Task 4: Popular Dishes

For task #4 I decided to figure out what is the most popular sandwich. I had from the previous task a set of dishes from 'American (New)' category. In additional I decided to not stop only o sandwich but try to find most closes dishes to sandwiches.

Before move on to descriptions of my step, I must notice that I’ve completed the task using python language and did it in iPython notebook (http://ipython.org/notebook.html). The original iPython with source codes can be found in my dropbox: https://www.dropbox.com/s/msumemo7a4hs33j/Capstone%20project%204%20and%205.html?dl=0

1. Synonymous to sandwich

With help of Apache Spark DataFrame (https://spark.apache.org/docs/latest/sql-programming-guide.html) I extracted all reviews that belong to businesses in the category 'American (New)'. Then I filter out all punctuation and make all words lower case. Based on this dataset I trained the Word2Vec and find the closest words for a word ‘sandwich’. The four top words is presented below:

```
panini: 1.69841198009
sandwhich: 1.69731582704
wrap: 1.64400842964
reuben: 1.62241896087
```

All words above and a word ‘sandwich’ I placed as a core set. The next was to find all match in dish list of these target words.

2. Build dish dataset

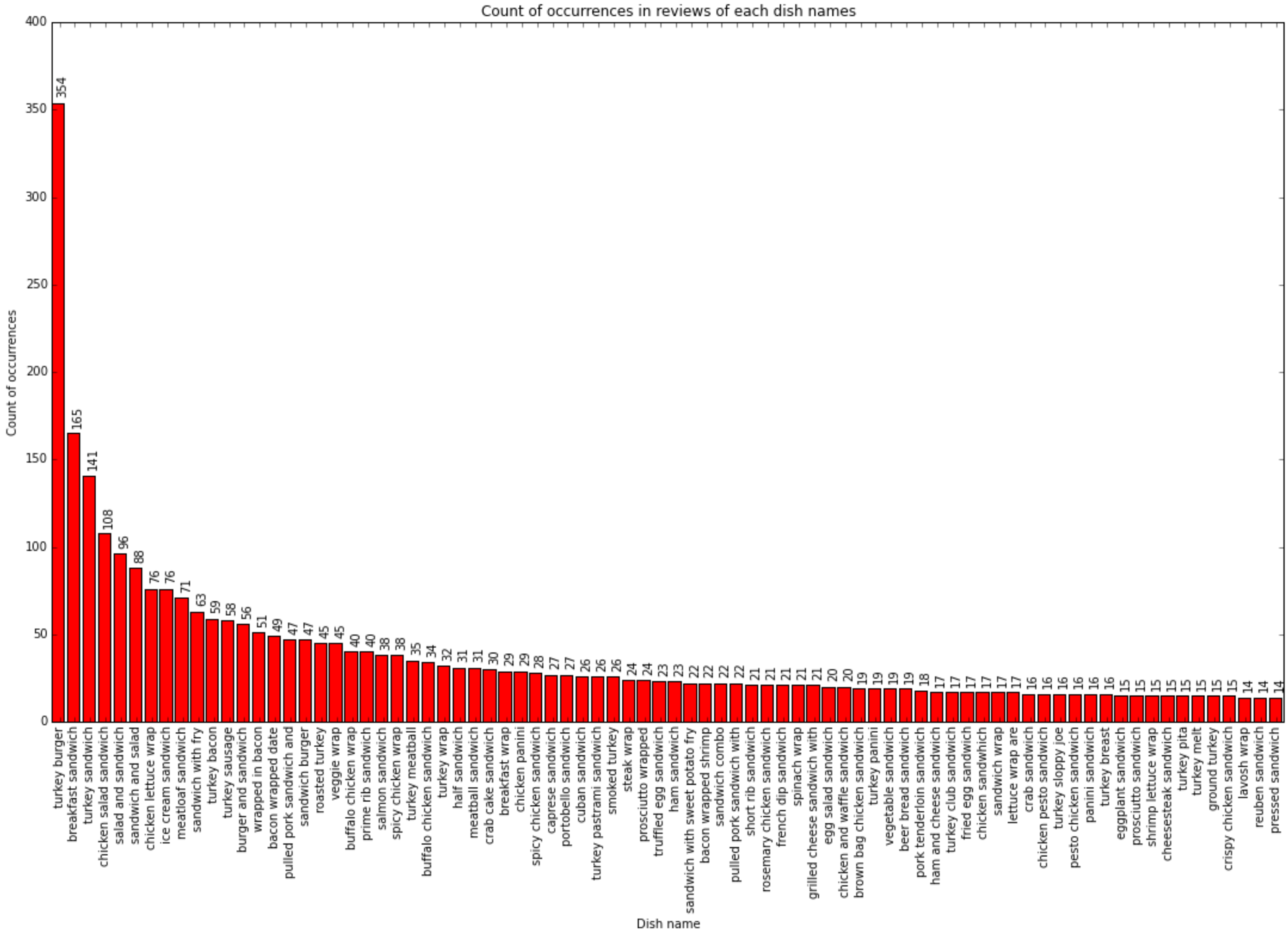
Simplest search by substring in dish list from task #3 gave me **268** dishes, but there were duplicates of names because of slightly different wording, for example ‘chicken sandwich’ and ‘chicken sandwiches’. So I decided to apply lemmatizer (http://www.nltk.org/api/nltk.stem.html) to put all words in regular form. Also I decided to focus on all long dish names, I mean not just ‘sandwich’, because all reviews with the word ‘sandwich’ includes all specific reviews like ‘whopper sandwich’. As a result, I’ve got **192** not overlapped dish names.

3. Extract all reviews for dish dataset

The next task was to extract all reviews related to defined dataset. Again with help of powerful Apache Spark and applying lemmatizer to normalize all words I selected 10733 reviews where any of dish names meet.

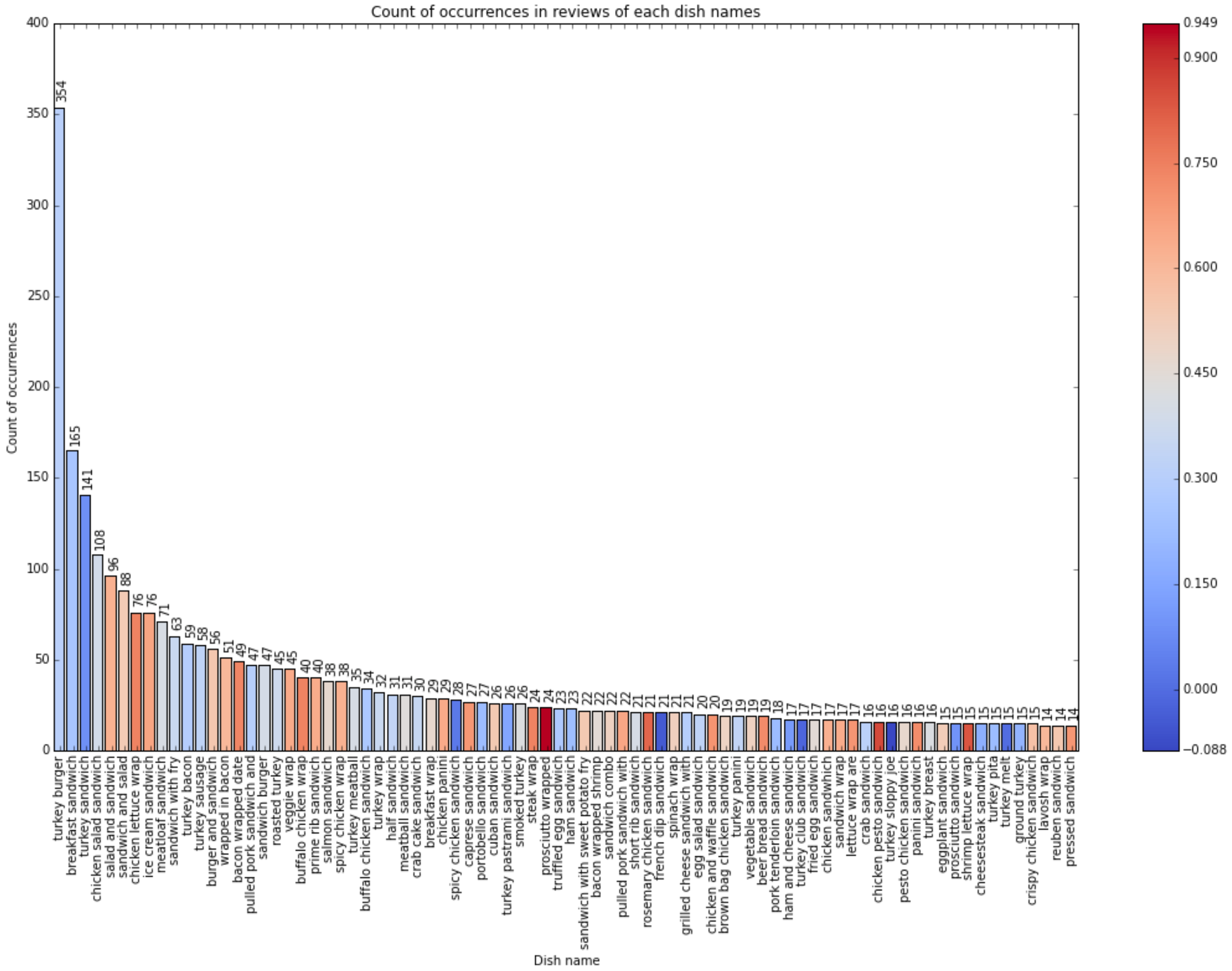
4. Build number of occurrences

For each dish from the dataset I count number of occurrence and it filtered out 30% of my proposed dish names, thus I left only with 134. Due to reason of better representation I worked further only with 80 most occurred dishes. The bar diagram for number of occurrences of each dish is presented below:



5. Sentiment analysis

Then I set a goal to perform sentiment analysis. For this purpose I used python TextBlob library (https://textblob.readthedocs.org/en/dev/quickstart.html#sentiment-analysis) that relies on python nltk tool. The algorithm for sentiment analysis in TextBlob uses naïve Bayes classifier from python scikit-learn (http://scikit-learn.org/) and it uses as a training data the movie review data presented here (http://www.cs.cornell.edu/people/pabo/movie-review-data/). TextBlob performs punctuation removing as well as Lemmatizer to normalize words. I decided to use average value of sentiment score for each dish and the result plot is presented below. It includes both elements: count and sentiment, for sentiment I used color components of each bar.

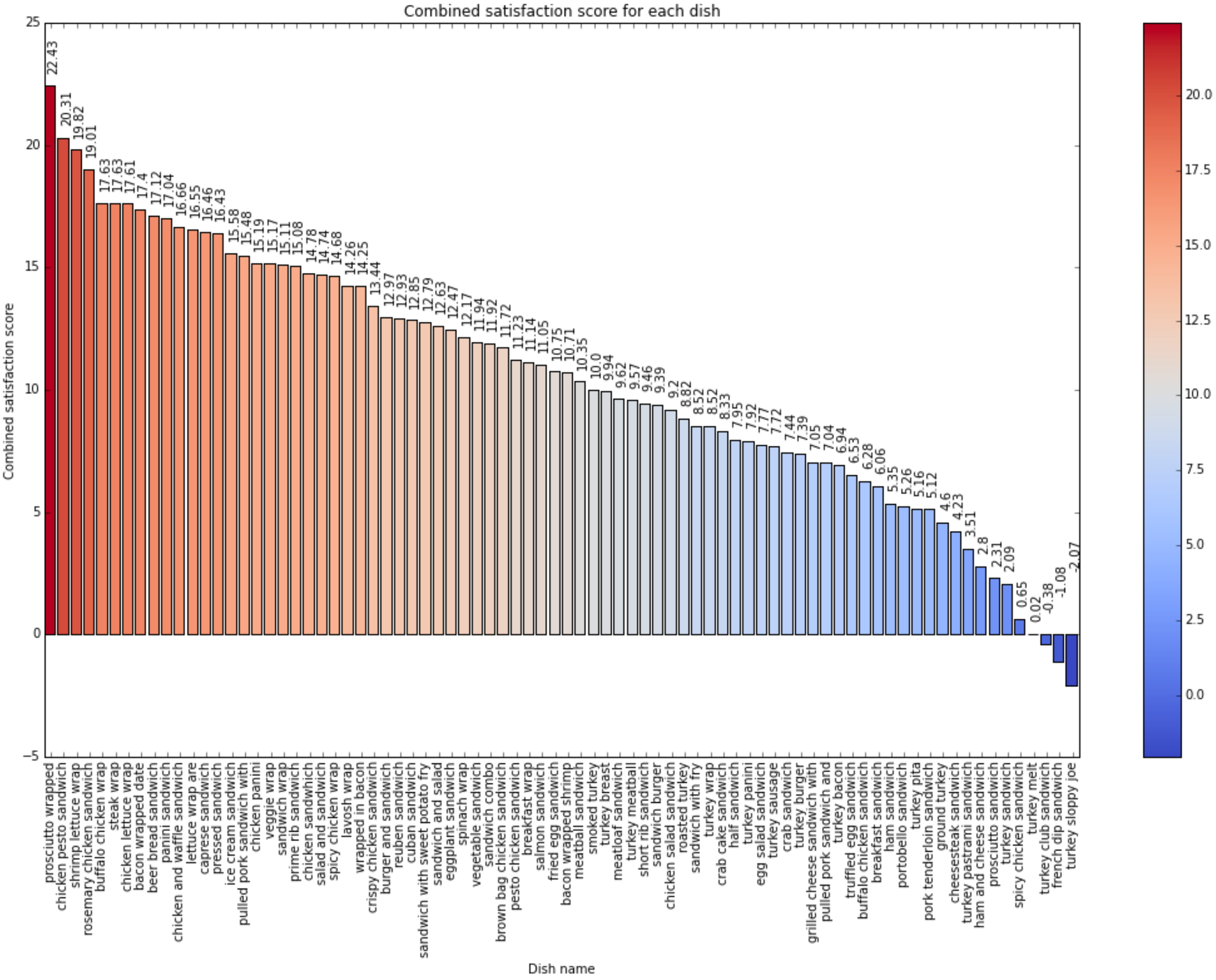


Here we can see that despite of high number of occurrences for turkey burger the average sentiment is not so high. If we consider only sentiment score then prosciutto should take the first place, chicken pesto sandwich the second one.

6. Combined popularity metric

I tried to combine two characteristic of each dish: sentiment score and number of reviews. Indeed, what we can recommend is something what make us satisfied, in our case we can say sentiment score plays this role. But we we are all different and we don't want to try something 'risky'. In our case the number of reviews show how confident we can be relying on reviews. Thus we need to penalty dishes with lower number of reviews, I decided to decrease it in proportion of max number of reviews, this leads that 'temperature' of scores comes down if dish has just a few reviews. I named this combined metric as a combined satisfaction and it is defined formally as:

$Sat_i = \frac{s_i}{(1+\log(\max_j c_j/c_i)}$, where s_i is sentiment score for i-th dish and c_i is occurrences count for for i-th dish.



Task 5: Restaurant Recommendation

In task #5 I continued the idea around sandwiches. So I tried to find the best place to try sandwiches that you just could dream.

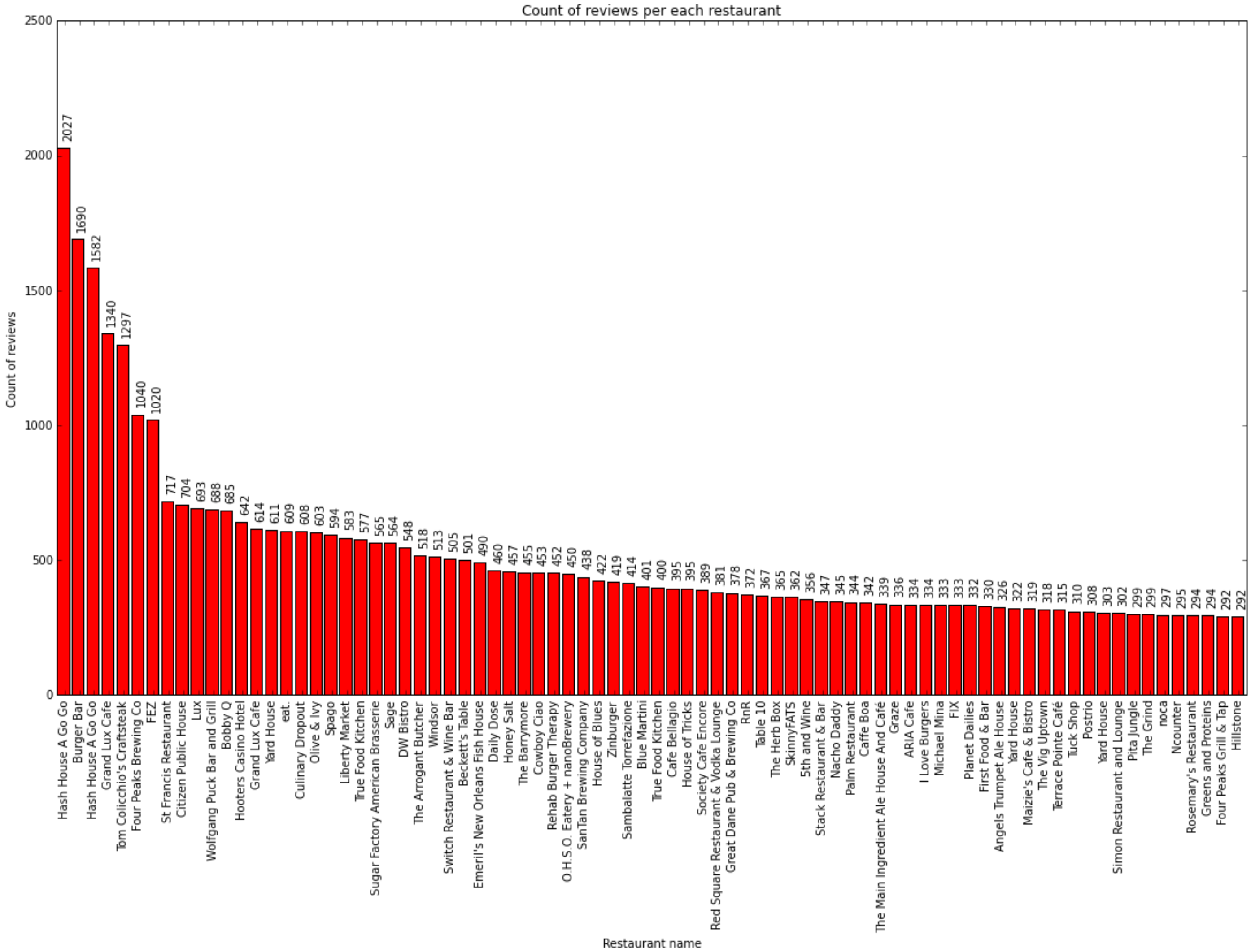
1. Collect all target reviews and businesses

On this step with help of Apache Spark SQL I performed search all business that have reviews where we can meet a word ‘sandwich’. It gave me **727** restaurants. After that I collected all reviews written for those restaurants to analyze and convert them into a sequence of triples: business id, restaurant name, review. The total number of such triples was **86047**. This was my target dataset.

2. Simple statistic of reviews for restaurants

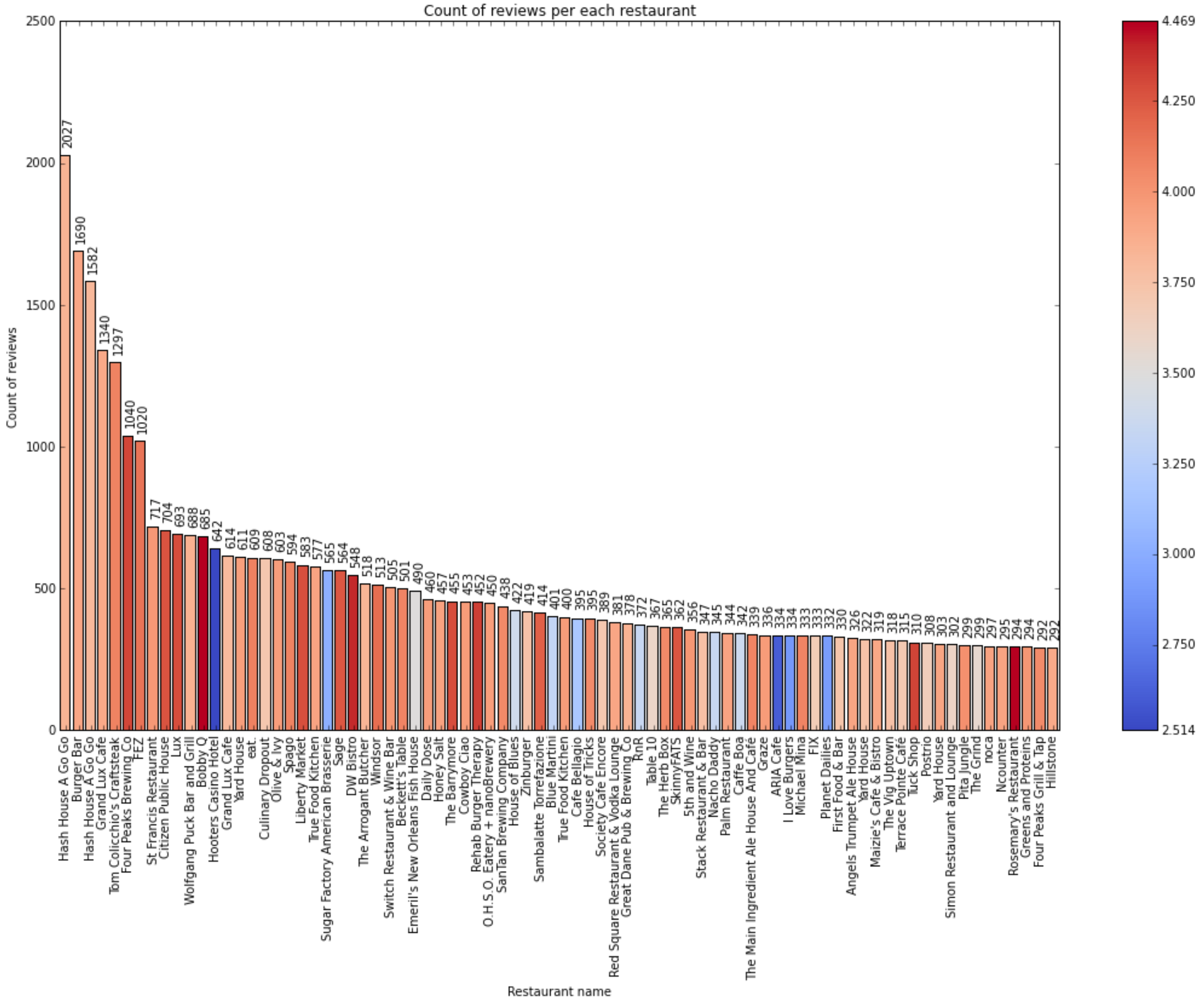
The easiest way to make recommendation is suggest visiting the most popular restaurant. In our case popularity can be defined as a most reviewed restaurant. The plot below presents such metric.

The leader is ‘Hash House A Go Go’. It is necessary to notice that the same name we can one third place, these restaurants have different business_id so most likely this is two restaurants from one network. The next question how fearfully is our assumption that this restaurant is the best in class. Maybe all these reviews are negative.



3. Review statistics with average stars

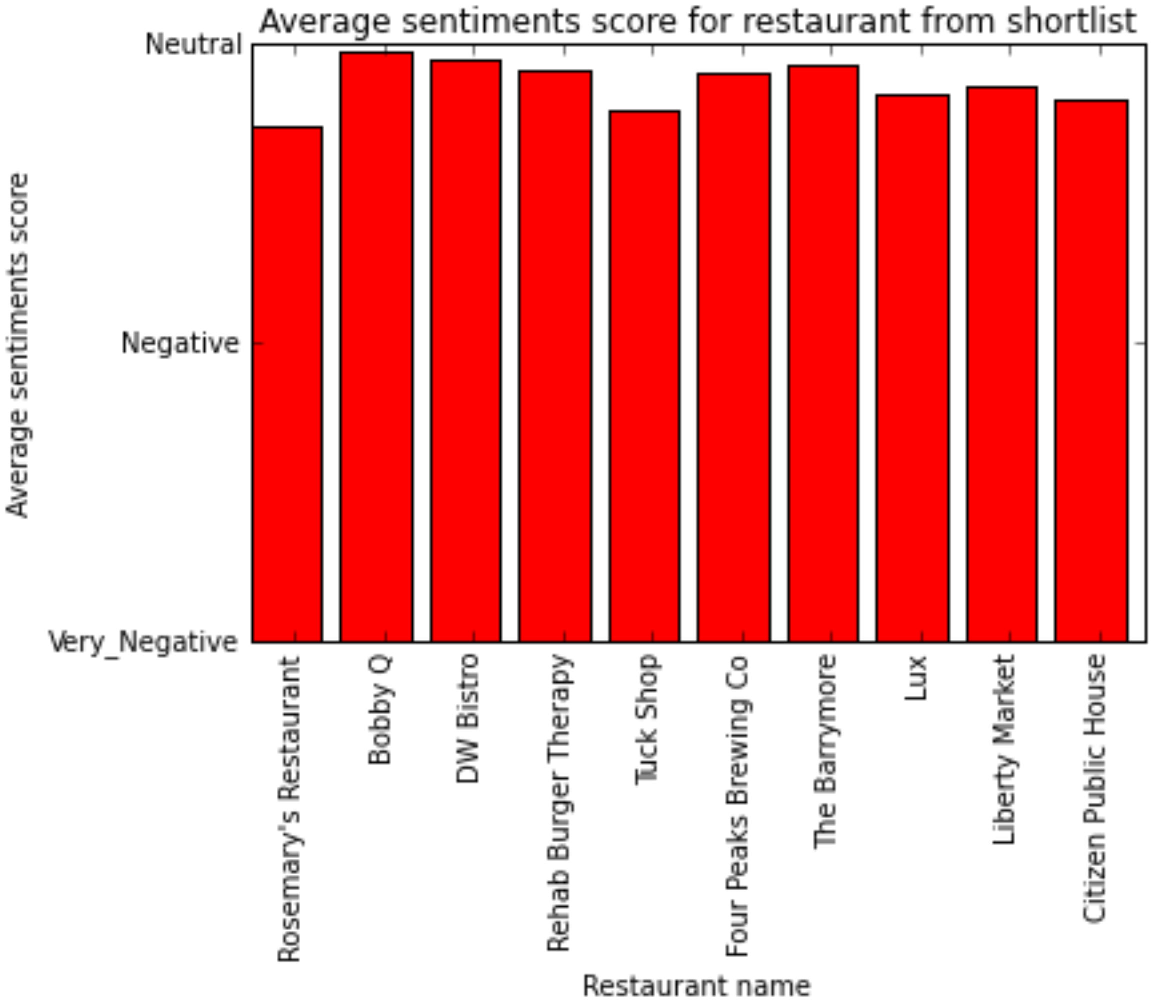
To check previous assumption, I evaluate average stars rating for each restaurant. I added this information as a color component to previous diagram. Based on this diagram we can see that almost all restaurants have average stars rating about 4. There is only a couple of anti-leaders that have quite low rating (below 3). Thus I can make conclusion that only based on star rating it is quite hard to make decision which of these restaurants is the best place.



Because of really big number of reviews I focused only on top five restaurants from our list with highest average rating. Also I used for this purpose the provided java-based tool because it turned out to be significant faster.

```
'Rosemary's Restaurant' number of reviews: 294      average rating: 4.4693877551
'Bobby Q' number of reviews: 685      average rating: 4.46277372263
'DW Bistro' number of reviews: 548    average rating: 4.39781021898
'Rehab Burger Therapy' number of reviews: 452       average rating: 4.33849557522
'Tuck Shop' number of reviews: 310    average rating: 4.32258064516
'Four Peaks Brewing Co' number of reviews: 1040     average rating: 4.31153846154
'The Barrymore' number of reviews: 455       average rating: 4.29230769231
'Lux' number of reviews: 693  average rating: 4.29148629149
'Liberty Market' number of reviews: 583      average rating: 4.27958833619
'Citizen Public House' number of reviews: 704       average rating: 4.26704545455
```

The graph for average sentiments score is presented below



It was found that despite the high average rating of the previous leader, a more refined analysis based on the sentiment analysis indicated that the general mood of the comments for 'Rosemary's Restaurant’ is below than the next applicant on the list.

Here we can see that the the final leader is ‘**Bobby Q**’ place. The address of this place is **8501 N 27th Ave Phoenix, AZ 85051**.