In [1]:

```
cd D:\karan\amazon_review_database
```

```
D:\karan\amazon_review_database
```

In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import sqlite3
```

In [3]:

```
con=sqlite3.connect('final.sqlite')
```

In [4]:

```
## We had cleansed and saved the final data for TSNE assignment. We use the same for this assignme
nt.

review_data=pd.read_sql_query("""Select * from  Reviews""",con)
```

In [5]:

```
review_data.shape
```

Out[5]:

```
(364171, 12)
```

In [6]:

```
review_data=review_data[review_data['ProductId'].str.startswith('B')]
print(review_data.shape)
```

```
(364131, 12)
```

In [7]:

```
sampled_data=review_data[['Time','Score','CleanedText']].sample(5000)
```

In [8]:

```
sampled_data=sampled_data.sort_values('Time')
```

In [9]:

```
sampled_data['Score']=sampled_data['Score'].replace(('positive'),(1))
```

In [10]:

```
sampled_data['Score']=sampled_data['Score'].replace(('negtive'),(0))
```

In [11]:

```
x_train=sampled_data['CleanedText'].iloc[:3500]
```

In [12]:

```
y_train=sampled_data['Score'].iloc[:3500]
```

In [13]:

```
x_test=sampled_data['CleanedText'].iloc[3500:]
```

In [14]:

```
y_test=sampled_data['Score'].iloc[3500:]
```

Word2Vec

training data

In [15]:

```
from gensim.models import Word2Vec
```

In [16]:

```
sent_list=[]
for sen in x_train:
    x=sen.split()
    sent_list.append(x)
```

In [17]:

```
len(sent_list)
```

Out[17]:

```
3500
```

In [18]:

```
w2v_model=Word2Vec(sent_list,size=20,workers=8,min_count=1)
```

In [19]:

```
word2vec_word=list(w2v_model.wv.vocab)
```

In [20]:

```
avg_w2vec_sent=[]
for sent in sent_list:
    cnt_words=0
    sent_vect=np.zeros(20)
    for word in sent:
        if word in word2vec_word:
            word_vec=w2v_model.wv[word]
            cnt_words+=1
            sent_vect+=word_vec
    if cnt_words !=0:
        sent_vect/=cnt_words
    avg_w2vec_sent.append(sent_vect)


print(len(avg_w2vec_sent))
print(len(avg_w2vec_sent[0]))
```

```
3500
20
```

In [21]:

```
train_vector=avg_w2vec_sent
```

test data

In [22]:

```
sent_list=[]
for sen in x_test:
    x=sen.split()
    sent_list.append(x)
```

In [23]:

```
len(sent_list)
```

Out[23]:

```
1500
```

In [24]:

```
w2v_model=Word2Vec(sent_list,size=20,workers=8,min_count=1)
```

In [25]:

```
avg_w2vec_sent=[]
for sent in sent_list:
    cnt_words=0
    sent_vect=np.zeros(20)
    for word in sent:
        if word in word2vec_word:
            word_vec=w2v_model.wv[word]
            cnt_words+=1
            sent_vect+=word_vec
    if cnt_words !=0:
        sent_vect/=cnt_words
    avg_w2vec_sent.append(sent_vect)


print(len(avg_w2vec_sent))
print(len(avg_w2vec_sent[0]))
```

```
1500
20
```

In [ ]:

```

```

In [26]:

```
test_vector=avg_w2vec_sent
```

Finding the optimal C

In [27]:

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score, f1_score, precision_score,
recall_score,roc_curve,roc_auc_score,auc
```

In [28]:

```
c=[0.0001,0.001,0.01,0.1,1,10,100,1000]
```

In [29]:

```
c
```

Out[29]:

```
[0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000]
```

In [30]:

```
param_grid=[
    {'penalty':['l2'],
    'C' : [0.00001,0.0001,0.001,0.01,1,10,100,1000],
    'max_iter':[100,500,1000,2500,5000,8000]
    }
]
```

In [31]:

```
parameters={'C':c}
model=LogisticRegression(class_weight='balanced',penalty='l2',solver='lbfgs')
classifier=GridSearchCV(model,param_grid=param_grid,scoring='roc_auc',cv=10)
classifier.fit(train_vector, y_train)
```

```
C:\anaconda\lib\site-packages\sklearn\linear_model\logistic.py:947: ConvergenceWarning: lbfgs
failed to converge. Increase the number of iterations.
  "of iterations.", ConvergenceWarning)
C:\anaconda\lib\site-packages\sklearn\linear_model\logistic.py:947: ConvergenceWarning: lbfgs
failed to converge. Increase the number of iterations.
  "of iterations.", ConvergenceWarning)
C:\anaconda\lib\site-packages\sklearn\linear_model\logistic.py:947: ConvergenceWarning: lbfgs
failed to converge. Increase the number of iterations.
  "of iterations.", ConvergenceWarning)
C:\anaconda\lib\site-packages\sklearn\linear_model\logistic.py:947: ConvergenceWarning: lbfgs
failed to converge. Increase the number of iterations.
  "of iterations.", ConvergenceWarning)
C:\anaconda\lib\site-packages\sklearn\linear_model\logistic.py:947: ConvergenceWarning: lbfgs
failed to converge. Increase the number of iterations.
  "of iterations.", ConvergenceWarning)
C:\anaconda\lib\site-packages\sklearn\linear_model\logistic.py:947: ConvergenceWarning: lbfgs
failed to converge. Increase the number of iterations.
  "of iterations.", ConvergenceWarning)
C:\anaconda\lib\site-packages\sklearn\linear_model\logistic.py:947: ConvergenceWarning: lbfgs
failed to converge. Increase the number of iterations.
  "of iterations.", ConvergenceWarning)
C:\anaconda\lib\site-packages\sklearn\linear_model\logistic.py:947: ConvergenceWarning: lbfgs
failed to converge. Increase the number of iterations.
  "of iterations.", ConvergenceWarning)
C:\anaconda\lib\site-packages\sklearn\linear_model\logistic.py:947: ConvergenceWarning: lbfgs
failed to converge. Increase the number of iterations.
  "of iterations.", ConvergenceWarning)
C:\anaconda\lib\site-packages\sklearn\linear_model\logistic.py:947: ConvergenceWarning: lbfgs
failed to converge. Increase the number of iterations.
  "of iterations.", ConvergenceWarning)
C:\anaconda\lib\site-packages\sklearn\linear_model\logistic.py:947: ConvergenceWarning: lbfgs
failed to converge. Increase the number of iterations.
  "of iterations.", ConvergenceWarning)
C:\anaconda\lib\site-packages\sklearn\linear_model\logistic.py:947: ConvergenceWarning: lbfgs
failed to converge. Increase the number of iterations.
  "of iterations.", ConvergenceWarning)
C:\anaconda\lib\site-packages\sklearn\linear_model\logistic.py:947: ConvergenceWarning: lbfgs
failed to converge. Increase the number of iterations.
  "of iterations.", ConvergenceWarning)
C:\anaconda\lib\site-packages\sklearn\linear_model\logistic.py:947: ConvergenceWarning: lbfgs
failed to converge. Increase the number of iterations.
  "of iterations.", ConvergenceWarning)
C:\anaconda\lib\site-packages\sklearn\linear_model\logistic.py:947: ConvergenceWarning: lbfgs
failed to converge. Increase the number of iterations.
```

```
  "of iterations.", ConvergenceWarning)
C:\anaconda\lib\site-packages\sklearn\linear_model\logistic.py:947: ConvergenceWarning: lbfgs
failed to converge. Increase the number of iterations.
  "of iterations.", ConvergenceWarning)
C:\anaconda\lib\site-packages\sklearn\linear_model\logistic.py:947: ConvergenceWarning: lbfgs
failed to converge. Increase the number of iterations.
  "of iterations.", ConvergenceWarning)
C:\anaconda\lib\site-packages\sklearn\linear_model\logistic.py:947: ConvergenceWarning: lbfgs
failed to converge. Increase the number of iterations.
  "of iterations.", ConvergenceWarning)
C:\anaconda\lib\site-packages\sklearn\linear_model\logistic.py:947: ConvergenceWarning: lbfgs
failed to converge. Increase the number of iterations.
  "of iterations.", ConvergenceWarning)
C:\anaconda\lib\site-packages\sklearn\linear_model\logistic.py:947: ConvergenceWarning: lbfgs
failed to converge. Increase the number of iterations.
  "of iterations.", ConvergenceWarning)
C:\anaconda\lib\site-packages\sklearn\linear_model\logistic.py:947: ConvergenceWarning: lbfgs
failed to converge. Increase the number of iterations.
  "of iterations.", ConvergenceWarning)
C:\anaconda\lib\site-packages\sklearn\linear_model\logistic.py:947: ConvergenceWarning: lbfgs
failed to converge. Increase the number of iterations.
  "of iterations.", ConvergenceWarning)
C:\anaconda\lib\site-packages\sklearn\linear_model\logistic.py:947: ConvergenceWarning: lbfgs
failed to converge. Increase the number of iterations.
  "of iterations.", ConvergenceWarning)
C:\anaconda\lib\site-packages\sklearn\linear_model\logistic.py:947: ConvergenceWarning: lbfgs
failed to converge. Increase the number of iterations.
  "of iterations.", ConvergenceWarning)
C:\anaconda\lib\site-packages\sklearn\linear_model\logistic.py:947: ConvergenceWarning: lbfgs
failed to converge. Increase the number of iterations.
  "of iterations.", ConvergenceWarning)
C:\anaconda\lib\site-packages\sklearn\linear_model\logistic.py:947: ConvergenceWarning: lbfgs
failed to converge. Increase the number of iterations.
  "of iterations.", ConvergenceWarning)
```

Out[31]:

```
GridSearchCV(cv=10, error_score='raise-deprecating',
             estimator=LogisticRegression(C=1.0, class_weight='balanced',
                                          dual=False, fit_intercept=True,
                                          intercept_scaling=1, l1_ratio=None,
                                          max_iter=100, multi_class='warn',
                                          n_jobs=None, penalty='l2',
                                          random_state=None, solver='lbfgs',
                                          tol=0.0001, verbose=0,
                                          warm_start=False),
             iid='warn', n_jobs=None,
             param_grid=[{'C': [1e-05, 0.0001, 0.001, 0.01, 1, 10, 100, 1000],
                          'max_iter': [100, 500, 1000, 2500, 5000, 8000],
                          'penalty': ['l2']}],
             pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
             scoring='roc_auc', verbose=0)
```

In [32]:

```
print(classifier.best_estimator_)
print(classifier.score(train_vector,y_train))
```

```
LogisticRegression(C=1000, class_weight='balanced', dual=False,
                   fit_intercept=True, intercept_scaling=1, l1_ratio=None,
                   max_iter=500, multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
0.7758151090114223
```

In [33]:

```
LogisticRegression(C=1000, class_weight='balanced', dual=False,
                   fit_intercept=True, intercept_scaling=1, l1_ratio=None,
                   max_iter=500, multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
model.fit(train_vector, y_train)
```

Out[33]:

```
LogisticRegression(C=1.0, class_weight='balanced', dual=False,
                   fit_intercept=True, intercept_scaling=1, l1_ratio=None,
                   max_iter=100, multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

In [34]:

```
y_pred=model.predict(test_vector)
```

In [35]:

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
```

In [36]:

```
confusion_matrix(y_test,y_pred)
```

Out[36]:

```
array([[ 272,    0],
       [1228,    0]], dtype=int64)
```

In [37]:

```
roc_auc_score(y_test,y_pred)
```

Out[37]:

```
0.5
```

In [38]:

```
fpr,tpr,threshold=roc_curve(y_test,y_pred)
```

In [39]:

```
auc(fpr,tpr)
```

Out[39]:

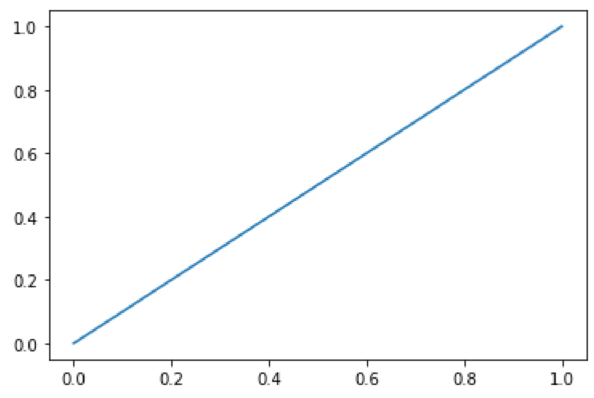
```
0.5
```

In [40]:

```
plt.plot(fpr, tpr, label="train AUC ="+str(auc(fpr,tpr)))
```

Out[40]:

```
[<matplotlib.lines.Line2D at 0x1fed9d85808>]
```


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0