

# Task 4 & 5

Duc Thien Bui

## Task 4:

### Popularity

In this task, I will compare the popularity of chinese dishes that I found in week 3.

```
reviews = pd.read_csv('categories/Chinese.txt', sep='\n\n',  
                      encoding='utf-8', header=None, engine='python').apply(lambda x: (x[0]),  
lower(), 1).as_matrix().tolist()  
with open('dishes.txt') as f:  
    dishes = set([l.strip() for l in f.readlines()])
```

For each dish, I will calculate number of review that contain this dish name.

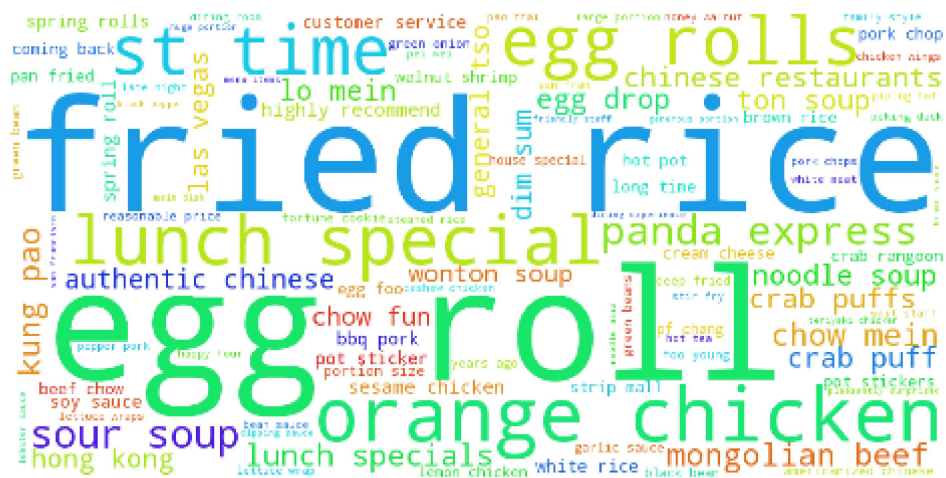
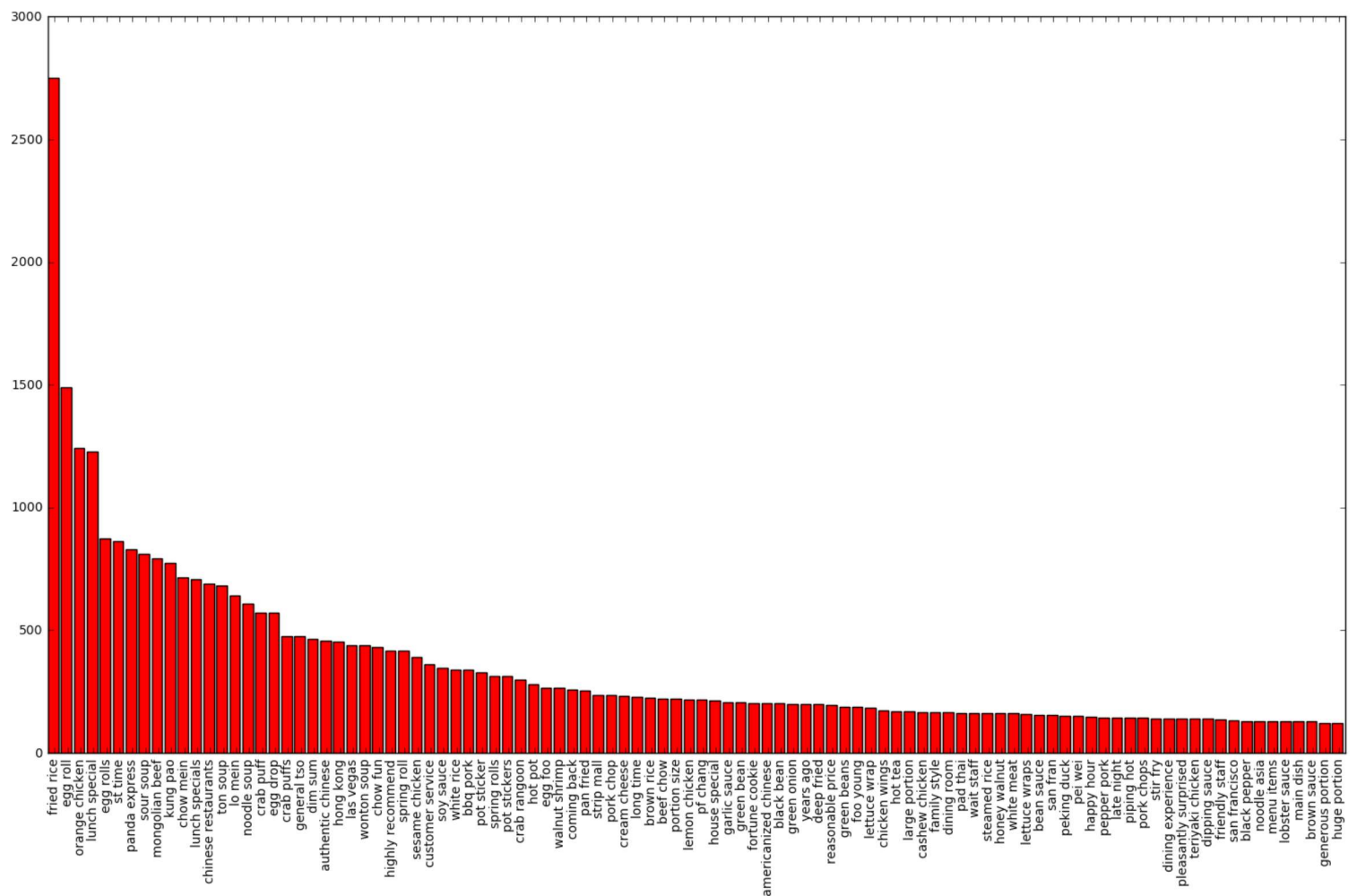
```
counter = Counter()  
for dish in dishes:  
    for review in reviews:  
        if review.find(dish) >= 0:  
            counter[dish] += 1
```

As we can see, top 3 dishes that are reviewed are *fried rice*, *egg roll* and *orange chicken*

```
counter.most_common(3)  
  
[('fried rice', 2749),  
 ('egg roll', 1490),  
 ('orange chicken', 1243)]
```

Base on the counter, I printed the frequency of top 100 dishes. We see that *fried rice* are being mentioned much more frequently than other dishes.

```
draw(counter.most_common(100))  
word_cloud(counter.most_common(100))
```



## Satisfaction

After that, I use NaiveBayesAnalyzer analyze the reviews. It is a algorithm that calculate the probility of a sentence being positive. I will calculate mean of the probility to fingure out which is most satisfied dish.

```
naiveBayesAnalyzer = NaiveBayesAnalyzer()
naiveBayesAnalyzer.train()
blobber = Blobber(analyzer = naiveBayesAnalyzer)
```

```

cnt = defaultdict(float)
for review in reviews:
    blob = blobber(review)
    for dish in dishes:
        if review.find(dish) >= 0:
            cnt[dish] += blob.sentiment.p_pos

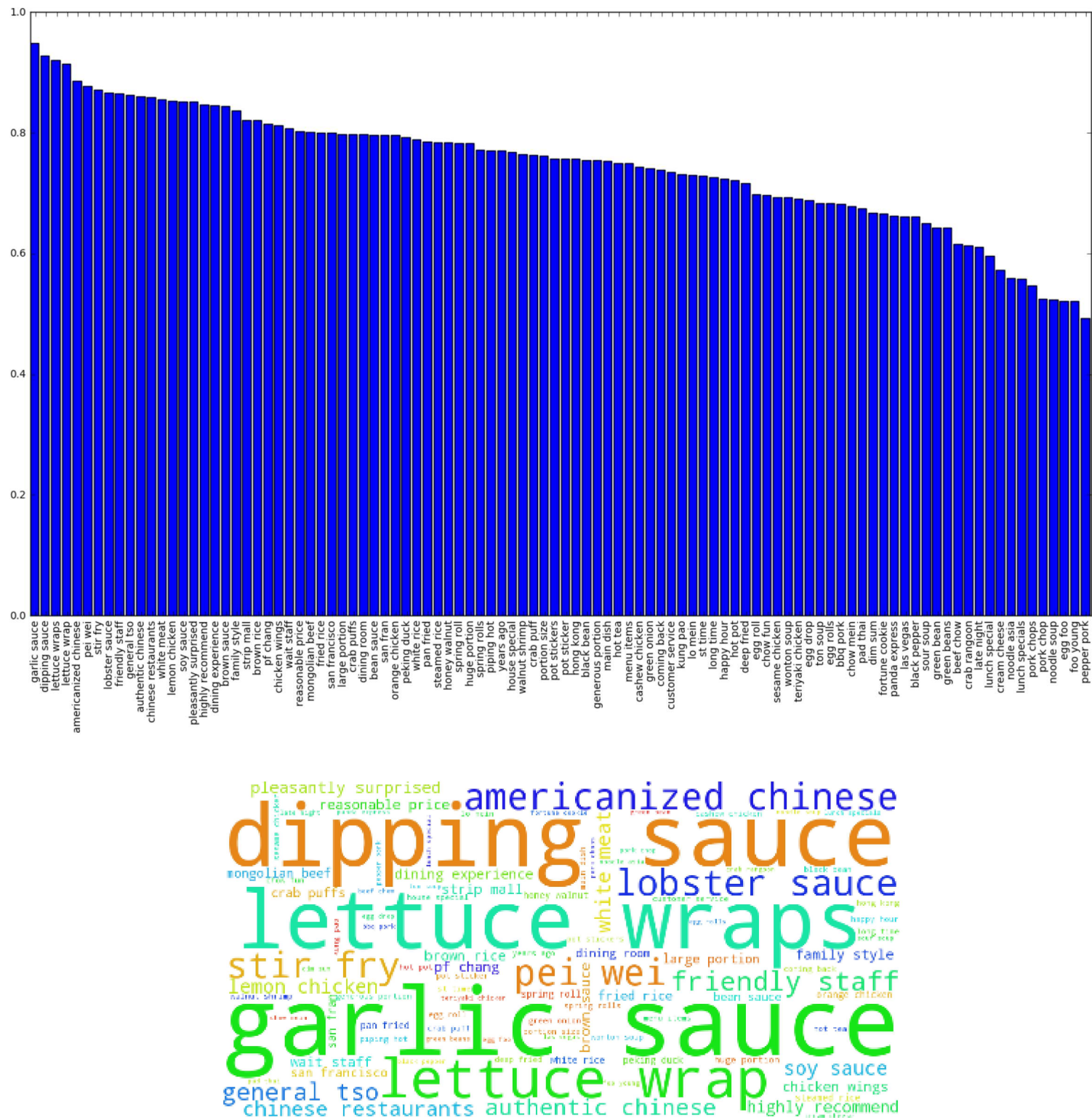
```

As can be seen, the food that has highest positive probility is garlic sauce and the lowest is pepper pork.

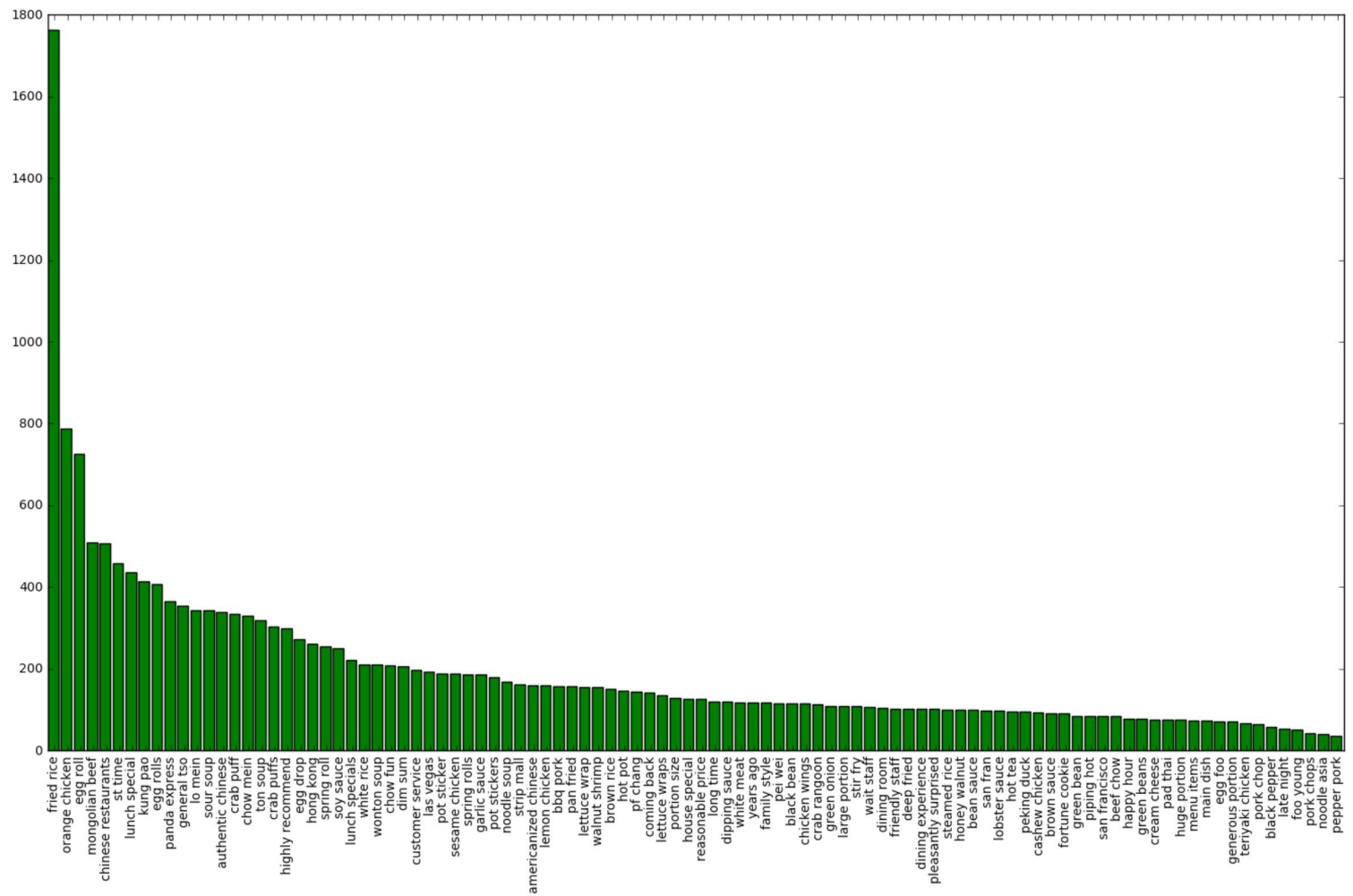
```

draw(sorted([(key,cnt[key]/value) for key,value in counter.most_common(100)],key=lambda x:
    -x[1]),'b')
word_cloud([(key,cnt[key]/value) for key,value in counter.most_common(100)])

```



```
draw(sorted([(key,cnt[key]*(cnt[key]/value)) for key,value in counter.most_common(100)],key
y=lambda x: -x[1]),'g')
word_cloud([(key,cnt[key]*(cnt[key]/value)) for key,value in counter.most_common(100)])
```



## Task 5:

In this task I will use yelp original dataset.

```
reviews_df = gl.SFrame.read_json('yelp_dataset_challenge_academic_dataset/yelp_academic_da
taset_review.json'
                                ,orient='lines')
business_df = gl.SFrame.read_json('yelp_dataset_challenge_academic_dataset/yelp_academic_d
ataset_business.json'
                                   ,orient='lines')
```

I set the target value to be the most popular dish (*fried rice*) and It can be set to be any value.

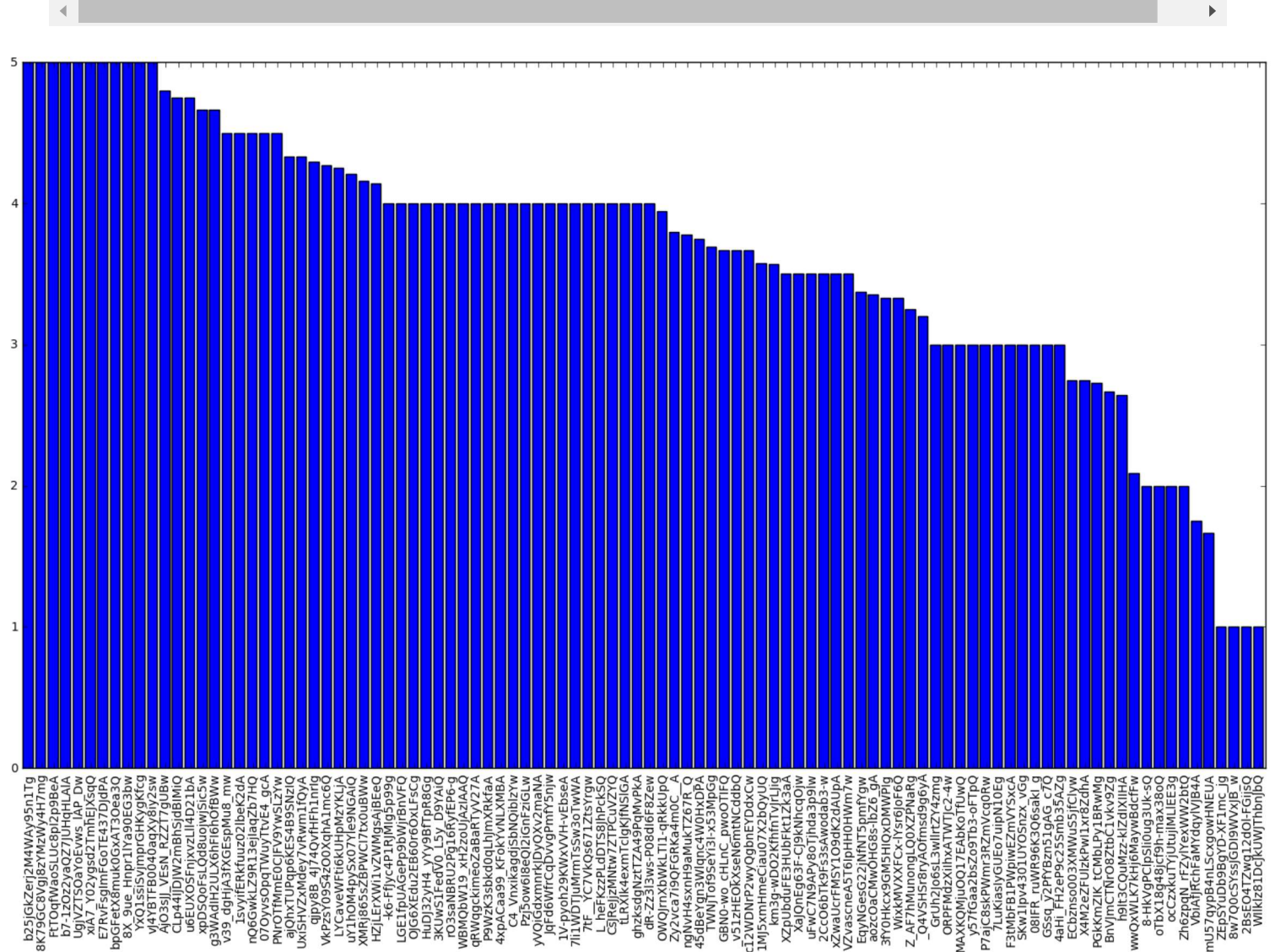
```
target = counter.most_common(1)[0][0]
```

First I will only use the reviews that mentioned target dish. Then I group all the reviews base on their **business\_id** and calculate mean of all review's star.

```
filtered_reviews = reviews_df[reviews_df.apply(lambda x: x['text'].find(target) >= 0)]
business_rating = filtered_reviews.groupby(key_columns='business_id',
                                           operations={'Mean Rating':
gl.aggregate.MEAN('stars'),'Count':gl.aggregate.COUNT())})
```

I can see some restaurants have full star, that maybe because they have too few reviews. I also do not want to recommend a restaurant that do not have sufficient number of review. So after that, I filtered out all restaurants have less than or equal 25 reviews about *fried rice*. The Plot have much smaller data and the highest average star is roundly 4.5

```
draw(sorted(list(business_rating[['business_id','Mean Rating']].to_numpy().tolist()[0:100]),
y=lambda x: -float(x[1])) , 'b')
```

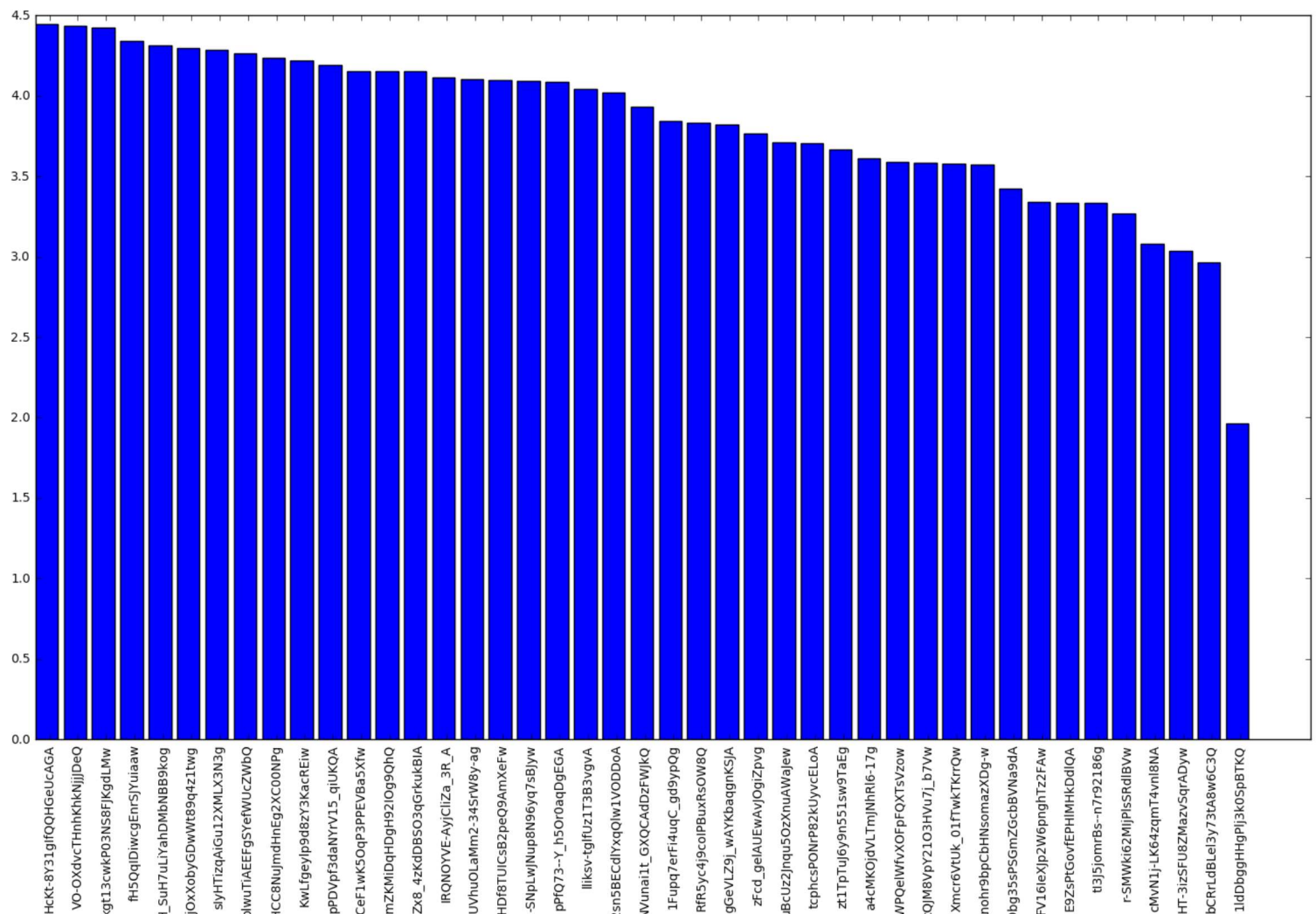




```
business_rating = business_rating[business_rating['Count']>25]
```

```
best_res = sorted(list(business_rating[['business_id', 'Mean Rating']].to_numpy().tolist()),
                  =lambda x: -float(x[1]))
```

```
draw(best_res, 'b')
```



```
best_choice = business_df[business_df['business_id'] == best_res[0][0]]
```

Now I can recommend the place that has the highest rated *fried rice*.

```
best_choice[0]['name']
```

```
'Rice Trax Teriyaki Grill'
```

```
best_choice[0]['full_address']
```

```
'7780 S Jones Blvd\nSouthwest\nLas Vegas, NV 89139'
```