Data Mining Capstone Project Report

Overview

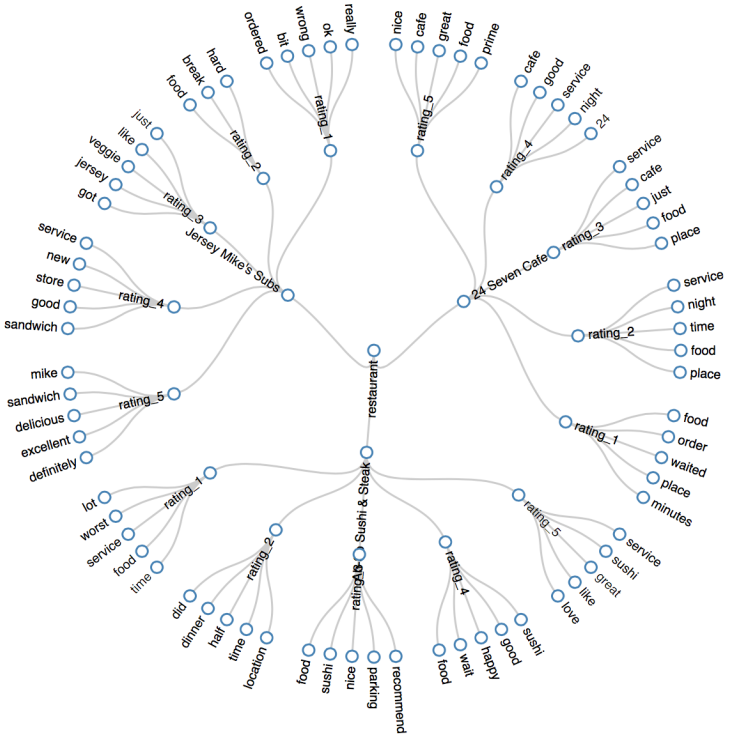
The primary goal of the Capstone has been to apply all the learned knowledge and skills from the Data Mining Specialization to solve real-world data mining problems.

Project Summary

For this capstone project, there are 6 tasks performed to explore the given Yelp dataset.

Task1 – Exploration of data set

Use a topic model (e.g., PLSA or LDA) to extract topics from all the review text (or a large sample of them) and visualize the topics to understand what people have talked about in these reviews. Then compare topics from two subsets to help understand the similarity and differences between these topics.



Left graph gives a view of the popular topics of three restaurants, and classified them by rating so that we can compare same rating topics from different restaurants.

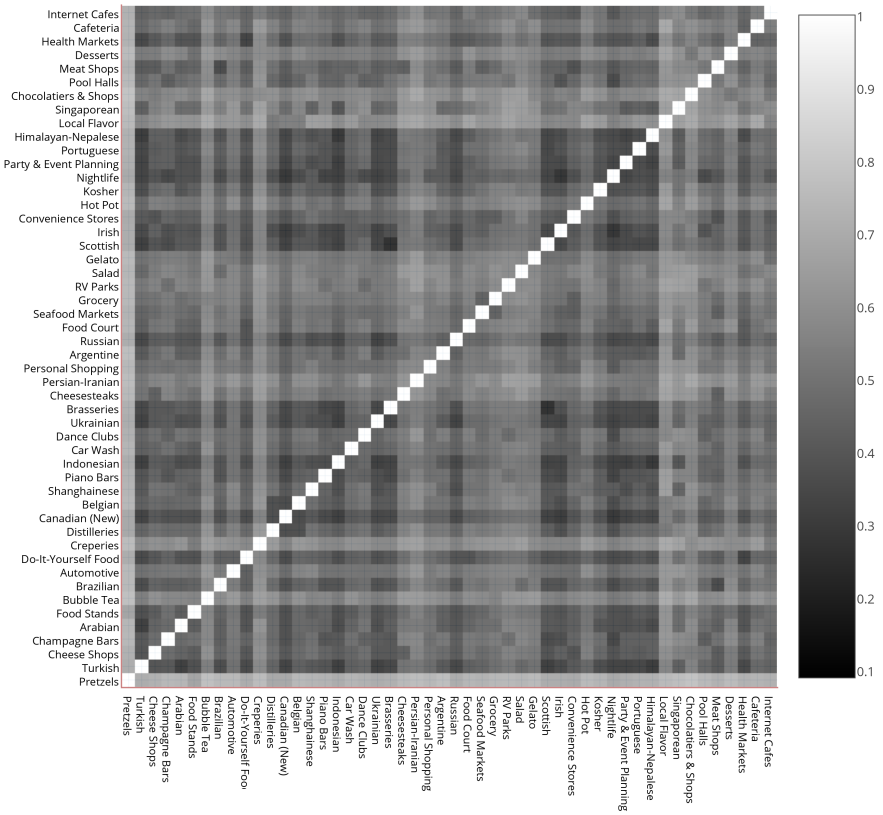
From this task, I got a general view of the data set that I'll be using for the coming weeks, and practiced how to process necessary information from JSON file, also got familiar with D3 library.

Task2 – Cuisine clustering and map construction

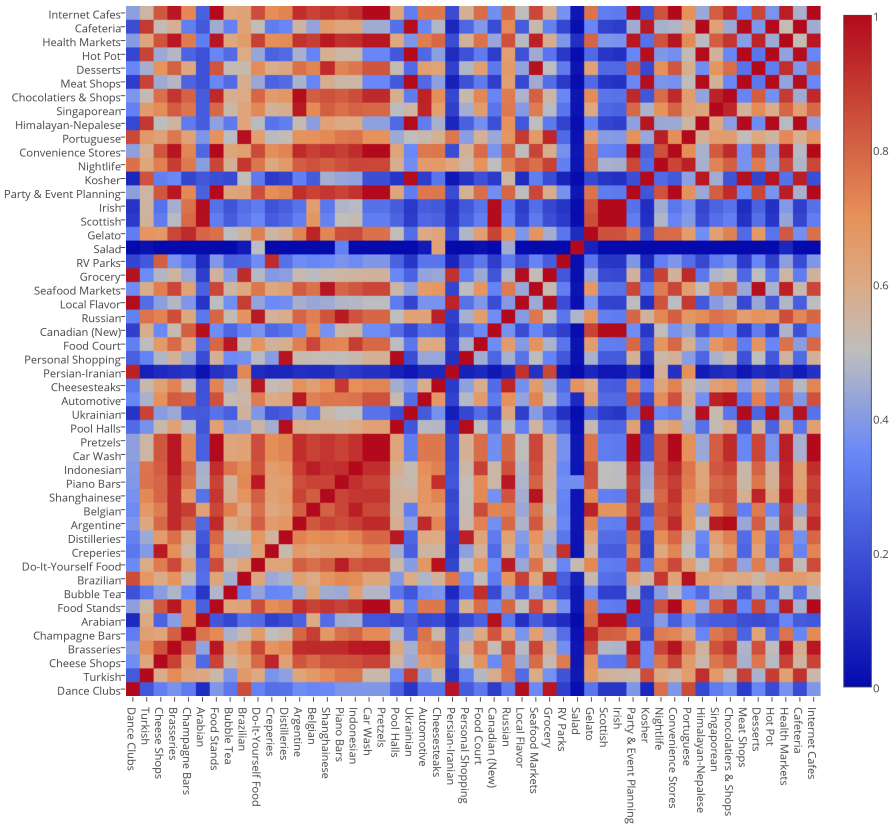
The cuisine map could visually help the users understand what cuisines are available and their relations, which allows for the discovery of new cuisines, thus facilitating exploration of unfamiliar cuisines.

From the right graph, we can get the cosine similarity between each two cuisines, which not only helps us find potential connection between dishes but also could serve as a reference for future recommendation.

task2.1_cuisine_similarity



task2.2_similarity_LDA_TFIDF



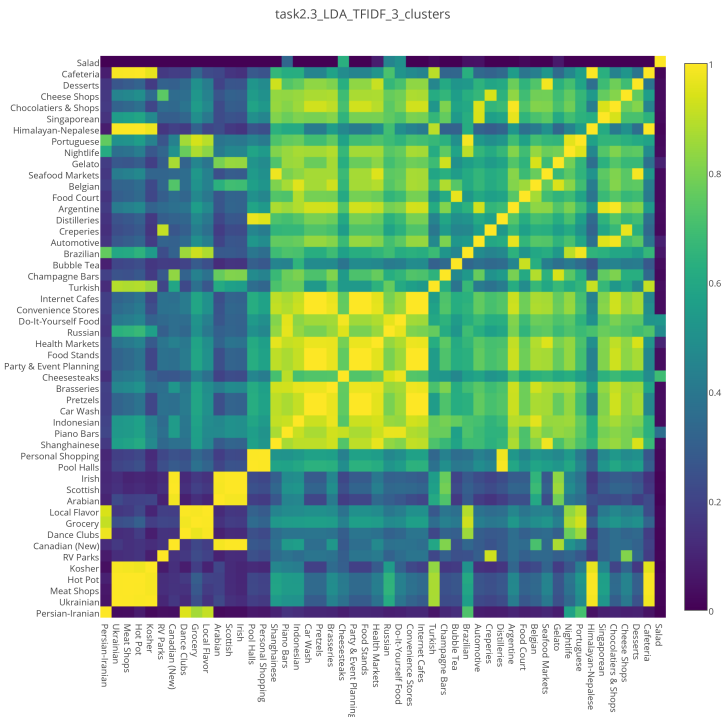
In addition, such similarity map based on tags could be improved by:

- Varying text representation
- Computing the review similarity
- Cluster analysis

By concatenating all the reviews of a cuisine to get text representation, I applied LDA model to extract the similarity between them.

The left graph shows a bunch of cuisines can be clustered comparing with above cuisine similarity graph.

Also different clusters will provide us more detailed information about which



cuisines can be grouped together, the more cluster, the more likely we'll get closely related cuisines.

Task3 – Dish recognition

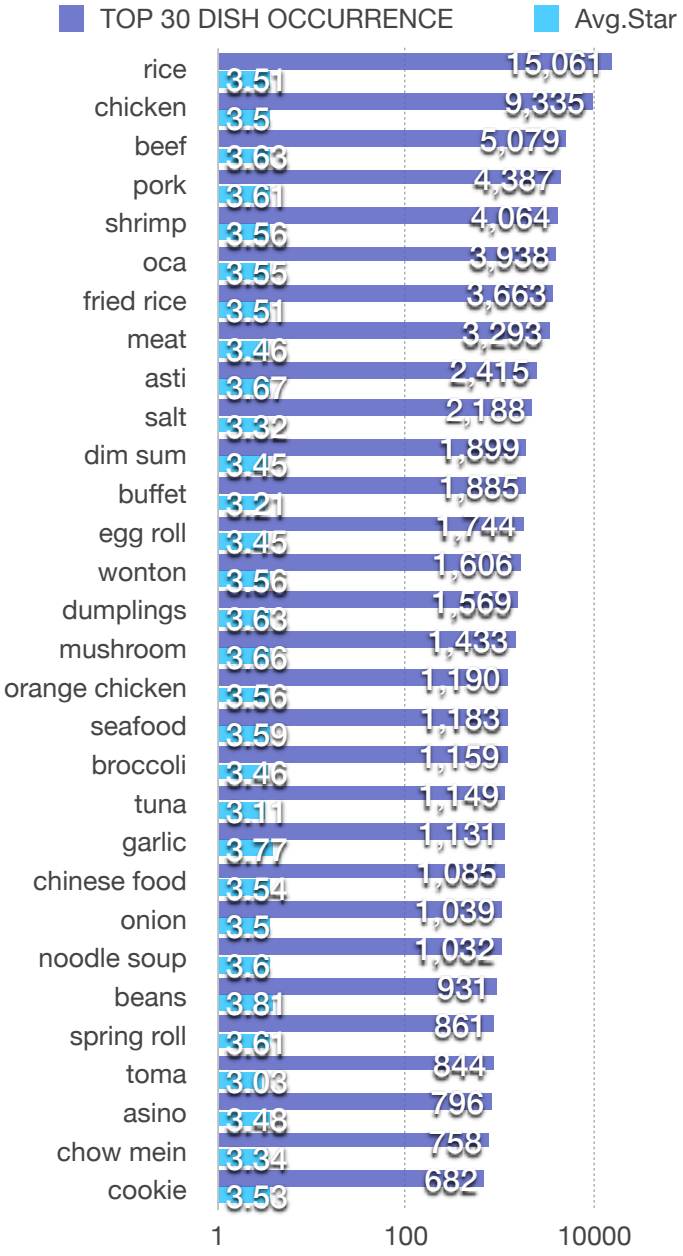
For this task, I tried to explore common/popular dishes of Chinese cuisine from customer reviews.

By manual annotating, I managed to remove some false-positive non-dish name phrase and change false negative dish name phrase to positive label. Through powerful mining tools like SegPhrase and ToPMine, I was able to find some missing dish names and complete the Chinese cuisine dish set, which has been reused in the coming tasks.

Task4 – Popular dishes

Mining popular dishes among people is a difficult task since the judgment can be made by various criteria. Visualised ranking gives a clear view of dishes from different perspective, thus could provide a more comprehensive introduction to people who wish to try new dish.

I mainly applied statistical knowledge to this task.



Left graph ranks the top 30 most popular dishes according to their occurrence in customer reviews.

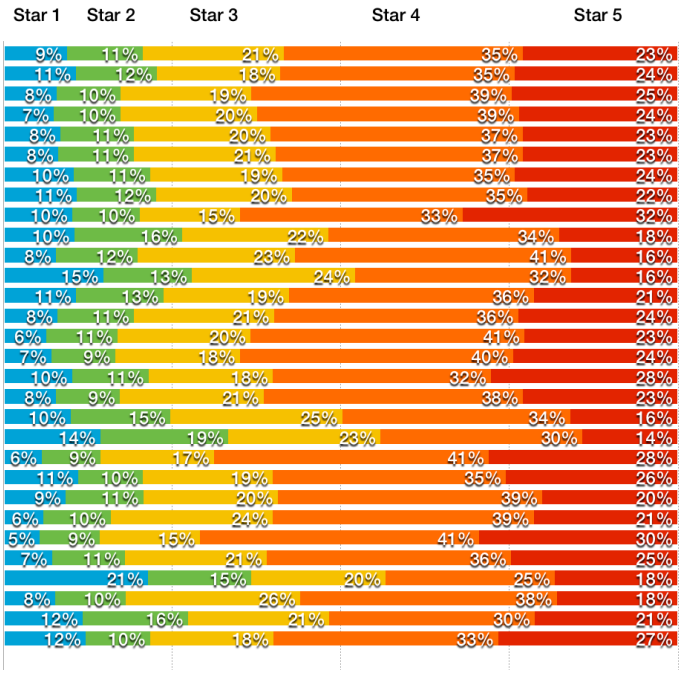
In many cases, occurrence is a good indicator of popularity, the more a dish mentioned by people the more popular it could be.

But sometimes, occurrence does not tell the whole story, average rating could provide a different perspective since both positive and negative reviews could contribute to the overall occurrence, but average rating will show the general feeling towards a dish.

Still, occurrence and average rating may not reflect the real situation since both of them can not tell the distribution or attitude of reviews.

Therefore, rating distribution, as shown on the right, can reflect the proportion of positive, neutral and negative reviews.

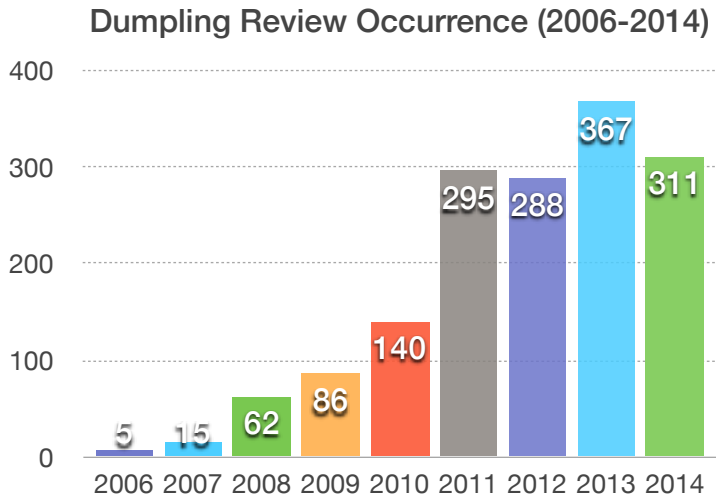
In rare cases, the average rating of a dish could be high, however the distribution might be polarised where only favour and dislike reviews exist, such case should be considered carefully since those favoured or disliked ones could come from people with something important to say.



For instance, 'buffet' is mentioned 1885 times with 3.21 average rating, but from rating distribution, we can find that there are 15% of reviews are 1-star, which means there's something that a mount of people doesn't like.

A detailed case study about "dumpling" could give us more information. I classified the reviews according by date, then collect all the reviews within each year.

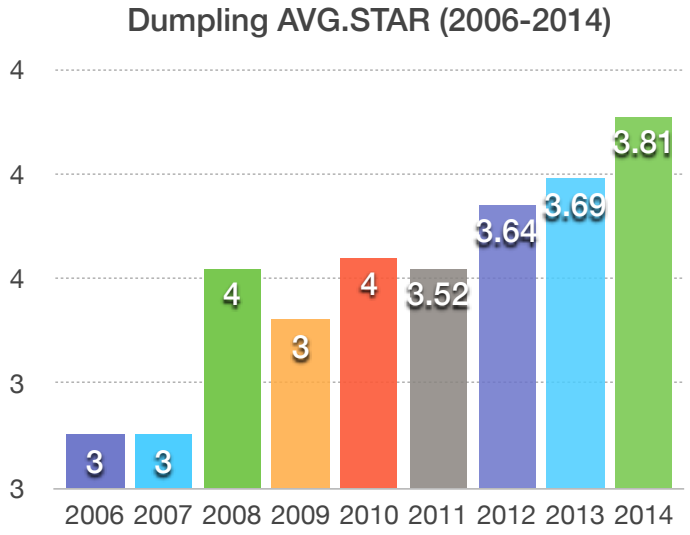
From the below graph we can find that dumpling related reviews are growing on yearly basis, which implies two possible reasons:



1.dumplings are gaining popularity among people, more and more people tried this dish and gave feedback

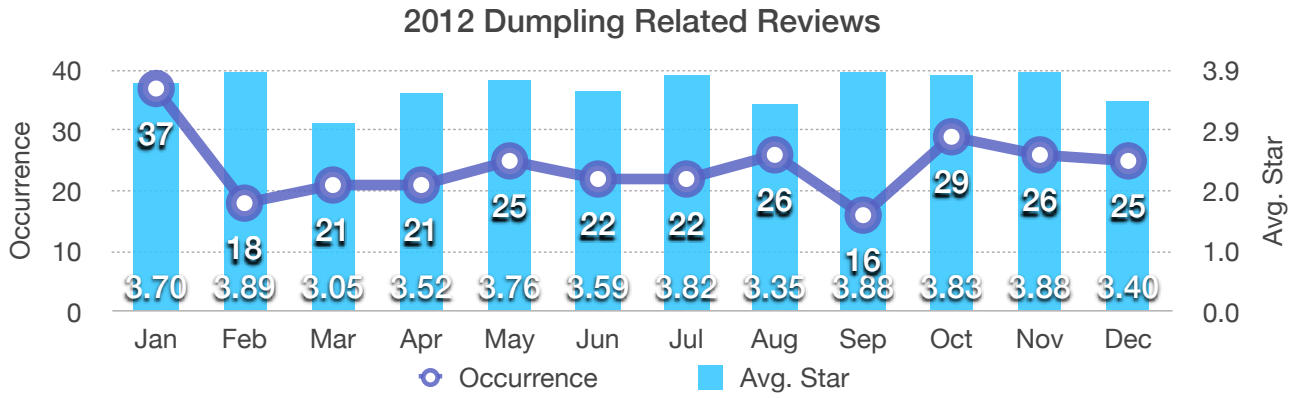
2.more and more people submit reviews on internet so that they can be used as a reference for other people

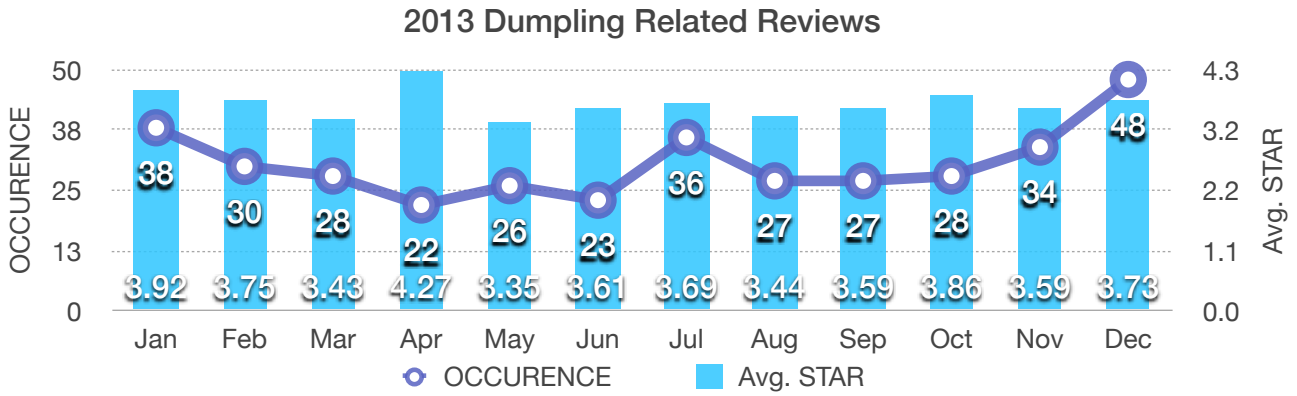
Another important factor is the average star, below are the average star people gave for dumpling during 2006 to 2014.

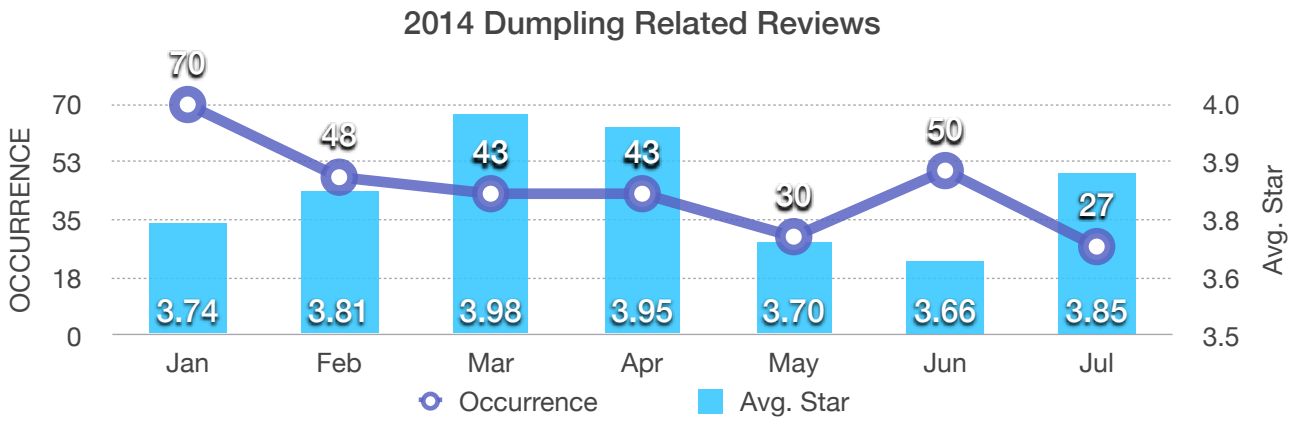


From left graph, we are sure that the assumption that dumpling is gaining popularity seems to be the case, according to the occurrence & average star, we can say that not only more and more people tried dumpling, but also gave positive feedback for this dish.

More specifically, during the year 2012, 2013 and 2014 (latest data from dataset), we can get the dumpling review occurrence and avg.star.







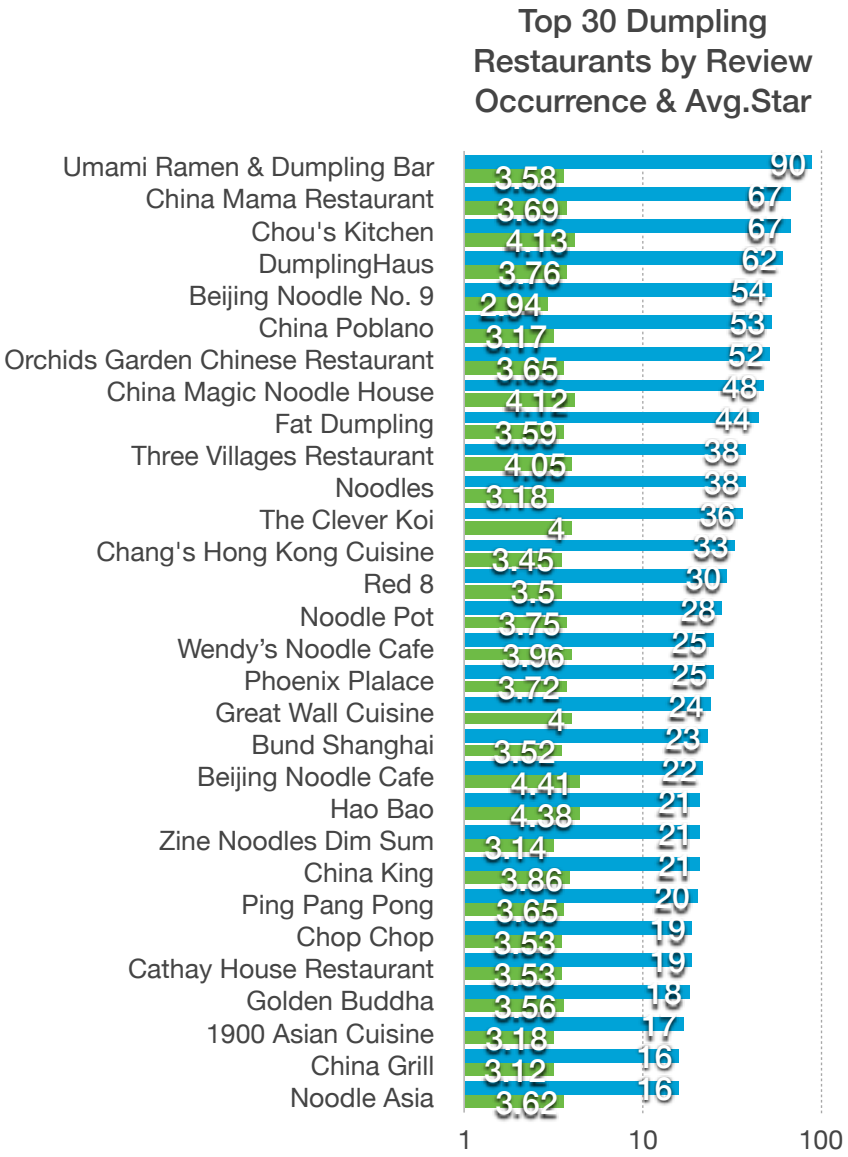
From above graphs, it is very clear that dumpling is gaining popularity from 2012 to 2014. Not only in terms of the monthly review counts but also the average star shows a rising trend.

Another interesting thing is that the review occurrence and avg.Star of second half year, especially winter (Nov, Dec, Jan), is higher than other seasons, it could be due to some Chinese related activities within the area. Another possible reason could be dumpling is popular during winter for its nutrition or other special effects.
Similarly, we can get such kind of statistics for other years, and analyse the trend regarding the popularity of dumpling.

Task5 – Restaurant recommendation

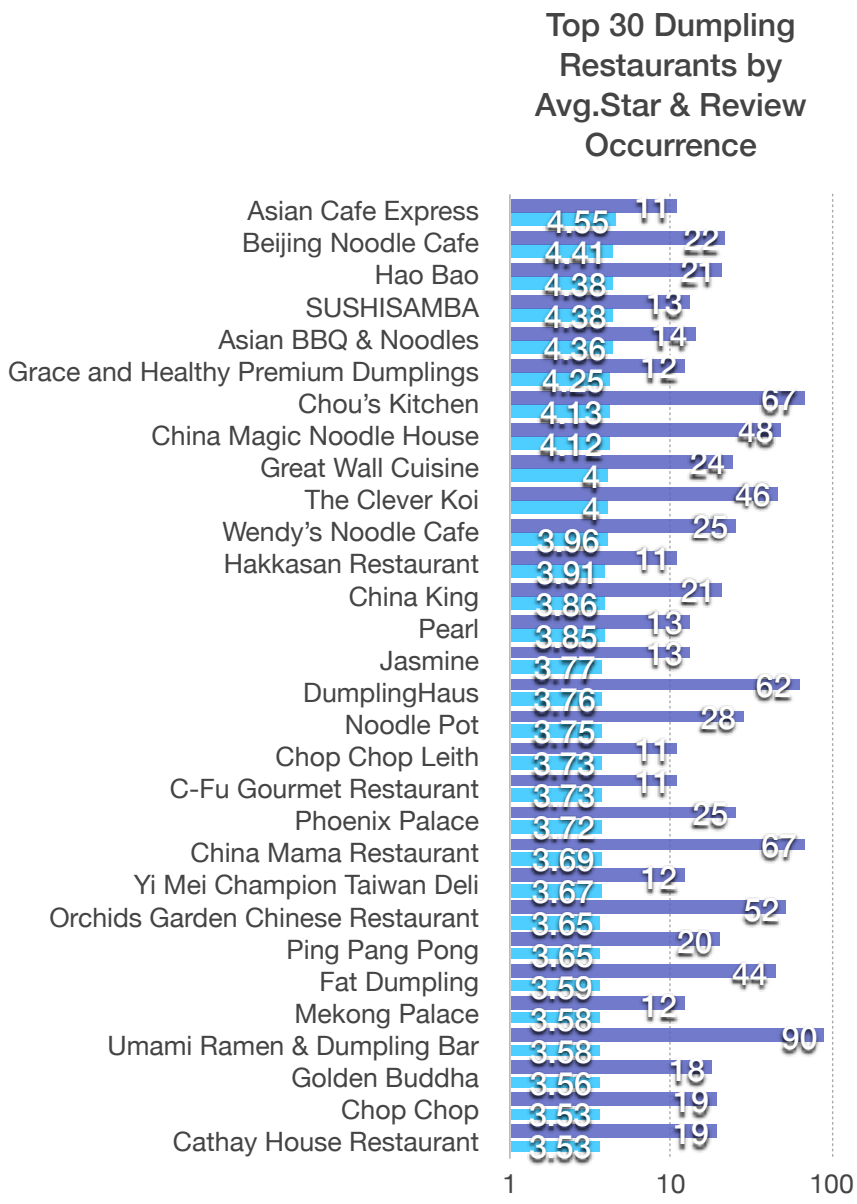
This task requires to make restaurant recommendation, in other words, a dish-specific ranking algorithm should be designed for restaurant ranking.

Generally, it's difficult to rank restaurant since each restaurant serves a wide variety of dishes, and almost all reviews are closely related to dishes rather than restaurant itself. Thus, a specific dish should be used as a baseline for ranking, like what I did in task4 case study. Similarly, the restaurant can be ranked according to occurrence and average rating of a particular dish. This time I still used "dumpling".



In the left graph, the first criteria is occurrence, the axis uses logarithmic format to reduce the gap caused by sharp value difference. As discussed in task4, occurrence does not reflect the quality of dishes since it lacks the detail rating of dishes and review's attitude.

In the right graph, the first criteria is average star, and I eliminate some restaurants whose review occurrence is less than 10, since such kind of review frequency may not be able to reflect the real situation.



From the above two graph, we can find that some restaurants receive lots of feedbacks about dumpling while at the same time provide high quality food according to their average star.

Thus combing the review occurrence and average star, we are sure that restaurants like"Beijing Noodle Cafe" and "Hao Bao" are really good at delicious dumplings. Actually, for dish recommendation, I think restaurants with more than 15 reviews and with average star at least 3.50 could be the candidate for dumpling recommendation.

Moreover, recommendation by occurrence and average star is not enough because all these information are based on the whole dataset, which ignores one important factor: location. Imagine how useless it would be if the system recommends a restaurant in New York while the user is looking for restaurants in Log Angeles.

Therefore, I take address into consideration and classify above restaurants by address. (** means recommended, ** means strongly recommended*)

Dumpling Restaurants in Edinburgh

Restaurant	Avg.Star	Rev.Occurrence	Address
Cool Jade Chinese Restaurant	5.00	1	Edinburgh EH12 7AU,3-4 Downie Terrace
Rainbow Arch Restaurant	5.00	1	Edinburgh EH3 8BJ,8-16a Morrison Street West End
Stack Dim Sum Bar *	4.25	4	Edinburgh EH6 8RG,42 Dalmeny Street Leith
Wing Sing Inn	4.00	1	Edinburgh EH11 1BP,147 Dundee St
Yum Yum HK Diner	4.00	1	Edinburgh EH8 9EF,13 W Richmond St Newington
Happiness	4.00	1	Edinburgh EH8 9PY,34 W Preston Street Newington
Chop Chop Leith **	3.73	11	Edinburgh EH6 6LX,76 Commercial Street Leith
Chop Chop **	3.53	19	Edinburgh EH3 8DT,248 Morrison Street West End
Saigon Saigon *	3.50	6	Edinburgh EH2 2AZ,14 S St Andrew St New Town
Hong Fu Noodle	3.00	2	Edinburgh EH1 3BG,3-7 Waterloo Pl
Wok & Wine	3.00	1	Edinburgh EH2 1LH,57a Frederick Street New Town
Loon Fung	1.00	1	Edinburgh EH3 5LE,2 Warriston Pl Stockbridge

This graph shows the restaurant which serves dumpling in Edinburgh, the restaurants are ranked by average star.

From this graph, I would like to recommend “Stack Dim Sum Bar”, “Chop Chop Leith”, “Chop Chop” and “Saigon Saigon” for people in Edinburgh who would like to try dumpling.

“Chop Chop Leith” and “Chop Chop” are strongly recommended since they have more than 10 reviews and holds a comparatively high average rating.

Similarly, I can classify restaurants in other places before making recommendation.

Dumpling Restaurant in WI

Restaurant	Avg.Star	Rev.Occurrence	Address
Main Moon Chinese Restaurant	5.00	1	4850 Larson Beach Rd Mc Farland, WI 53558
Panda Garden	5.00	2	922 Windsor St Sun Prairie, WI 53590
Magic Wok	5.00	1	2044 Atwood Ave Schenk - Atwood Madison, WI 53704
Chang Jiang Restaurant	5.00	1	5710 Raymond Rd Meadowood Madison, WI 53711
VIP	4.50	2	6718-6722 Odana Rd Madison, WI 53719
Double 10 *	4.43	7	3306 A University Ave Madison, WI 53705
Flaming Wok *	4.20	5	4237 Lien Rd Ste H Mayfair Park Madison, WI 53704
Orient House *	4.20	5	626 S Park St Greenbush Madison, WI 53715
Red Pepper	4.00	1	1019 N Gammon Rd Middleton, WI 53562
Grand China Restaurant	4.00	1	2608 Allen Blvd Middleton, WI 53562
House of Mei	4.00	1	120 E Main St Sun Prairie, WI 53590
Asian Kitchen	4.00	1	449 State St Capitol Madison, WI 53703
QQ Asian Buffet	4.00	1	1291 N Sherman Ave Madison, WI 53704
China Wok	4.00	4	1724 Fordem Ave Madison, WI 53704
Imperial Garden Chinese Restaurant	4.00	1	4214 E Washington Ave Norman Acres Madison, WI 53704
China Star	4.00	1	515 Junction Rd Junction Ridge Madison, WI 53717
China One West	4.00	1	518 Grand Canyon Dr Madison, WI 53719
DumplingHaus **	3.76	62	702 N Midvale Blvd Madison, WI 53705
Wah Kee Wonton Noodle *	3.62	8	600 Williamson St Ste E Williamson - Marquette Madison, WI 53703
Umami Ramen & Dumpling Bar **	3.58	90	923 Williamson St Williamson - Marquette Madison, WI 53703

From above graph, “Double 10”, “Flaming Wok”, “Orient House”, “DumplingHaus”, “Wah Kee Wonton Noodle” and “Umami Ramen & Dumpling Bar” are the good candidates for people live in WI.

From above graphs and analysis, there are several things deserve to be highlighted:
1. Occurrence could be a good indicator of popularity, however sometimes it should combine with average star when making recommendations, only those with a certain amount of reviews as well as comparatively high rating should be recommended to customers.

2. Location should also be an important factor for recommendation, especially when people trying to get something worth trying nearby instead of giving some out- of-

scope recommendations. For instance, from location based analysis, we can find out some good restaurants for people in specific state.

Task6 - Hygiene prediction

In this task, I was asked to predict whether a set of restaurants will pass the public health inspection tests given the corresponding Yelp text reviews along with some additional information such as the locations and cuisines offered in these restaurants.

For this task, I tried many ways to make prediction as accurate as possible.

I tried to process all the training reviews using nltk, like what I did in task 2, removing stop words, stemming, remove low frequency words, etc. Then I collected the most frequent words (freq >= 3) from these processed training reviews and manually removed number and meaningless phrases such as "on","at","\" from frequent word set. Then according to these most frequent words, I convert the training reviews to array representation list by occurrence. Likewise, I did the same thing for the rest reviews and convert them to array representations (e.g. [2,5,1,0,4,…]) .

Then I use SVC model in scikit-learn to train and predict labels.

From the feedback I find that strange words and numbers should be removed, in this way I could eliminate the possibility that some frequent outlier words affects the overall result. It's possible that someone prefer to use numbers or strange words in reviews, and give many reviews to different restaurants, thus affect other reviews.

Moreover, I think list containing occurrence count is more useful in that it keeps the popularity of specific words in the review which could been lost by representation of 0 and 1 list.

However, additional info such as review count and average rating does not help in improving predict results. Also the SegPhrase result doesn't help. In addition, I tested with several models such as Naive Bayes and LDA model to predict the hygiene condition, but the results are not as good as simply refined frequent words result.

Highlights

Usefulness of results

1. In task2, similar cuisines are explored based on reviews. From the mining result, we can find similar cuisines that might share something in common. During this process, LDA topic model plays an important role in identifying potentially related cuisines. The result from this task serves as the reference for task3 which is to recognize dishes according to similarity map.

2. In task4, ranking is always an interesting area to work with since it's difficult to design a fair and efficient algorithm. This task I tried to place popular dishes obtained from task3 by three categories: occurrence in review, average rating, and rating distribution.

   The ranking by occurrence and average rating give credit to the dish popularity, but rating distribution (sometimes can be viewed as attitude) should also play a role in the result.

   In a case study targeting 'dumpling', I collected review occurrence and average rating on yearly basis and monthly basis separately, from the trend I could make some predictions and verify them accordingly.

   In all, considering the above three aspects can we make a comparatively fair judgment of the dish popularity.

3. In task5, I was supposed to recommend restaurant to people who wants to try specific cuisine or dish. I applied similar strategy used in task4 to this task, which is considering the overall review occurrence, average rating before making recommendation.

   In order to make the recommendation more convincing, I delved into a specific case about 'dumpling' so that I was able to recommend restaurant which serves dumpling. Moreover, by considering location information, the recommendation can be targeted within particular area instead of giving a too wide-range recommendation.

4. In task6, making predictions based on current available information is the fundamental technique in data mining and machine learning.

Novelty

1. Sentiment mining

In task4 dish recommendation sentiment mining should be considered in that reviewers often express their feeling directly or indirectly through words and rating. Therefore classifying rating distribution could help in explaining some strange cases such as a dish with polarized rating distribution. The rating distribution could reflect the collective attitude of reviews thus help to make recommendation.

2. LBS recommendation

In task5 restaurant recommendation, besides occurrence and average rating, I also considered location information so that the information within a specific is available. In this way the user would more likely to try the recommended restaurants, if the system recommended a restaurant in another state, the user might not try it even if the restaurant is superb than those located nearby.

Contribution of new knowledge

Plotly & D3

I learned how to use Plotly in Data Visualization course and also used it to display graph in task2, but I didn't try D3 until the first task of this project, the rich library helped me a lot.

nltk

Before this project, I've never heard of nltk, however during implementation of task2, I found that nltk is such a powerful tool in processing text mining tasks. From task2 I obtained some basic steps like remove stop words, stemming, remove low frequent words, etc to process text mining. I also applied similar skills in task6.

I'll spend more time on nltk since it's really good in mining text.

SegPhrase & ToPMine

It's not easy to find meaningful phrases from huge amount of texts, for instance, exploring dish names from numerous reviews, thanks to these tools I was able to dig out hidden dishes in task3.

scikit-learn

I used scikit-learn before when I took another online course, in task6 I tried with several models in scikit-learn, the API and document are very clear and easy to use. Especially it incorporates NumPy and SciPy which are very popular in data mining and machine learning tool-kit.

Prospect

In order to make the data mining result more useful, I think a few ways can be applied:

1. **Adjust recommendation according to user feedback**

   The whole tasks are trying to extract potential correlated information from what we've got so far, such as the business information, review information, etc. These information is collected from past thus may not be able to fully reflect the current need of customer.

   In order to get the latest needs and make recommendation more accurate, customer feedback should be considered as an indispensable part.

   For instance, if a customer tries to search 'kong pao chicken' and several related restaurants are provided, user should be able to give feedback about these restaurants, whether the recommended ones are actually serving this dish or if the dish is good or just so so.

   The system could adjust the recommendation strategy correspondingly, in this way, the recommendation system is alive and keeping improving rather than a static system which totally relied on past data.

2. **Tag based classification and recommendation**

   Nowadays, another efficient way to find correlated information is through predefined tags.

   For instance, for dish like "dumpling", system could provide tags such as "Gorgeous", "Delicious", "Not bad", "Disappointed", etc. In such way, the system can mine the data just by checking associated tags to explore potential sentiment instead of trying to infer from the whole review, it could improve both the efficiency and accuracy of mining result.

3. **Make recommendation according to cuisine similarity result**

Another useful way is to make recommendation from task2 cuisine similarity result. In some cases, when user searches dish like "dumpling", he/she maybe:

1. Want to try "dumpling"
2. Actually want to try something looks like dumpling such as "wonton" or even "stemmed stuffed bun".

So in cases where user input something that similar to what they really want, the similarity recommendation result would be more preferable.

4. **LBS recommendation**

LBS has been in the air for several years, thus recommendation without considering location would provide not so much information as to almost nothing. It's a really bad user experience if the system recommends Chinese restaurant in China Town, San Francisco when user is located in New York.

Therefore, in task4 dish recommendation, I tried to classify the potential restaurant by state so that user from New York will not receive recommendation from California.

Finally I would express my appreciation to this project which gives me an opportunity to try and explore real data set, to apply what I learned in the past several month into practice, and most importantly, to get familiar with powerful data mining tools such as nltk, scikit-learn, SegPhrase, etc.