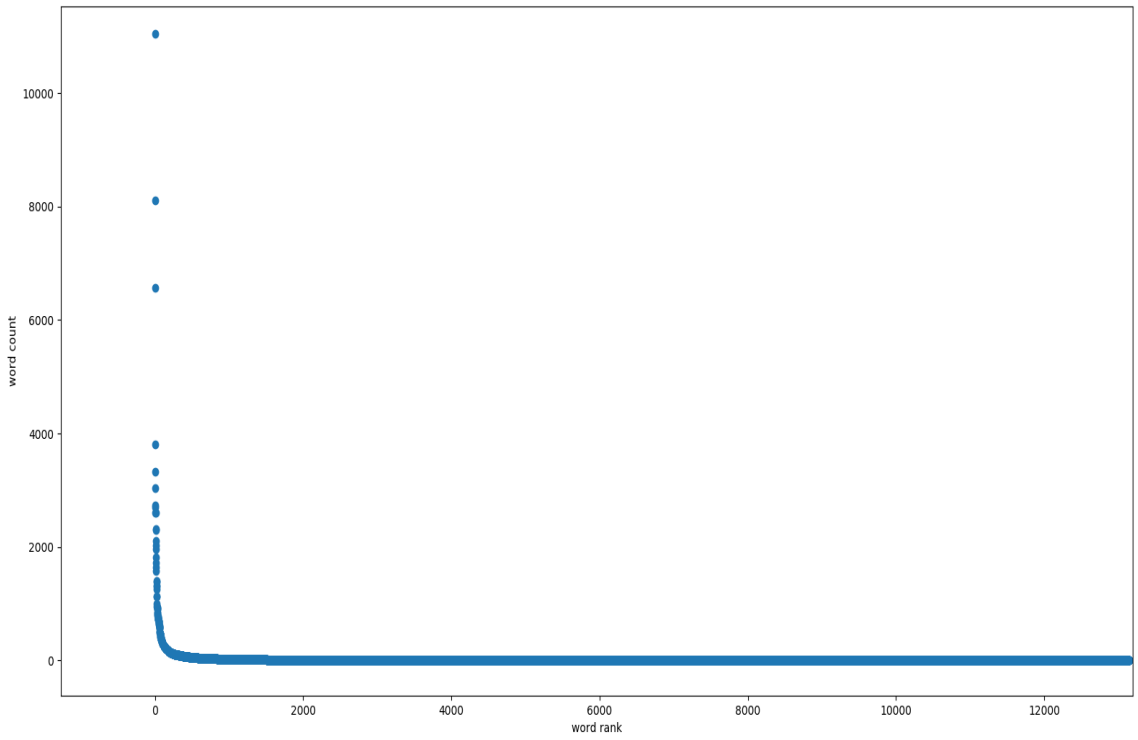
1.Distribution Graph:



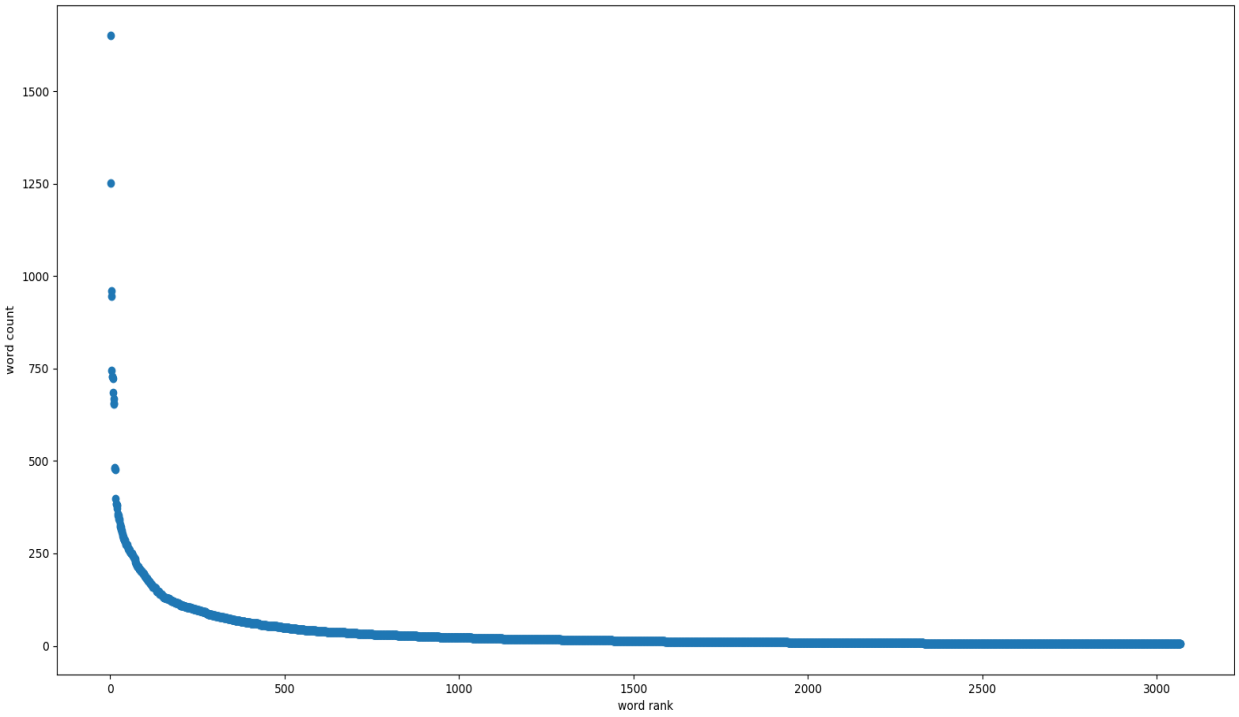
2. Identify the stop words:

Stopping words:
who, has, then, her, more, said, got, did, which, by, after, even, only, do, been, us, your, what, can, or, them, about, go, will, here, their, just, an, up, back, she, our, get, service, would, out, when, all, if, he, as, there, be, are, were, at, me, had, but, have, on, with, you, we, this, they, that, is, my, in, for, of, it, was, to, and, the
(I picked these words from most commonly occurred words, but remove some word that have strong meaning such as “like”, “good”, “great”, etc)

Minimum count: 5

Maximum occurrence ratio: 0.1

3.Distribution graph after removing stop word, min-word occurance, and max document frequency:



4.Code Snippets:

Convert to bag-of-words:

```
def getBagOfWords(documents, stopWords, minThreshold, maxThreshold):
    vectorizer = CountVectorizer()
    vectorizer.stop_words = stopWords
    vectorizer.min_df = minThreshold
    vectorizer.max_df = maxThreshold
    X = vectorizer.fit_transform(documents)
    return vectorizer, X.toarray()
```

Nearest-neightbours with cos-distance:

```
def getNNModel(wordVectors):
    neigh = NearestNeighbors()
    neigh.metric = 'cosine'
    neigh.fit(wordVectors)
    return neigh
```

5.
5 original reviews and its repective distance(Note the meaning of distance is different from the book, the range here is from 0 to 1 where 0 represent most simliar documents and 1 represent least similar documents):
`0.27239312, 0.42264973, 0.54250429, 0.5527864, 0.6619383`

service was horrible came with a major attitude. payed 30 for lasagna and was no where worth it. won't ever be going back and will never recommend this place. was treated absolutely horrible. horrible.
horrible horrible horrible!!! avoid at all costs!!!

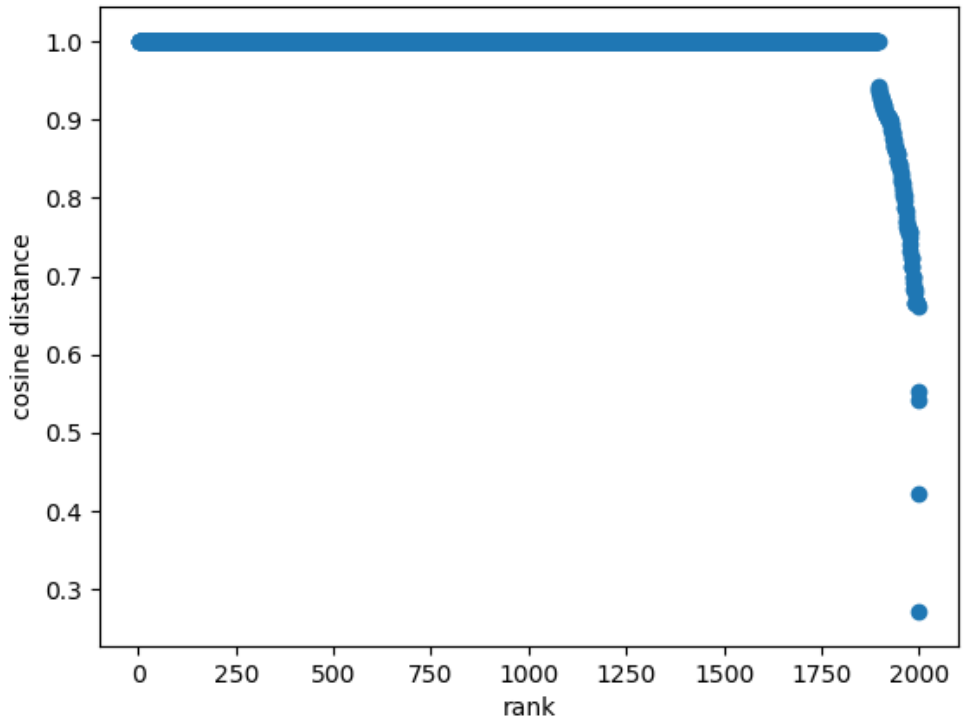
i had some work done at swing shift auto and i was helped by keith. he was very arrogant and had little time for me. i just needed new brake discs and pads. i was overcharged, the repairs took two days, and when i got home i noticed that the discs had not been replaced, only the pads!!!!

total ripofff!!! never go here, please!!!
horrible service, horrible customer service, and horrible quality of service! do not waste your time or money using this company for your pool needs. dan (602)363-8267 broke my pool filtration system and left it in a nonworking condition. he will not repair the issue he caused, and told me to go somewhere else.

save yourself the hassle, there are plenty of other quality pool companies out there.

take care!
i was in there a few weeks ago, the lady who took my order dene was horrible....not only did she give me the wrong change but had a terrible attitude and also put in my order wrong not to mention it looked as if she was hungover
if i could give this a negative star i would...what a horrible representation of this place
horrible experience! got there at 1 am and the front desk worker wasn't there. the lights were turned off so i called with the after hours phone. after 10 minutes, someone let us in and we stood at the counter and he finally walked up, then told us we couldn't check in for an hour because the computer was down. finally got to our room 2 hours later! horrible experience!

6. Query results:
(Note the meaning of distance is different from the book, the range here is from 0 to 1 where 0 represent most simliar documents and 1 represent least similar documents):



If threshold is at 0.9, three are 157 documents with similar meaning to “horriable customer service”

7. Accuracy with threshld 0.5

Code for creating classifier

```
#split data into test and train
test, testLabels, train, trainLabels = pData.splitData(wordVectors, testProb,
stars)
logitModel = LogisticRegression().fit(train, trainLabels)

#get accuracy on training and testing data
trainAcc = logitModel.score(train, trainLabels)
testAcc = logitModel.score(test, testLabels)
```

Train accuracy: `0.9988883`

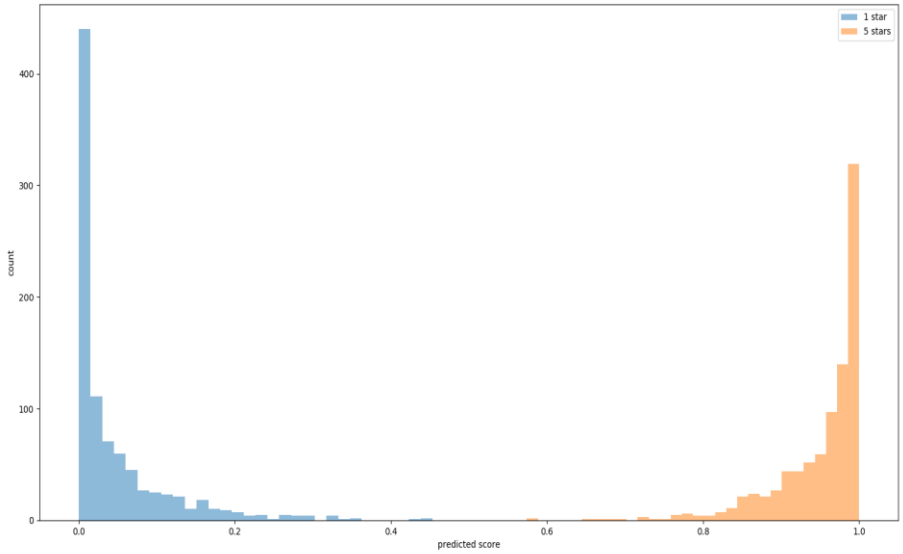
Test accuracy: `0.9203980`

8. Predicted scores:

Code for plot:

```
one, five, prob = [],[], logitModel.predict_proba(data)
for p in prob:
    if p[1] <= 0.5:
        one.append(p[1])
    else:
        five.append(p[1])
plt.hist(one,30, alpha=0.5, label='1 star')
plt.hist(five,30,alpha=0.5, label='5 stars' )
plt.xlabel('predicted score')
plt.ylabel('count')
plt.legend(loc='north')
plt.show()
```

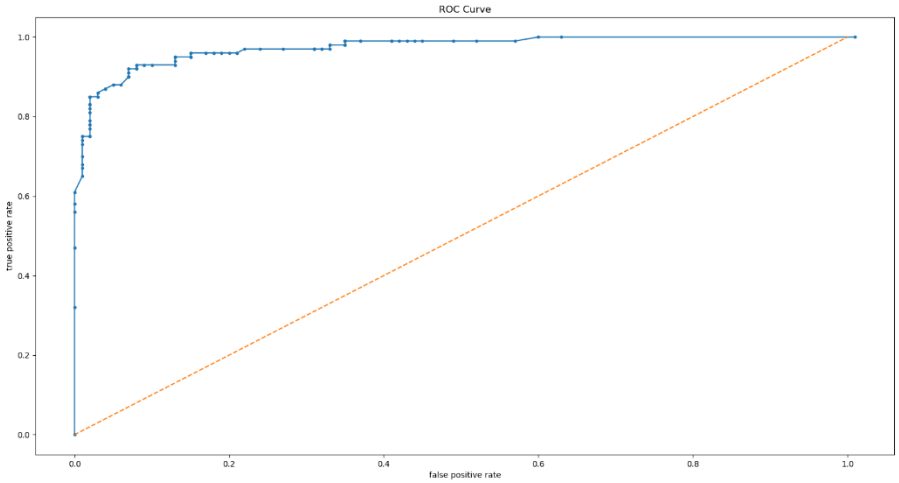
Plot:



9. Accuracy again and curve

Using new threshold of 0.52, I choose the midpoint between the probabilities of the largest star 1 probability and smallest star 5 probability from the probabilities of star 5.
New Accuracy for training: `0.99889197`
New Accuracy for testing: `0.92307692`
There is minor improvement compare to previous threshold of 0.5

ROC curve:



10. Best threshold

The best threshold is 0.4444445

I test with 100 thresholds equally space in range(0,1), for each threshold I found the true positive rate and false positive rate. Then for each rates, I subtract true postive rate from false positive rate. The threshold that give the maximum difference is my answer, which maximuize the true positive rate while minimize the false positive rate.