Course Project CSE 5334 : Yelp dataset challenge

# Motivation

Let’s suppose you are a budding business man wanting to open a restaurant in Charlotte, North Carolina. Major factors you need to consider before starting a business is the target area in the city, the current trend in the city with regard to cuisines, popularity of competitors, price range of restaurants around the area etc… these factors can help you place your restaurant in a most productive place with best profit outcome. Yelp provides us with a restaurant’s cuisine, location and price range information. The cuisines is divided into various types, it could be American food, Seafood, Thai, Italian etc... While locations can be roughly divided into the 4 categories say north, south, west or east side of the city. What if there was a way to combine data of all restaurant running in a particular city and provide a visual display of which type of cuisine (food category) is doing the best in which part of the city? Which regions in the city have the most expensive restaurants and more affordable ones? The goal of our project is to give this visualized information. A look into the web page can give you a review of trending cuisines in different regions of the city. From the vast degree of data provided in the yelp dataset we have reduced the data set to business in the city of Charlotte, North Carolina.

# Data Dusting and Method Design

Some of the major challenges we faced with the data was that the raw data obtained from the yelp data set was extremely vast. Also the data was in json format which was difficult to be parsed in our programs. The conversion of json to csv occupied a considerable part of effort. Since the json content did not have any strict rules regarding the required headers, each business record in the yelp dataset differed from the other, merely on checkin hours, categories and attributes like wifi available, ambience, parking etc.. Converting such varying json headers onto a strict csv header was challenging. Different python files were written to extract different attributes for business records.

Since we are focusing our project on businesses in North Carolina state, we had to extract this bit of data from the original dataset.

*Add\_Checkins.py* takes in the yelp checkin data set and sums up the checkins for each business in the set. *NC\_latlong.py* read through each of the records in yelp businesses, checked if they are of type restaurants and belonging to NC state and retrieved latitude and longitude all such records discarding other records.

Apart from extraction of the data we had to categorize the data obtained into different categories based on their offered cuisines. There are more than 35 cuisine categories, but representing all categories visually will not result in a very pleasing visualization, hence we categorized the cuisine types into 8 major categories.

**CATEGORIES:**

***American*** - consisting of Delis, Steakhouses, Bagels, American (New), American (Traditional), Breakfast & Brunch, Coffee & Tea sort or restaurants.

***Asian*** - consisting of Chinese, Thai, Indian, Japanese

***Fastfood*** – consisting of Burgers, Sandwiches ,Hot Dogs

***NightLife***/***Pubs*** - consisting of Bars, Lounges, Pubs

***Italian*** – Pizza, Ravioli

***Mexican*** – Taco, Burrito

***Seafood*** – Generally all seafood restaurants

***Buffets and Dinners*** – Mostly all the other categories that do not belong to the above mentioned categories.

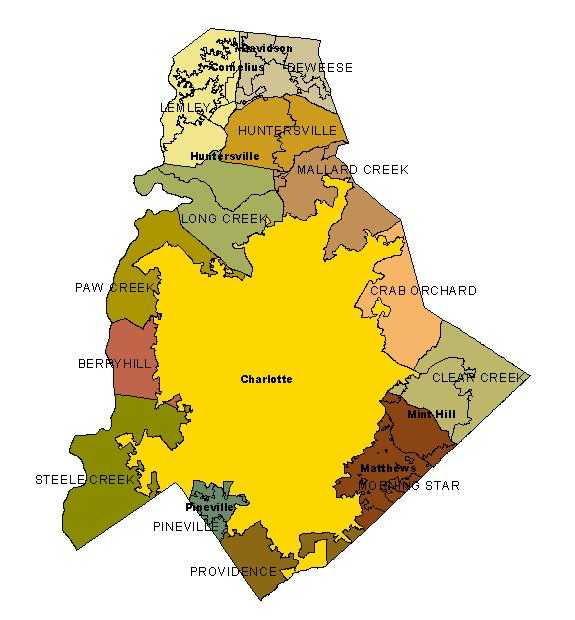
*NC\_category\_attribute.py* consists of the code that does the above categorization of the records. It first checks if the record belongs to the restaurant category, if yes it reads the sub types of categories with the above list mentioned and places the records in corresponding category.

Finally *combine.py* merges different csv files into one keeping business id as a common factor. Thus initially combine.py is used to combine the lat long or businesses to its categorized data, further the same combine.py with minor changes combines checkins of each business to its corresponding records.

The data obtained finally had truncated business data according to the needs of our project. The next step was to use **k-means clustering algorithm** to obtain clusters of the data.

Weka is a collection machine learning algorithms and the weka library can be used to write code in java. The clustering of the data was done using weka code *[Cluster.java]*. The initial thought process for clustering involved of using Euclidean distance of latitude, longitude, categories and sum of checkins among the many attributes. But the clusters obtained using this approach did not provide us with any insights on data. Hence we reduced the clustering attributes in k-means and used only the Euclidean distance of latitude and longitude to divide the records into four major clusters.

# Data Interpretations



The weka clustering helped us divide the restaurants into 4 major clusters. As we did our research on the locations of these restaurants we deduced that they are clustered into 4 major regions of Mecklenburg County in Charlotte City: **Matthews, Berry Hill, Pineville and Mallard Creek.**

Matthews to south east, Berry Hill west of the city, Pineville south west and Mallard creek at the north.

Each of them have been assigned a specified color for visual understanding in webpage.



# Visualization - D3.js Bubble Chart

Once the entire dataset in concern was cleaned using Weka and truncated to show only the required data along with their cluster information, we directed it to our web page.[*java\_csv\_op.csv*] We used D3.js library to visualize the data into an interactive bubble chart visualization. The major challenge here was the d3 library itself. Bubble chart visualization creates a single bubble for every entry in the csv data file used. This was an issue since we had close to 2000 business data for Charlotte city. Converting the clustered data into d3 usable data was the next step. We obtained that using some basic pivot chart techniques. At the end of this we had our final data to be used into visualization.

The final csv file [*stats\_data.csv*] included the data summarized according to the above mentioned regions (clusters) for each of the food category we made. For each category, we deduced the sum of checkin info and total number of businesses. The data also consists of number of restaurants belonging to $ through $$$$ price range.

The bubble chart now displays bubbles for each of clusters divided into 8 categories with varied radius sizes. Each of the bubble takes one of the above mentioned colors to represent the regions of the city.

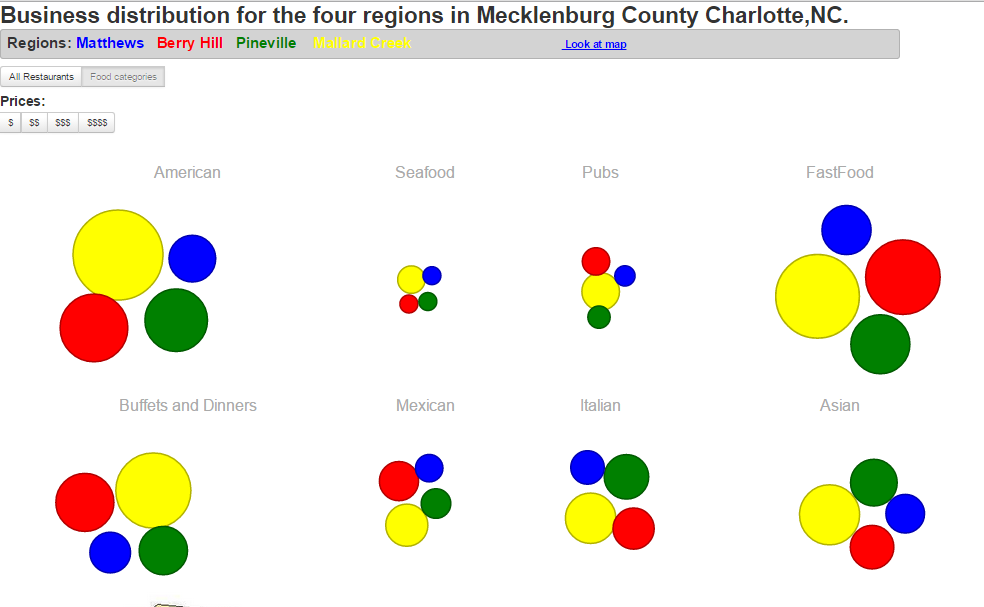
D3.js allows us to change the radius of the bubbles according to the set radius scale. We used this factor to set various scales for the bubbles. Total number of restaurants, number of cheap expensive or moderate priced restaurants are some of the scales we used. Another factor d3 allows us is the interactive and amusing movement of the bubbles. For this animation to occur we need to set the *gravity, friction* and *charge* of each of the bubble. *Gravity* here is the force that pushes each of the bubble towards the center of towards other bubbles. Whereas *charge* is an attribute that makes the bubble repel and not overlap on each other. *Friction* helps you control the speed at which each bubble moves. The *transition* assigned to each bubble controls the time duration of radius shifting.

**\*\***The coding part of the web page was inspired by the idea as seen in http://vallandingham.me/bubble\_charts\_in\_d3.html. The tutorial in the blog helped us understand various attributes of d3 and working on bubbles to display in the best way possible. The git hub repository https://github.com/vlandham/gates\_bubbles also provided by the same tutorial gave us the foundation to start coding for the web page.

# Visualization – Website

Consider some of the examples shown below before you explore our web page.

If you wish to look at the number of businesses running in regions categorized according to food category, Choose the button “Food Categories”. The radius of the bubbles will depict the count of businesses.

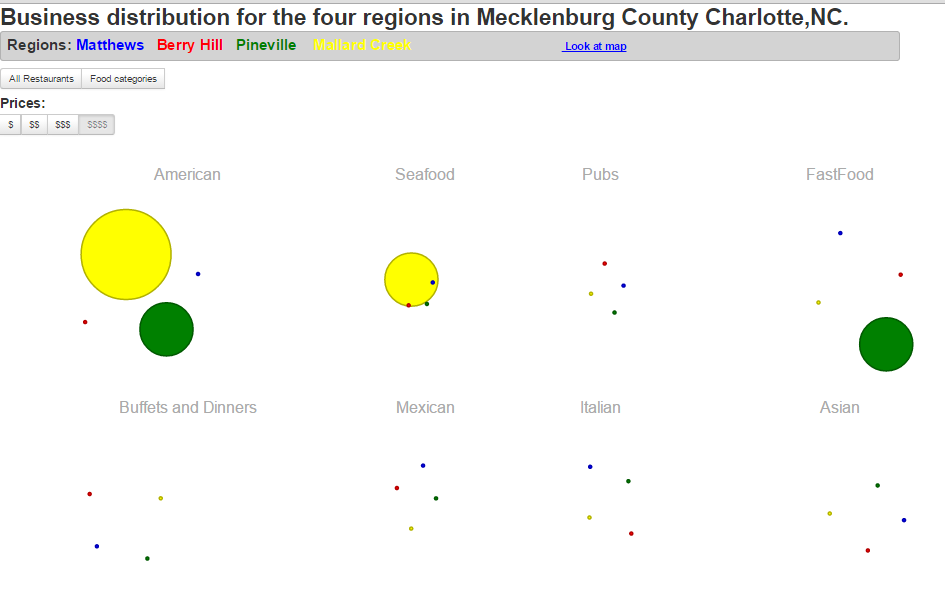


The above snapshot from the web page shows us the division of restaurants into the 8 food categories. It also shows us the percentage of restaurants in for each category type.

Various deductions from the above visualization:

1. There are very few sea food restaurants in the entire city in comparison to American and fast food restaurants. (Size of bubbles in seafood category is smaller compared to bubbles American category)
2. All the sea food restaurants are placed almost equally in all four regions of the city.(same size bubbles in seafood category)
3. Mallard creek has the most American restaurants. The size of yellow bubble is the largest in American category. ( This information helps you know the target crowd in Mallard creek )

If you want to look at ratio of restaurants offering price range $$$$ (most expensive restaurants) category wise, choose the price buttons, in this case “$$$$” the radius will reflect the number of $$$$ restaurants. If the bubbles take a size of a tiny dot, you can consider that there are NO restaurants in that region which offer high rates.



As shown above in the snap shot from the webpage, the selected price range is $$$$ (most expensive) range restaurants.

From the visualization we can easily tell

1. American, Seafood and Fastfood restaurants are the only categories to have expensive rates.
2. Pineville and Mallard creek are the most expensive places in charlotte (In terms of restaurants of course).

Additionally on hovering over each of the bubble it displays the information regarding the number of restaurants and number of checkins recorded for the region in concern.

Various other types of deductions can be obtained if you match the results of price with the food category view.

Project has been hosted on UTA’s omega server: <http://omega.uta.edu/~mxa1002/yelpdata_visualization/>

# References

<https://weka.wikispaces.com/Use+WEKA+in+your+Java+code>

<http://www.programcreek.com/2013/01/use-k-nearest-neighbors-knn-classifier-in-java>

<http://vallandingham.me/bubble_charts_in_d3.html>

<http://ncgenweb.us/nc/mecklenburg/resources/maps/>