




Yelp Review

Web Scraping, Sentiment Analysis

Frank Wang


5/22/ 2016


example



[Home](#) [About Me](#) [Write a Review](#) [Find Friends](#) [Messages](#) [Talk](#) [Events](#)


ABC Kitchen

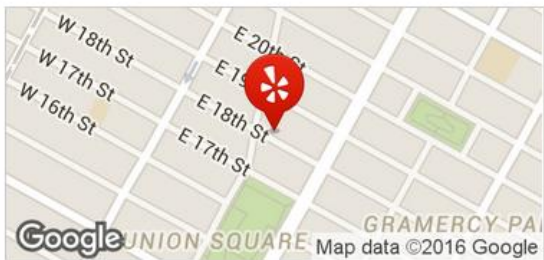
 2108 reviews


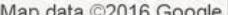
 Details


\$\$\$

Modern American, French

 Edit




 


 35 E 18th St


New York, NY 10003

b/t Broadway & S Park Ave



Union Square, Flatiron


 Get Directions

 **N Q R** 14 St. - Union Sq and 2 more stations

 (212) 475-5829


 abckitchennyc.com




 **pesto fettuccine and prosciutto and...** by Eric J.

But there's always room for dessert! We got the Basil and Mint Panna cotta with Meyer Lemon Sorbet! The Panna Cotta was soft and light. It came out looking like an egg because of the thinly sliced lemon on top. The lemon sorbet was a good palette cleanser and keep your mouth guessing. Will I get the tart sorbet or the creamy panna cotta this time.


P.S. They have a gender neutral bathroom!



Roasted carrots, avocado salad AND the soft shell crab special




Confit with mash potatoes





Lemon Panacotta with lemon sorbet with


See all photos from Joanne K. for ABC Kitchen

Was this review ...?

 Useful 2


 Funny

 Cool



Kelly P.
Jersey City, NJ
360 friends
128 reviews

 5/19/2016

 2 check-ins

The fresh closures of Telepan, Perilla, and Fishtail exhibit the caliber of New York's restaurant scene. It is not only over saturated but also dense with merciless critics. Even Michelin celebrity chefs are unable to persevere. Spaces are quickly transformed with new successors that share the same doomed fate... and eventually they become a Chase bank branch.

ABC Kitchen's business is booming. Jean Georges Vongerichten, the Michelin awarded chef pleases the eager palettes of his patrons at this establishment, located minutes away from Union Square. Like a speakeasy, there is a back entrance through a home furniture store. All décor is furnished courtesy of ABC Carpet and Home which expresses their passion for "green" through sustainable furnishings. Brick walls are painted with thick layers of milky white lacquer. Pillars of dark wood, untouched by varnish are piped throughout the dining room for a striking contrast. Simple metal pendant light fixtures, bare bulbs, and traditional Victorian chandeliers dot the ceiling. After sunset, flickering candles materialize on white table surfaces illuminating the dining room with a romantic dim luminescence. The environment compliments the restaurant's ingredients perfectly: strictly local and fair trade farm-to-table. Pesticides, antibiotics, and hormones

Yelp review scrapy, Sentiment Analysis and NB Prediction

This Project first scrapy yelp review data, then we do Sentiment Analysis, prediction using Naive Bayes Frank Lanfa Wang, 5/2016, FrankWanglf@gmail.com

This section is app for Yelp review download

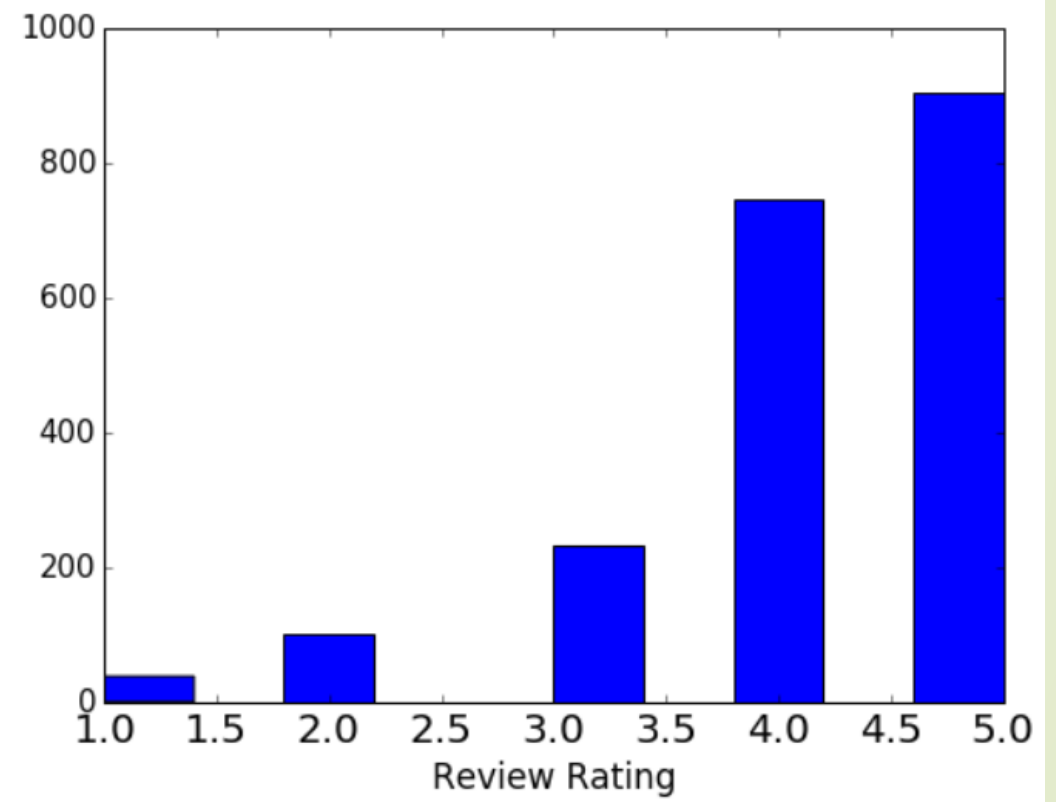
```
In [1]: ## Project 3 Web scraping
%matplotlib inline
import pandas as pd
from bs4 import BeautifulSoup
from urllib.request import urlopen
import re

queries = 0
tot_reviews=0
tot_authors=0
tot_ratings=0
authors_rev=[]
ratings_rev=[]
Features=[]

f=open("summary_auth_rating_tmp.txt", encoding='utf-8', mode="w")
f1=open("reviews_tmp.txt", encoding='utf-8', mode="w")
f2=open("author_tmp.txt", encoding='utf-8', mode="w")
f3=open("rating_tmp.txt", "w")

while queries < 2020:
    stringQ = str(queries)
    page =urlopen('http://www.yelp.com/biz/abc-kitchen-new-york?start=' + stringQ)
    soup = BeautifulSoup(page,"lxml")
    reviews = soup.findAll('p', attrs={'itemprop':'description'})
    authors = soup.findAll('meta',attrs={'itemprop':'author'})
    ratings= soup.findAll('meta',attrs={'itemprop':'ratingValue'})
    flag = True
    indexOf = 1
    for it,review in enumerate(reviews):
        dirtyEntry = str(review)
        tot_reviews+=1
```

	Name	Rating
0	Joanne K.	5
1	Kelly P.	4
2	Jane R.	3
3	Qian H.	4
4	Serena A.	5
5	Eugenia L.	5
6	Sheila R.	5
7	Jason H.	3
8	Julia K.	4
9	Jenn P.	4



Sentiment Analysis

```
: from textblob import TextBlob
from textblob.sentiments import NaiveBayesAnalyzer
sa_score=[]
for ir in range(len(reviewers.Rating)):
    testimonial = TextBlob(Features[ir])
    sc=testimonial.sentiment.polarity
    #sc=sc*2.5+2.5
    sc=sc*2+3
    sa_score.append(sc)
reviewers['Sa_score']=sa_score
```

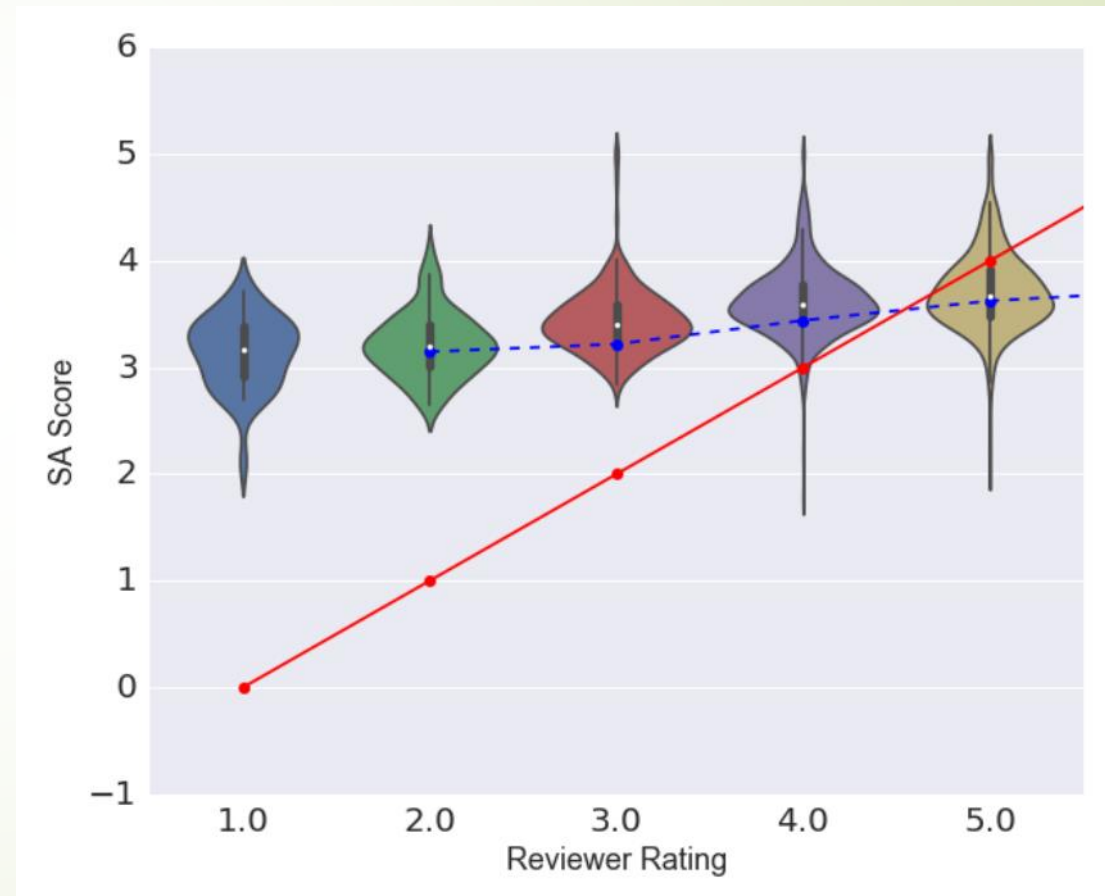
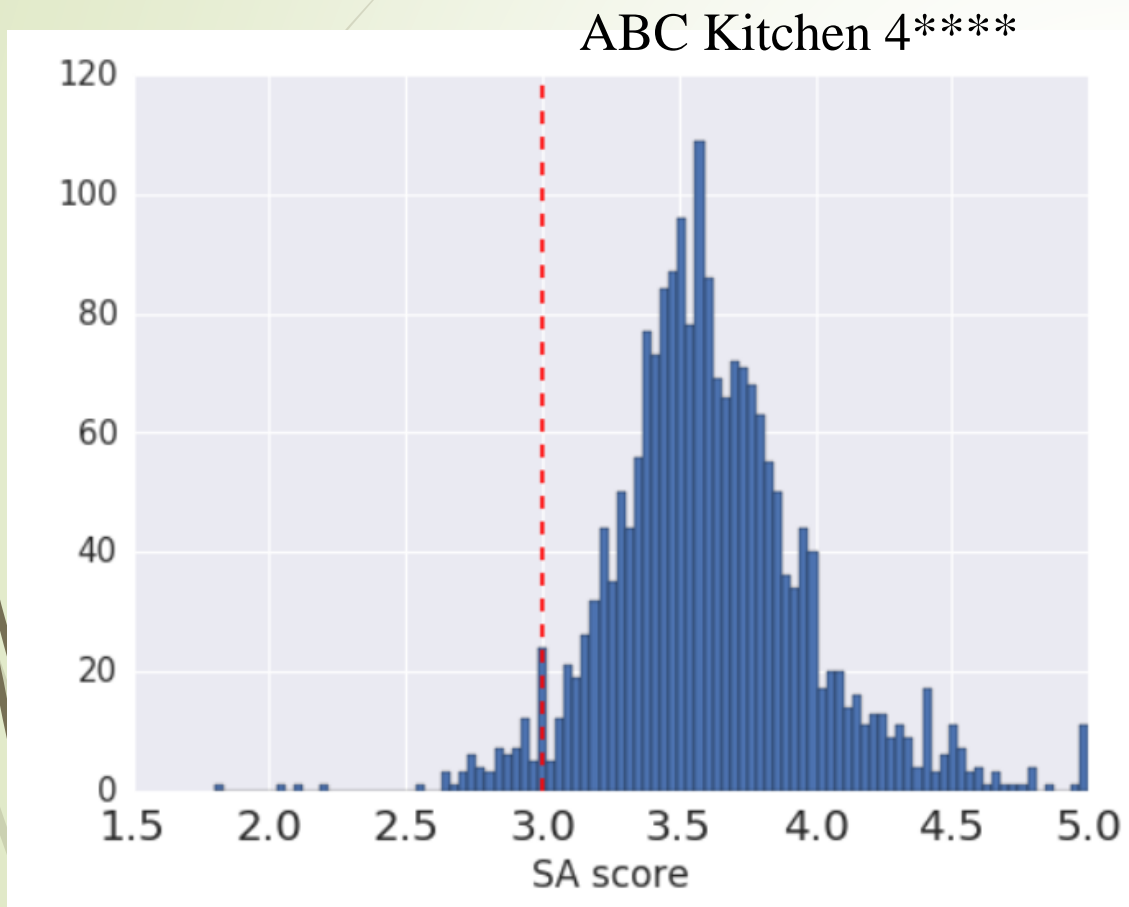
```
: accuacy_sa=[1 for x in sa_score if x>3]
accuacy_sa=sum(accuacy_sa)/len(sa_score)
print('The accuracy of SA prediction is : {}'.format(accuacy_sa))
```

The accuracy of SA prediction is : 0.9589108910891089

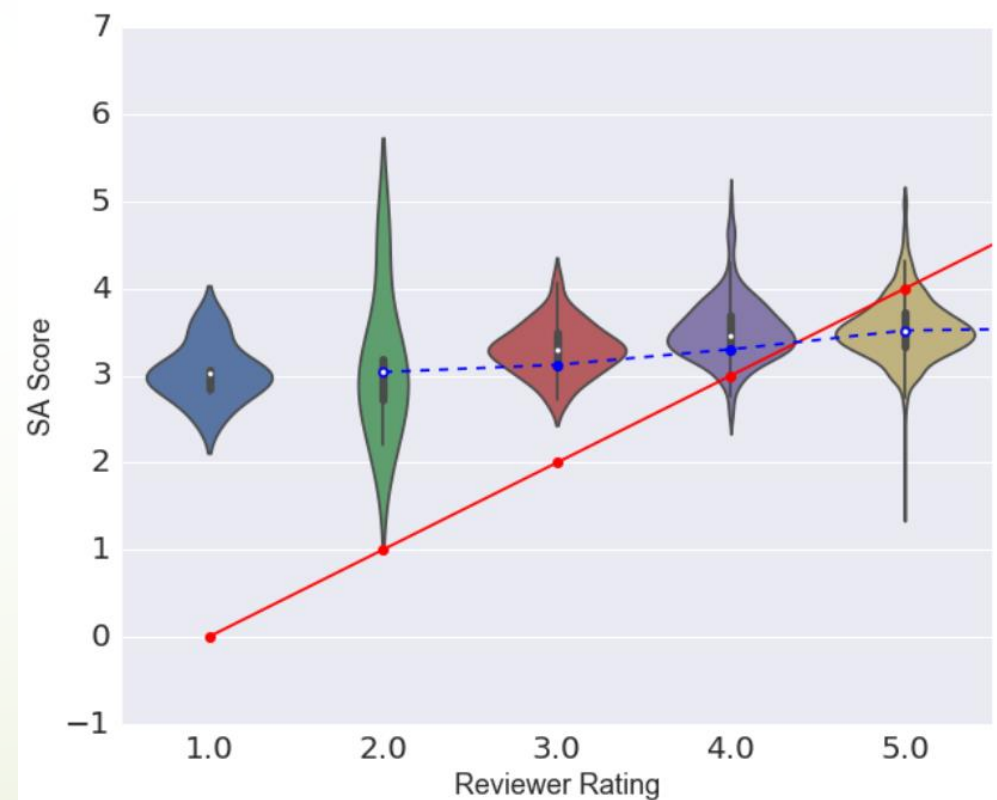
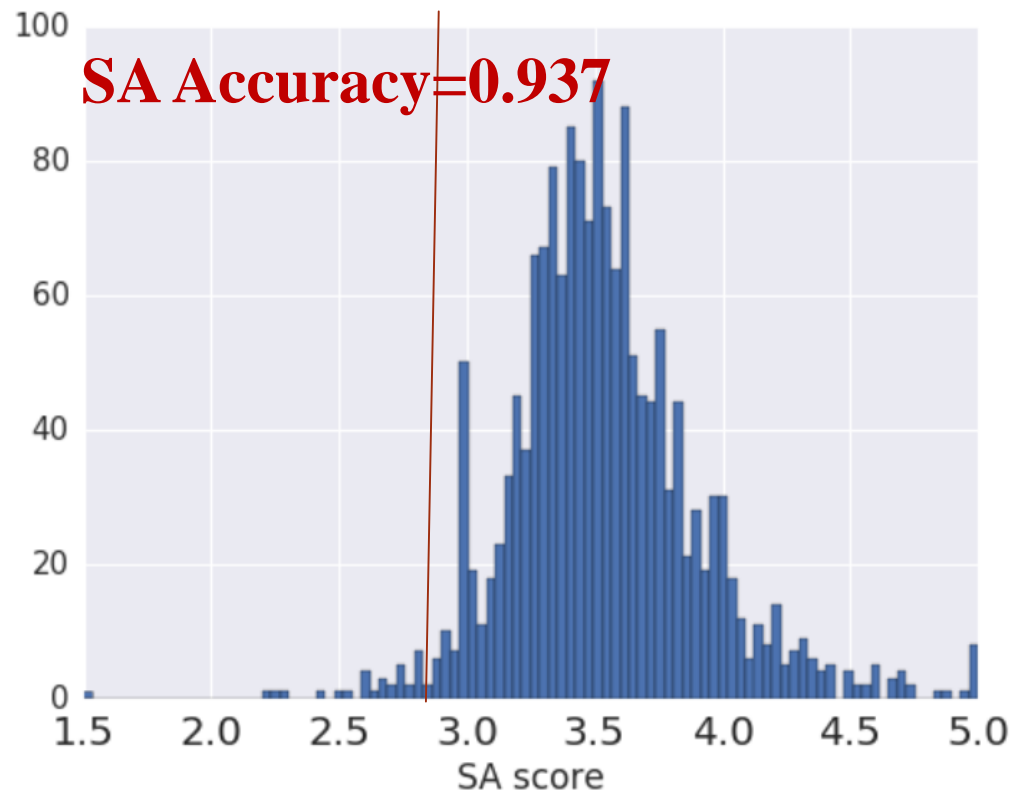
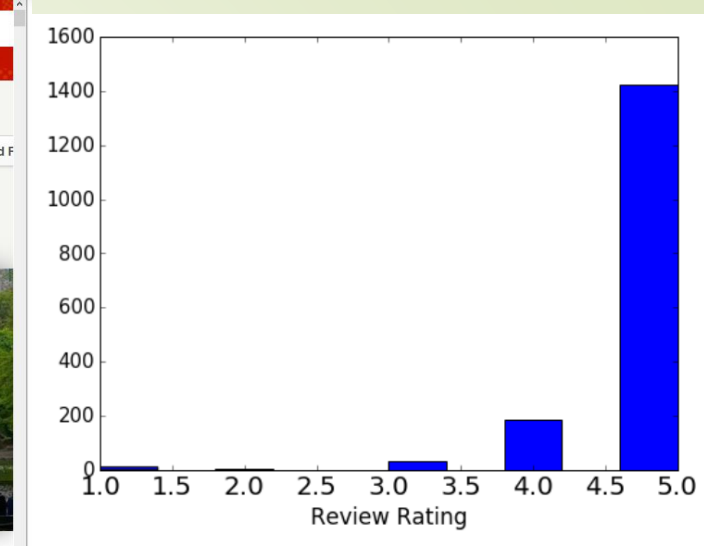
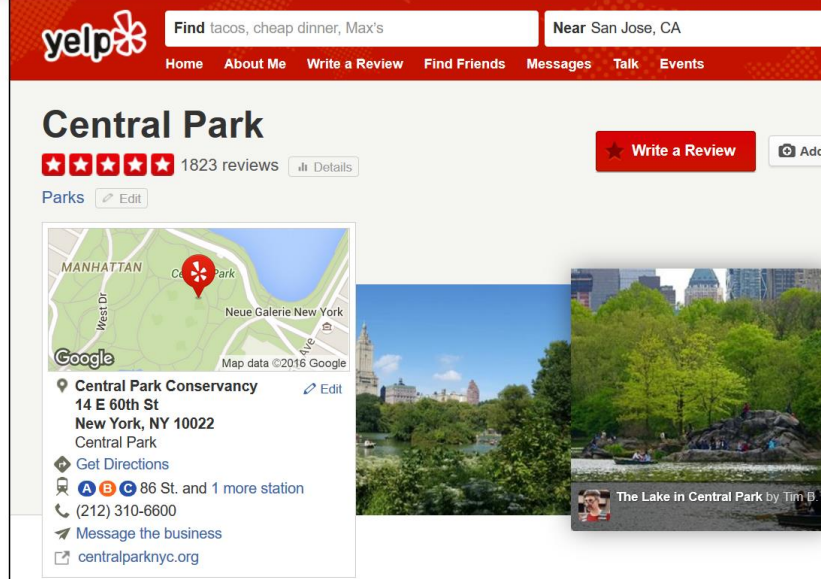
The accuracy of SA prediction is : 0.9589

SA for restaurant review

The accuracy of SA prediction is : 0.9589



Example 2:



Prediction using Navie Baye

```
10]: def features_extra(Features,reviewers):
    yelpFeatures=[]
    for ir,feature in enumerate(Features):
        Words = re.findall(r"[\w']+|[.,!?;]", feature.rstrip())
        wordb=dict([(word.lower(), True) for word in Words])
        if (reviewers.Rating[ir]>=3):
            Words = [wordb, 'pos']
        else:
            Words = [wordb, 'neg']
        yelpFeatures.append(Words)
    return yelpFeatures
## test for greeting is treated as a negative word
def features_extra_mod(Features,reviewers):
    yelpFeatures=[]
    postive_words=['greeting','ultimately','politely']
    for ir,feature in enumerate(Features):
        Words = re.findall(r"[\w']+|[.,!?;]", feature.rstrip())
        wordb=dict([(word.lower(), True) for word in Words])
        pol='neg'
        if (reviewers.Rating[ir]>=3):
            pol='pos'
        for wor in wordb.keys():
            if wor in postive_words:
                print('positiveword corrected={} in reviwer {}'.format(wor,ir))
                pol='pos'
        Words = [wordb, pol]
        yelpFeatures.append(Words)
    return yelpFeatures
```


Prediction of yelp feature

```
import re, math, collections, itertools, os
import nltk, nltk.classify.util, nltk.metrics
from nltk.classify import NaiveBayesClassifier
from nltk.metrics import BigramAssocMeasures
from nltk.probability import FreqDist, ConditionalFreqDist
import random

random.shuffle(yelpFeatures)
Cutoff = int(math.floor(len(yelpFeatures)*3/4))
trainFeatures = yelpFeatures[:Cutoff]
testFeatures = yelpFeatures[Cutoff:]

classifier = NaiveBayesClassifier.train(trainFeatures)
referenceSets = collections.defaultdict(set)
testSets = collections.defaultdict(set)
for i, (features, label) in enumerate(testFeatures):
    referenceSets[label].add(i)
    predicted = classifier.classify(features)
    testSets[predicted].add(i)
print ('train on %d instances, test on %d instances' % (len(trainFeatures), len(testFeatures)))
print ('accuracy:', nltk.classify.util.accuracy(classifier, testFeatures))
classifier.show_most_informative_features(20)
```

train on 4500 instances, test on 1500 instances

accuracy: 0.72

Most Informative Features

apologies = True	neg : pos	=	32.7 : 1.0
horribly = True	neg : pos	=	27.7 : 1.0
ripped = True	neg : pos	=	27.7 : 1.0
mistakes = True	neg : pos	=	27.7 : 1.0
message = True	neg : pos	=	27.7 : 1.0
compensate = True	neg : pos	=	22.6 : 1.0
dismissive = True	neg : pos	=	22.6 : 1.0

soft explanation aggravated apologizing
marketing blind compensate
mistakes hater acting stall unapologetic
exaggeration skills message rudely
agreeing poisoning edible shook sloppy
snotty dumb apologies sarcastic
purchase condescending dismissive allowing
piss miserable remotely substitution
horrible embarrassment

Small number of dataset

train on 1515 instances, test on 505 instances

accuracy: 0.6237623762376238

Most Informative Features

greeting = True	neg : pos	=	38.4 : 1.0
Ultimately = True	neg : pos	=	38.4 : 1.0
tasteless = True	neg : pos	=	29.8 : 1.0
55 = True	neg : pos	=	29.8 : 1.0
Sadly = True	neg : pos	=	29.8 : 1.0
politely = True	neg : pos	=	29.8 : 1.0
WAY = True	neg : pos	=	21.3 : 1.0
photographs = True	neg : pos	=	21.3 : 1.0
2016 = True	neg : pos	=	21.3 : 1.0
liquor = True	neg : pos	=	21.3 : 1.0
act = True	neg : pos	=	21.3 : 1.0
pity = True	neg : pos	=	21.3 : 1.0
Nougatine = True	neg : pos	=	21.3 : 1.0
they'd = True	neg : pos	=	21.3 : 1.0
hell = True	neg : pos	=	21.3 : 1.0
dirty = True	neg : pos	=	21.3 : 1.0
acted = True	neg : pos	=	21.3 : 1.0
Walking = True	neg : pos	=	21.3 : 1.0
improved = True	neg : pos	=	21.3 : 1.0
redeeming = True	neg : pos	=	21.3 : 1.0

➤ Really disappointed to our dining experience tonight. Thursday night, 6:30PM for 2 people. We made reservation 1 month earlier for the birthday dinner. ---Environment---
Dining area: The decoration and ambiance of main area was beautiful and great, HOWEVER, they sat us in the back bar area, which was a passage between ABC Kitchen, ABC Cocina, and abcmkt. It had no decoration at all. While we ate, some people were walking on the stairs above us (we were eating under the edge of stairs. I felt some dust were falling into our dishes from it...). The customers of abcmkt also randomly walked to our area and peaked into our dishes. I felt really uncomfortable about it. I understand they didn't want to waste the space and put more tables to make money, but they should probably make it only for the bar guests, not the customers who have the full dinner.-----.*Sundae: good idea to put salted caramel ice cream and popcorn together. It was tasty but it got very sweet at the end. The portion was big, so it's ideal to share it between 2-3 people. ---Service--- The staff were friendly. However, they were not attentive. It might be a busy night, so our waiter disappeared for a while time to time when we needed him. We waited for 20 mins to order dessert and 30 mins to get the bill. Some of the serving ways were also odd. Most of our dishes were

dropped without any words. **No greeting** (we were still in the middle of conversation), no explaining, no "bon appetit". They just came and left like we were air... If you ask questions, they would answer politely. But it was really awkward. I've been to many high-end or Michelin starred restaurants. By the dining experience tonight, I don't think ABC Kitchen deserved Michelin 3 stars. Foods were good but not excellent, and the service, back bar area, and bathroom were not on the level. Overall, it was a

train on 4500 instances, test on 1500 instances

accuracy: 0.722

Most Informative Features

message = True	neg : pos	=	32.8 : 1.0
poisoning = True	neg : pos	=	32.8 : 1.0
mistakes = True	neg : pos	=	27.7 : 1.0
apologize = True	neg : pos	=	24.8 : 1.0
ripped = True	neg : pos	=	22.7 : 1.0
horrible = True	neg : pos	=	19.7 : 1.0
snotty = True	neg : pos	=	19.7 : 1.0
hater = True	neg : pos	=	17.6 : 1.0
rudely = True	neg : pos	=	17.6 : 1.0
applies = True	neg : pos	=	17.6 : 1.0
increasingly = True	neg : pos	=	17.6 : 1.0
appalling = True	neg : pos	=	17.6 : 1.0
remotely = True	neg : pos	=	17.6 : 1.0
compensate = True	neg : pos	=	17.6 : 1.0
sauna = True	neg : pos	=	17.6 : 1.0
behavior = True	neg : pos	=	17.6 : 1.0
curt = True	neg : pos	=	17.6 : 1.0
updated = True	neg : pos	=	17.6 : 1.0
conservative = True	neg : pos	=	17.6 : 1.0
warmer = True	neg : pos	=	17.6 : 1.0

The following run Yelp prediction with Movie review training data!

The accuracy is pretty high 0.811

```
: Cutoff = int(math.floor(len(Features)*3/4))
trainFeatures1 = Features[:Cutoff]
testFeatures1 = Features[Cutoff:]

referenceSets = collections.defaultdict(set)
testSets = collections.defaultdict(set)
for i, (features, label) in enumerate(testFeatures):
    referenceSets[label].add(i)
    predicted = classifier.classify(features)
    testSets[predicted].add(i)
print ('train on %d instances, test on %d instances' % (len(trainFeatures1), len(testFeatures1)))
print ('accuracy:', nltk.classify.util.accuracy(classifier, testFeatures1))
classifier.show_most_informative_features(20)
```

train on 1515 instances, test on 505 instances

accuracy: 0.8118811881188119

Most Informative Features

magnificent = True	pos : neg	=	15.0 : 1.0
outstanding = True	pos : neg	=	13.6 : 1.0
insulting = True	neg : pos	=	13.0 : 1.0
vulnerable = True	pos : neg	=	12.3 : 1.0
ludicrous = True	neg : pos	=	11.8 : 1.0
avoids = True	pos : neg	=	11.7 : 1.0
uninvolving = True	neg : pos	=	11.7 : 1.0
astounding = True	pos : neg	=	10.3 : 1.0
fascination = True	pos : neg	=	10.3 : 1.0
idiotic = True	neg : pos	=	9.8 : 1.0
affecting = True	pos : neg	=	9.7 : 1.0
symbol = True	pos : neg	=	9.7 : 1.0
hated = True	neg : pos	=	9.0 : 1.0



Summary

- The Yelp review scarping and analysis using Python package
 - Sentiment Analysis for positive and negative prediction show good agreement (90%) with review rating
 - Supervised machine learning (Naive Bayes here) has a prediction accuracy is about 72% with 6000 reviewers
 - Further improvements with bigram and other technics.
- 