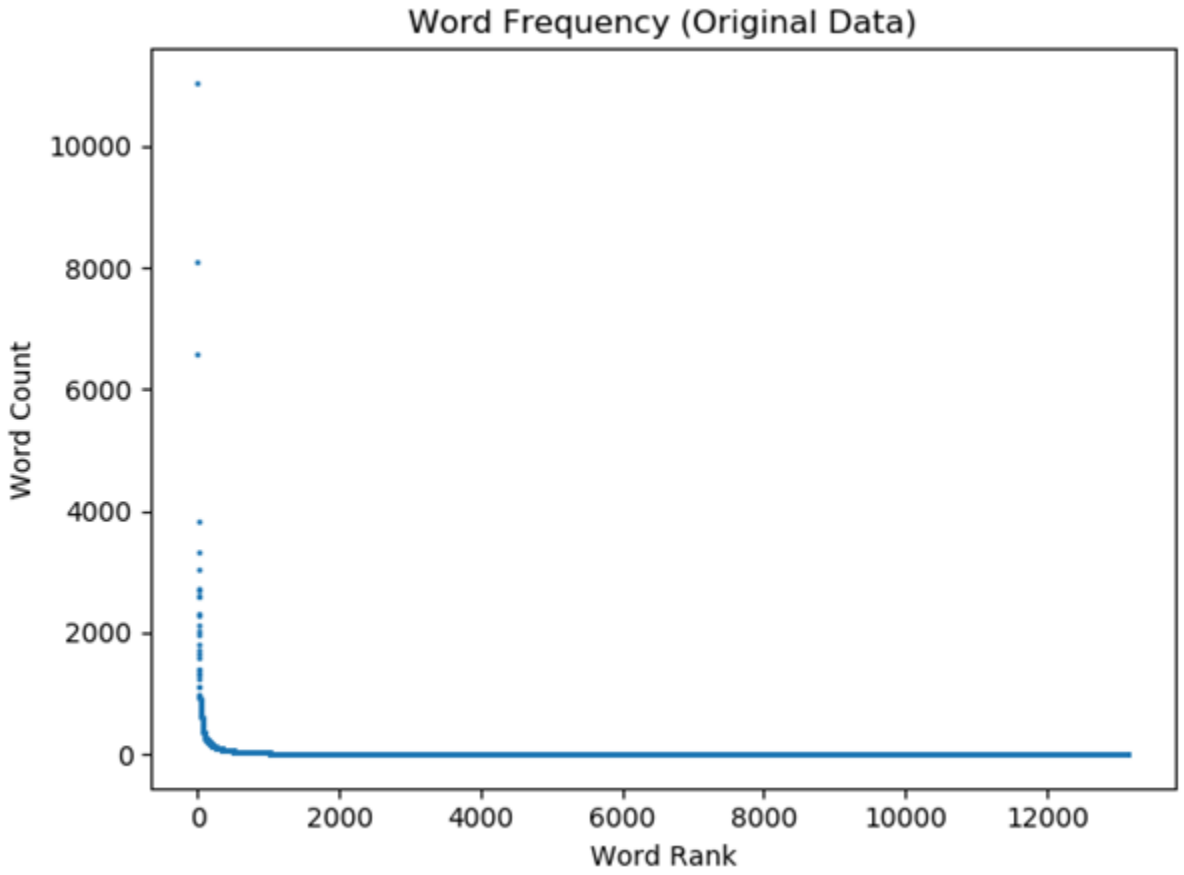
Page 1 Distribution graph (5 points)

Show the distribution graph of words counts vs word rank.



Page 2 Identify the stop words (5 points)

List of Stop words: (in descending order by frequency)

['the', 'and', 'to', 'was', 'it', 'of', 'for', 'in', 'my', 'is', 'that', 'they', 'this', 'we', 'you', 'with', 'on', 'not', 'have', 'but', 'had', 'me', 'at', 'so', 'were', 'are', 'be', 'place', 'food', 'there', 'as', 'he', 'if', 'all', 'when', 'out', 'would']

Number of stop words = 37

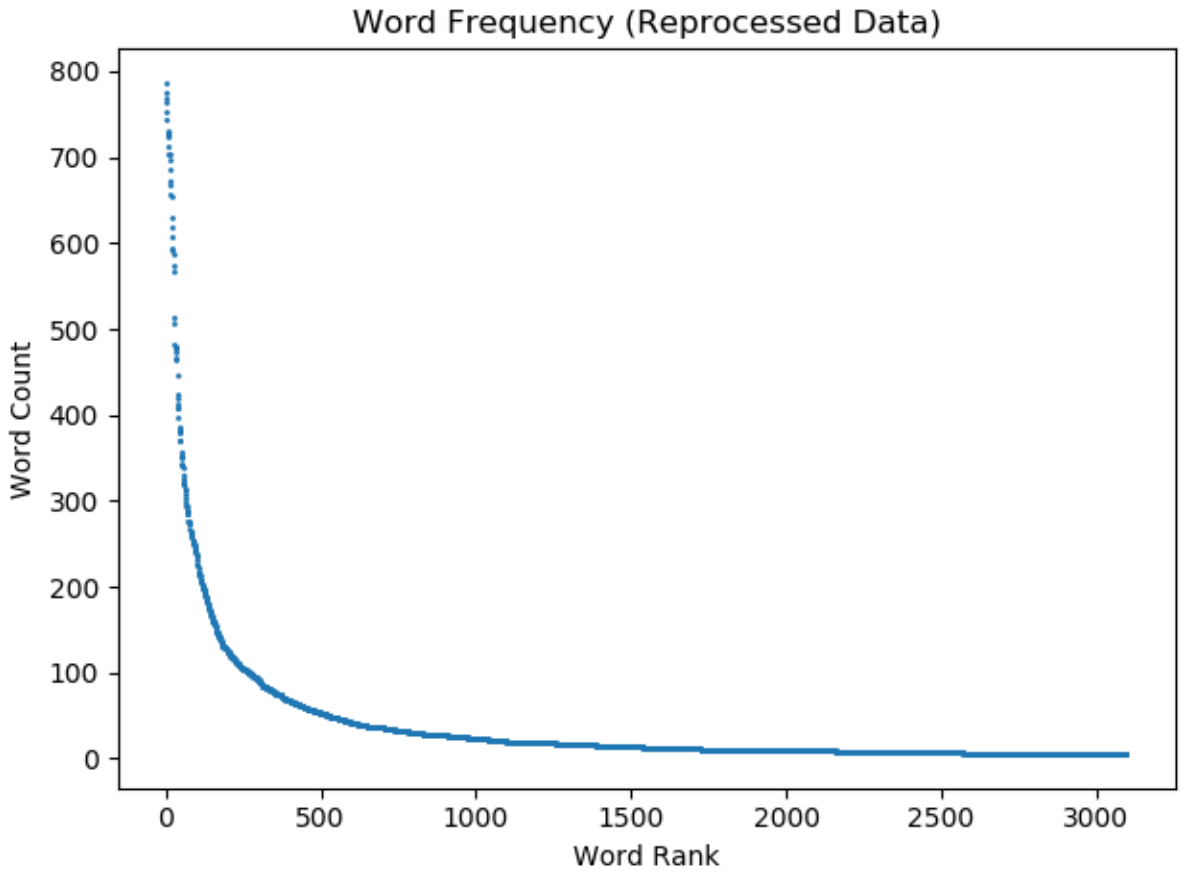
Frequency Threshold: 800

We manually looked at the most frequent words. These were the commonly used English words. All words which appeared more than 800 times were chosen as stop words.

Page 3 Distribution graph again (5 points)

After choosing the stop words, show the distribution graph of words counts vs word rank.

- Max document frequency threshold = max_df = 0.9 (float value representing a fraction)
- Minimum word occurrence = min_df = 5 (integer value)
- The data reprocessed using the stop-words list, max_df and max_df in sklearn CountVectorizer to cull less useful words.
- Now the features were stored as bag_of_words. The length of this list was 3098.



Page 4 Code snippets (15 points)

Code for bag-of-words formulation using chosen stop words:

```
# select stop words
freq_thresh = 800
No_stopwords = len(all_words[frequency>freq_thresh])
stopwords = all_words[indexbyfreq[:No_stopwords]]

# Re-Process Data
vectorizer2 = CountVectorizer(stop_words=list(stopwords),max_df=0.9,min_df=5)
X2 = vectorizer2.fit_transform(X).toarray()
frequency2 = np.sum(X2,axis=0)
bag_of_words = vectorizer2.get_feature_names()
```

Code for nearest-neighbors with cos-distance

```
target=['horrible custome service']
target=vectorizer2.transform(target)
sparse=vectorizer2.transform(X)
score=cosine_similarity(target,sparse)
sorted_index= np.fliplr((np.argsort(score)))
#setting so it can print out everything
np.set_printoptions(threshold=np.inf)
max_5=[sorted_index[0,i] for i in range(5)]
for j,i in enumerate(max_5):
    print("Review",j+1,":",X[i][:200])
    print("Score:",score[0,i])
```

Page 5 Reviews with score (10 points)

Review 1:

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

Score: 0.5547001962252291

Review 2:

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 syste

Score: 0.5262348115842175

Review 3:

rogers ...

1) is over priced

2) have horrible customer service

3) faulty and incorrect billing

4) poor customer service

5) not enough options

6) never arrive for an appointment

Score: 0.47434164902525683

Review 4:

the service is horrible. it's not bad inside, but really one of the most annoying clubs in vegas. i'm all for vegas clubs, but service here sucks.

Score: 0. 4629100498862757

Review 5: horrible service....what a mess upon ordering and paying. where is the manager on duty to fix this!

Score: 0. 42640143271122083

Page 6 Query results (10 points)

Show your document results and explain the reasons that you choose them.

Based on results of page 5, all of the documents found to be matching with the query. Moreover, scores were similar as well. So we decided to chose all the document results.

Review 1:

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

Score: 0.5547001962252291

Review 2:

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 syste

Score: 0.5262348115842175

Review 3:

rogers ...

1) is over priced

2) have horrible customer service

3) faulty and incorrect billing

4) poor customer service

5) not enough options

6) never arrive for an appointment

Score: 0.47434164902525683

Review 4:

the service is horrible. it's not bad inside, but really one of the most annoying clubs in vegas. i'm all for vegas clubs, but service here sucks.

Score: 0. 4629100498862757

Review 5: horrible service....what a mess upon ordering and paying. where is the manager on duty to fix this!

Score: 0. 42640143271122083

Page 7 Accuracy with threshold 0.5 (10 points)

Show your code for creating classifier. Report the accuracy on train and test dataset with threshold 0.5.

```
"""
Part 3: Classification with Logistic Regression
"""
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, accuracy_score

# train classifier and determine accuracies
X_train, X_test, y_train, y_test = train_test_split(X2, y, test_size = 0.1, random_state = 0)
classifier = LogisticRegression(random_state = 0).fit(X_train, y_train)
acc_train = classifier.score(X_train,y_train)
acc_test  = classifier.score(X_test,y_test)
print("Train Set Accuracy:",acc_train)
print("Test Set Accuracy:",acc_test)
```

Train Set Accuracy: 0.9994444444444445

Test Set Accuracy: 0.92

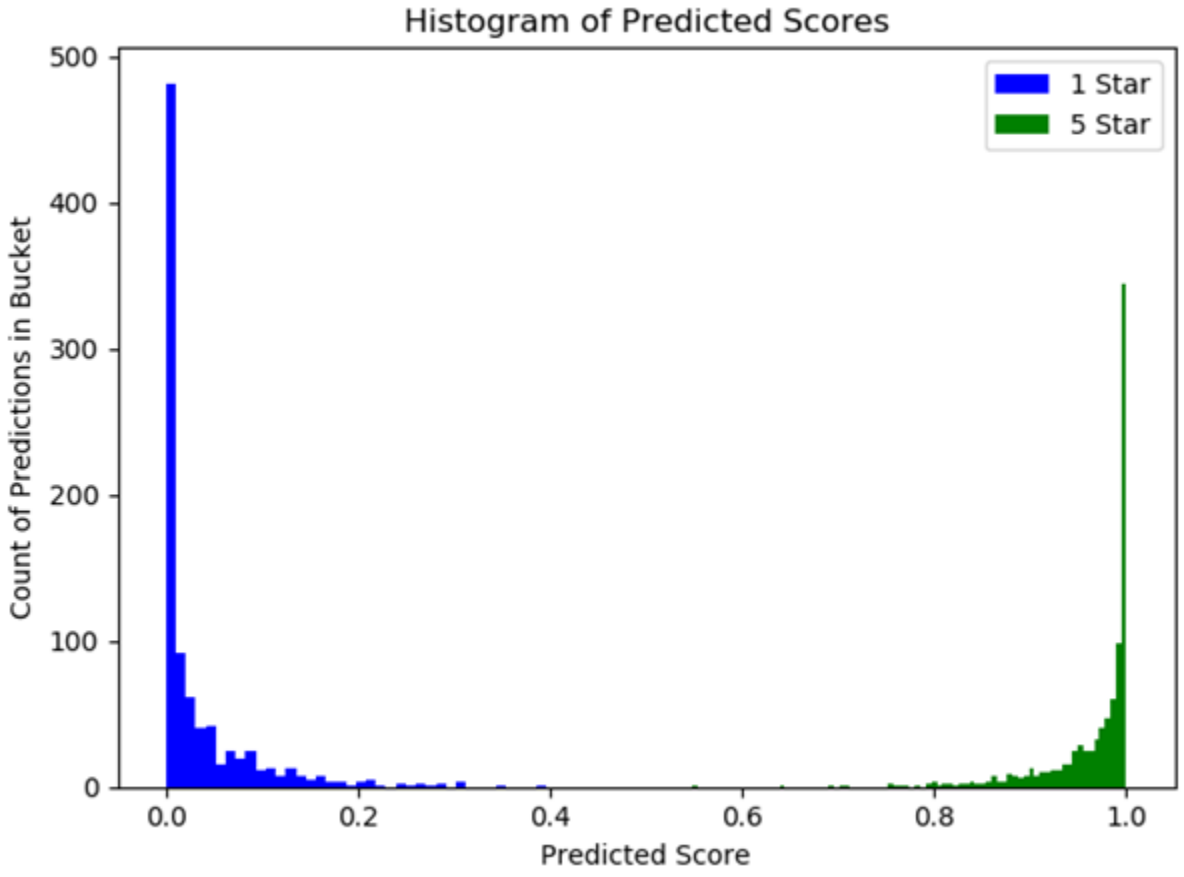
Page 8 Predicted scores (10 points)

Show your code for plotting predicted scores and show the figure.

Code:

We binarized the y labels into 0 and 1. So here 0 means 1 star and 1 means 5 star review.

```
# Plot Histogram of Predicted Scores
pred_score_train = classifier.predict_proba(X_train)[:,1]
plt.figure(3)
plt.hist(pred_score_train[y_train==0],bins=80,label='1 Star',color='b')
plt.hist(pred_score_train[y_train==1],bins=80,label='5 Star',color='g')
plt.title('Histogram of Predicted Scores')
plt.ylabel('Count of Predictions in Bucket')
plt.xlabel('Predicted Score')
plt.legend()
plt.show()
```



Page 9 Accuracy again and curve (20 points)

Report the accuracy on train and test dataset with a different threshold. Explain why you choose that threshold.

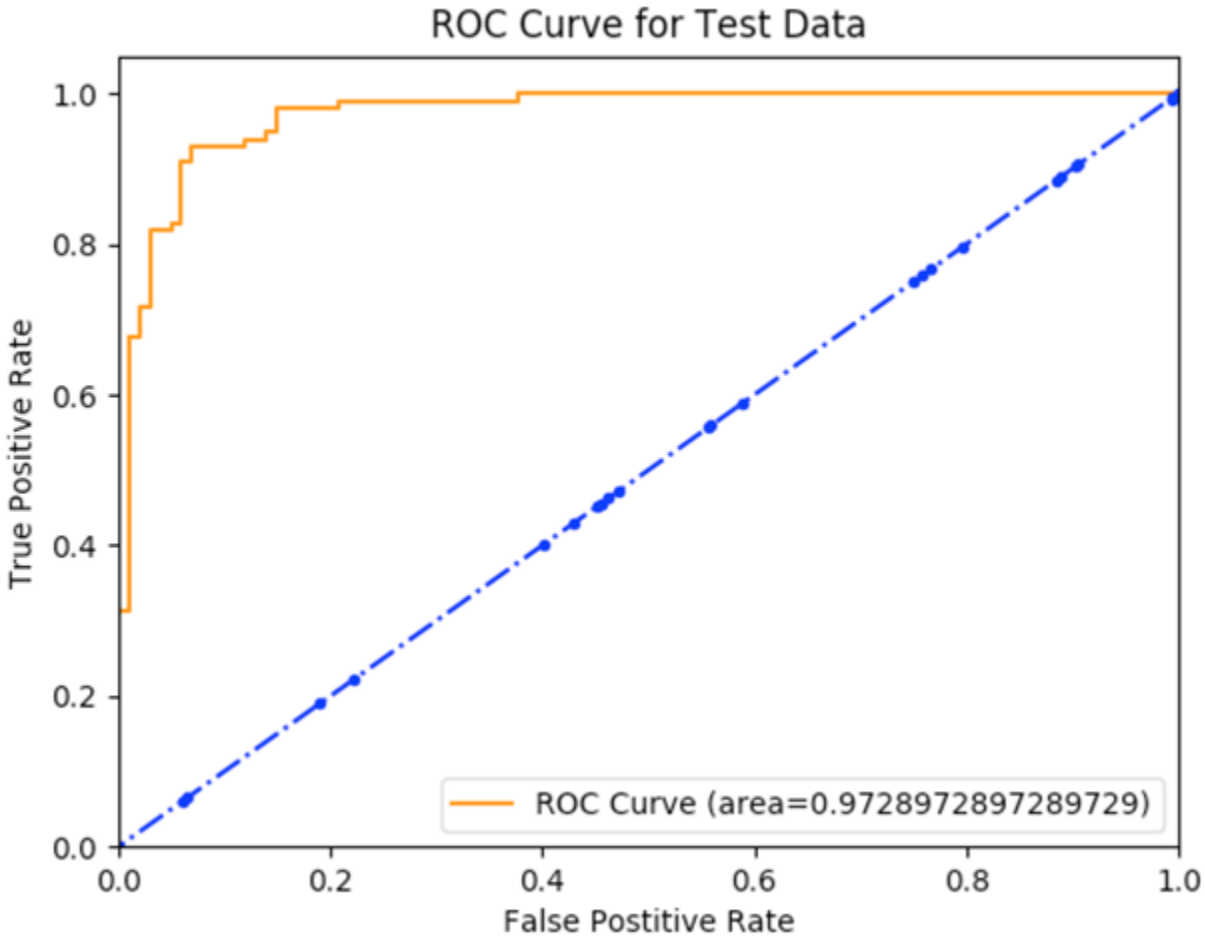
We got a slightly better test accuracy with a threshold hold of 0.55. We chose this value because it was giving the best separation between two classes in the histogram of predicted scores. .

New_threshold = 0.55

Train Set Accuracy: 0.9988888888888889

Test Set Accuracy: 0.93

Plot the ROC curve.



Page 10 Best threshold (10 points)

Choose the threshold that minimizes false positives while maximizing true positives. Explain your reason.

From the ROC curve, we look for the top left corner where the ROC curve makes a sudden change. That is our optimal point for choosing false positive and true positive rates. From ROC curve plot, these values can be judged.

Best Threshold = 0.4

Corresponding False Positive Rate = 0.14

Corresponding True Positive Rate = 0.97