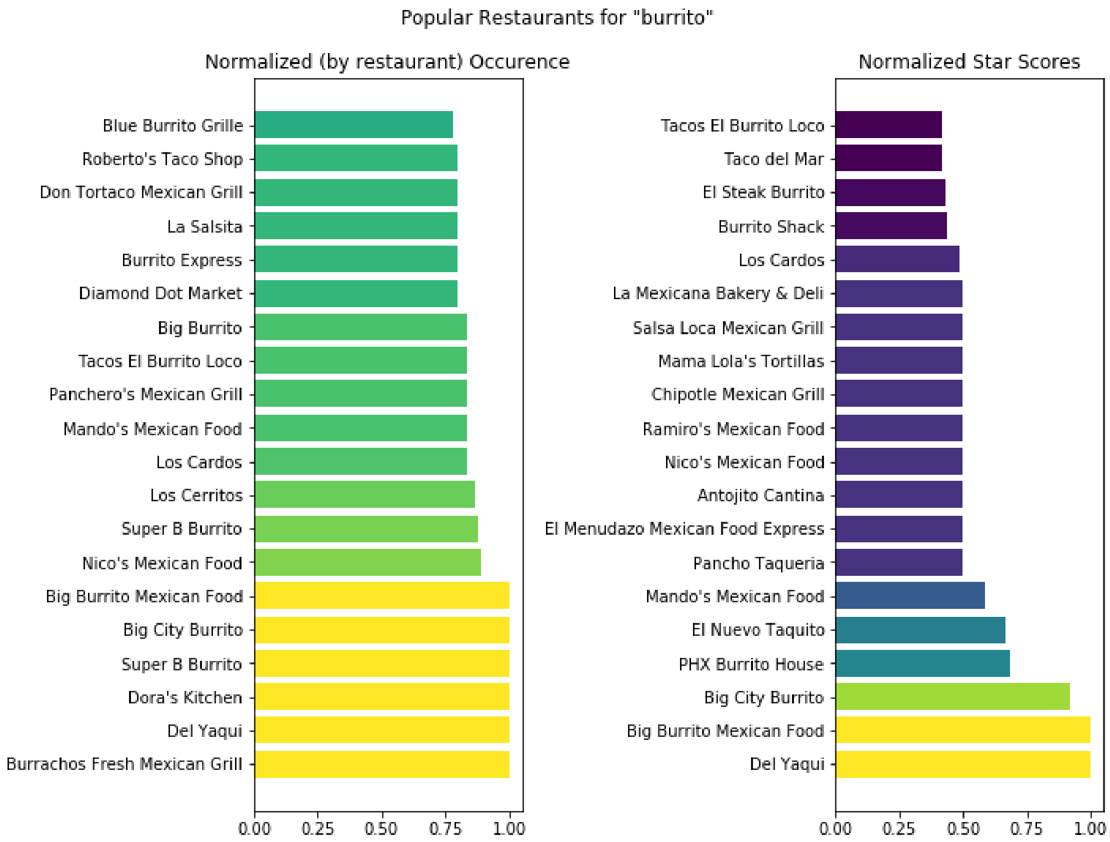
```
In [35]: %%time

         # Sum dish normalized stars across restaurants
         dishNormedStarsRestaurant = sp.sparse.coo_matrix((0, dishNormedStars.shape[1]))
         for b in dfYelpBusinesses.business_id:
             idxs = dfYelpReviews[dfYelpReviews.business_id == b].index
             dishNormedStarsRestaurant = sp.sparse.vstack([dishNormedStarsRestaurant, \
                                                          dishNormedStars[idxs, ].sum(axis=0)])
```

```
CPU times: user 10.7 s, sys: 15.6 ms, total: 10.7 s
Wall time: 11 s
```

```
In [37]: # Normalize by number of reviews for each resaurant
         dishStarRatioRestaurant = dishNormedRestaurant.multiply(\
                 1.0 / dfYelpBusinesses.my_review_count.values[:, np.newaxis])
```

```
In [39]: # Show plot
         _, _ = myPlot()
         plot.show()
```



Comparing the simple count to the [-1, 1] star score shows improvement:

- Fewer restaurants scored a perfect 1.0. Those that do now represent both good restaurants and good restaurants for burritos. I would recommend them without hesitation.
- Overall scores decreased. This better fits my real-world experience. Most Mexican restaurants offer burritos but few offer good burritos. Those that do usually specialize in just burritos.

The [-1, 1] star score graph would also work well in an recommendation system:

- It presents an interpretable list of restaurants for a specific dish
- It requires litte storage; my matrix of all Mexican restaurants and dishes had a sparsity score of 96.8%
- Several sparse matrix implementations offers fast access to all restaurant scores for a specific dish

### 4.3 Compare Top Dishes and Top Restaurants

Though I think the [-1, 1] star score graph would work well in a recommendation system, users will need a higher-level starting view that just one dish. I chose "burrito" as an example because of my own experience. I know the dish. Were I to try a new cuisine - say Ethiopean - I could not choose one dish to start.

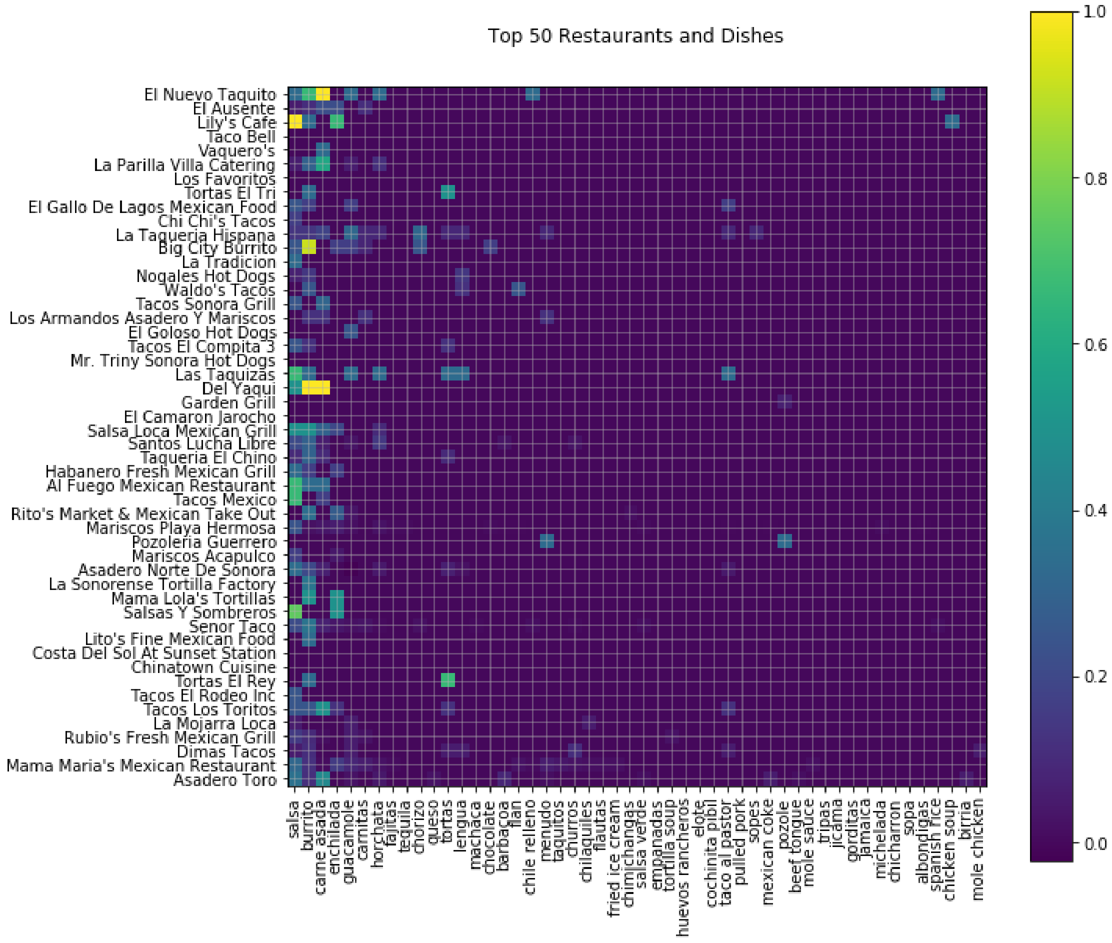
The earlier graph of popular dishes might work. I could pick three or four dishes to investigate for good restaurants. Seeing both popular dishes *and* restaurants might improve my experience even more.

```
In [40]: # Set top N for N restaurant by N dish matrix
         TOP_N = 50
```

```
In [41]: # Get top restaurants by star scores
         topRest = dfYelpBusinesses.nlargest(TOP_N, "stars")
         topRestIdx = topRest.index.values
```

```
In [42]: # Get top dishes by star scores
         topDish = dfMexDishes.nlargest(TOP_N, "score_in_reviews")
         topDishIdx = topDish.index.values
```

```
In [44]: # Show plot
         _, _ = myPlot()
         plot.show()
```



## 4.4 Conclusion

Perhaps viewing both popular dishes and restaurants makes a poor starting point for exploring a new cuisine. The top 50 restaurants (by average star score) and dishes (by summed [-1, 1] star score in reviews) shows few restaurants offering multiple good dishes. Someone exploring Mexican cuisine for the first time would need to visit multiple restaurants to taste the best versions of top dishes.

This *might* represent ground truth. I have eaten enough burritos to know most restaurants do not make a good version. However, I would not use the chart above in a recommendation system. Users expect a few, obvious choices. More simplification - using a restaurant's star score rather than summing only scores from reviews in which a dish appears - would clarify choices.