GBDT_and_Random_Forests_bow_tfidf[M]

June 2, 2018

In [19]:
```
!pip install gensim
!pip install imblearn
!pip install xgboost
```

```
Requirement already satisfied: gensim in /usr/local/lib/python3.6/dist-packages (3.4.0)
Requirement already satisfied: smart-open>=1.2.1 in /usr/local/lib/python3.6/dist-packages (from gensim) (1.5.7)
Requirement already satisfied: scipy>=0.18.1 in /usr/local/lib/python3.6/dist-packages (from gensim) (0.19.1)
Requirement already satisfied: numpy>=1.11.3 in /usr/local/lib/python3.6/dist-packages (from gensim) (1.14.3)
Requirement already satisfied: six>=1.5.0 in /usr/local/lib/python3.6/dist-packages (from gensim) (1.11.0)
Requirement already satisfied: bz2file in /usr/local/lib/python3.6/dist-packages (from smart-open>=1.2.1->gensim
Requirement already satisfied: requests in /usr/local/lib/python3.6/dist-packages (from smart-open>=1.2.1->gens
Requirement already satisfied: boto>=2.32 in /usr/local/lib/python3.6/dist-packages (from smart-open>=1.2.1->g
Requirement already satisfied: boto3 in /usr/local/lib/python3.6/dist-packages (from smart-open>=1.2.1->gensim
Requirement already satisfied: idna<2.7,>=2.5 in /usr/local/lib/python3.6/dist-packages (from requests->smart-op
Requirement already satisfied: urllib3<1.23,>=1.21.1 in /usr/local/lib/python3.6/dist-packages (from requests->sm
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.6/dist-packages (from requests->smar
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /usr/local/lib/python3.6/dist-packages (from requests->s
Requirement already satisfied: botocore<1.11.0,>=1.10.31 in /usr/local/lib/python3.6/dist-packages (from boto3-
Requirement already satisfied: s3transfer<0.2.0,>=0.1.10 in /usr/local/lib/python3.6/dist-packages (from boto3->
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /usr/local/lib/python3.6/dist-packages (from boto3->sm
Requirement already satisfied: python-dateutil<3.0.0,>=2.1; python_version >= "2.7" in /usr/local/lib/python3.6/di
Requirement already satisfied: docutils>=0.10 in /usr/local/lib/python3.6/dist-packages (from botocore<1.11.0,>=
Requirement already satisfied: imblearn in /usr/local/lib/python3.6/dist-packages (0.0)
Requirement already satisfied: imbalanced-learn in /usr/local/lib/python3.6/dist-packages (from imblearn) (0.3.3)
Requirement already satisfied: numpy in /usr/local/lib/python3.6/dist-packages (from imbalanced-learn->imblearn
Requirement already satisfied: scipy in /usr/local/lib/python3.6/dist-packages (from imbalanced-learn->imblearn)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.6/dist-packages (from imbalanced-learn->imb
Requirement already satisfied: xgboost in /usr/local/lib/python3.6/dist-packages (0.7.post4)
Requirement already satisfied: numpy in /usr/local/lib/python3.6/dist-packages (from xgboost) (1.14.3)
Requirement already satisfied: scipy in /usr/local/lib/python3.6/dist-packages (from xgboost) (0.19.1)
```

In [0]:
```
from sklearn.model_selection import train_test_split

from sklearn.grid_search import GridSearchCV
from sklearn.grid_search import RandomizedSearchCV
from scipy.stats import randint
```

```
from imblearn.over_sampling import SMOTE

import sqlite3

import pandas as pd
import numpy as np

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import CountVectorizer
import gensim

from sklearn.metrics import classification_report, accuracy_score, confusion_matrix

from sklearn.ensemble import RandomForestClassifier
from xgboost.sklearn import XGBClassifier
```

In [21]:
```
!pip install PyDrive
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials

# 1. Authenticate and create the PyDrive client.
auth.authenticate_user()
gauth = GoogleAuth()
gauth.credentials = GoogleCredentials.get_application_default()
drive = GoogleDrive(gauth)

file_list = drive.ListFile({'q': "'1pbLvjcsi6UtFm3sPciCJGbCG4NK3uyuS' in parents and trashed=false"}
for file1 in file_list:
  print('title: %s, id: %s' % (file1['title'], file1['id']))

sql = drive.CreateFile({'id': '1OzLc3k6-T55I-XRMq47ERyCbQbVw4caF'})
sql.GetContentFile('final.sqlite')
```

```
Requirement already satisfied: PyDrive in /usr/local/lib/python3.6/dist-packages (1.3.1)
Requirement already satisfied: google-api-python-client>=1.2 in /usr/local/lib/python3.6/dist-packages (from PyD
Requirement already satisfied: PyYAML>=3.0 in /usr/local/lib/python3.6/dist-packages (from PyDrive) (3.12)
Requirement already satisfied: oauth2client>=4.0.0 in /usr/local/lib/python3.6/dist-packages (from PyDrive) (4.1.
Requirement already satisfied: httplib2<1dev,>=0.9.2 in /usr/local/lib/python3.6/dist-packages (from google-api-p
Requirement already satisfied: uritemplate<4dev,>=3.0.0 in /usr/local/lib/python3.6/dist-packages (from google-a
Requirement already satisfied: six<2dev,>=1.6.1 in /usr/local/lib/python3.6/dist-packages (from google-api-pytho
Requirement already satisfied: rsa>=3.1.4 in /usr/local/lib/python3.6/dist-packages (from oauth2client>=4.0.0->P
Requirement already satisfied: pyasn1-modules>=0.0.5 in /usr/local/lib/python3.6/dist-packages (from oauth2clie
Requirement already satisfied: pyasn1>=0.1.7 in /usr/local/lib/python3.6/dist-packages (from oauth2client>=4.0.
title: GBDT and Random Forests [M].ipynb, id: 1NauX7hmr_HdwByih8sRQlekzLugLSkll
title: Apply SVM to Amazon reviews data set avg_w2vec [M].ipynb, id: 1ElWunFgWZPb1Iq6w4ZMmqoBSVuPt0C
title: Apply Logistic regression to Amazon reviews data set. [M].ipynb, id: 1Es1wP2edJ0vrKasA5wYJEO-zeZvrqe
```

title: Apply Naive Bayes to Amazon reviews [M].ipynb, id: 1qPxAZeYQUM-eqaKnOSM5ubK2IPIVmdyo
title: clean_final.sqlite, id: 1T0HyUqaVFyD8HfIQEM6WN8jF8SpEOsAo
title: KNN on Credit Card fraud detection.ipynb, id: 1CkA-RBfXqvubKkQrpnjbYUKVsC7VHlTl
title: creditcard.csv, id: 1VpeqlS0lPVrlzlMIqvQTzc3Pno_Cj4SV
title: creditcard.csv, id: 1bnZktEq3N_5wjoCH85oIXHxNwXUW_jx-
title: Untitled, id: 1K0wwkizWx3WO8d-zw-YewWIUrPdINYmp
title: final.sqlite, id: 1OzLc3k6-T55I-XRMq47ERyCbQbVw4caF
title: HeavyComputations.ipynb, id: 1aBORe3gqeFY-iNhzMtr-TIkzEyEvFxcG
title: LogisticRegression.ipynb, id: 1WcVTklMZBMu9VTCIWeupOK0r2aYbHk8p

In [0]:
```
con = sqlite3.connect('final.sqlite') # this is cleaned dataset
final = pd.read_sql_query("""
SELECT Score, Text_not_included
FROM reviews
""", con)[:2000]

for i, seq in enumerate(final['Text_not_included']):
    final['Text_not_included'][i]=final['Text_not_included'][i].decode('UTF-8')
X_train, X_test, y_train , y_test = train_test_split(final['Text_not_included'], final['Score'], test_size=0.2,
```

In [0]:
```
# Generate count BoW
count_vect = CountVectorizer(ngram_range=(1,2) )
count_vect.fit(X_train)
bow_train=count_vect.transform(X_train)
bow_test=count_vect.transform(X_test)

# Generate tf idf
tf_idf_vect=TfidfVectorizer(ngram_range=(1,2), min_df=10, dtype=float)
tf_idf_vect.fit(X_train)
tf_idf_train=tf_idf_vect.transform(X_train)
tf_idf_test=tf_idf_vect.transform(X_test)

# Generate average word2vec
sentences=[]
for review in X_train:
    sentence=[]
    for word in review.split():
        sentence.append(word)
    sentences.append(sentence)

w2vec_model=gensim.models.word2vec.Word2Vec(sentences, min_count=10)

avg_w2vec_train=np.zeros(shape=(len(X_train), 100), dtype=float)

for i, sentence in enumerate(sentences):
    for word in sentence:
        try:
```

```
            avg_w2vec_train[i]+=w2vec_model.wv[word]

        except KeyError:
            pass

    avg_w2vec_train[i]/=len(sentence)

sentences=[]
for review in X_test:
    sentence=[]
    for word in review.split():
        sentence.append(word)
    sentences.append(sentence)

avg_w2vec_test=np.zeros(shape=(len(X_test), 100), dtype=float)

for i, sentence in enumerate(sentences):
    for word in sentence:
        try:
            avg_w2vec_test[i]+=w2vec_model.wv[word]

        except KeyError:
            pass

    avg_w2vec_test[i]/=len(sentence)

# Generate tf idf weighted word2vec
sentences=[]
for review in X_train:
    sentence=[]
    for word in review.split():
        sentence.append(word)
    sentences.append(sentence)

tf_idf_w2vec_train=np.zeros((len(X_train), 100), dtype=float)
feat=tf_idf_vect.get_feature_names()
for i, sentence in enumerate(sentences):
    tf_idf_sum=0
    for word in sentence:
        try:
            tf_idf_w2vec_train[i]+=w2vec_model.wv[word]*tf_idf_train[i, feat.index(word)]
            tf_idf_sum+=tf_idf_train[i, feat.index(word)]
        except KeyError:
            pass
        except ValueError:
            pass
    tf_idf_w2vec_train[i]/=tf_idf_sum
```

```
sentences=[]
for review in X_test:
    sentence=[]
    for word in review.split():
        sentence.append(word)
    sentences.append(sentence)

tf_idf_w2vec_test=np.zeros((len(X_test), 100), dtype=float)

for i, sentence in enumerate(sentences):
    tf_idf_sum=0
    for word in sentence:
        try:
            tf_idf_w2vec_test[i]+=w2vec_model.wv[word]*tf_idf_test[i, feat.index(word)]
            tf_idf_sum+=tf_idf_test[i, feat.index(word)]
        except KeyError:
            pass
        except ValueError:
            pass
    tf_idf_w2vec_test[i]/=tf_idf_sum
```

In [0]:
```
# Upsampling minority class
over_sampler = SMOTE(ratio='minority')
bow_train_resampled, y_train_resampled = over_sampler.fit_sample(bow_train, y_train)
tf_idf_train_resampled, y_train_resampled = over_sampler.fit_sample(tf_idf_train, y_train)
avg_w2vec_train_resampled, y_train_resampled = over_sampler.fit_sample(avg_w2vec_train, y_train)
tf_idf_w2vec_train_resampled, y_train_resampled = over_sampler.fit_sample(tf_idf_w2vec_train, y_tra
```

1 Classification using RandomForest

In [0]:
```
tuned_parameters = {'n_estimators': np.arange(1,100,1)}
gscv = GridSearchCV(RandomForestClassifier(n_jobs=-1), tuned_parameters, scoring = 'accuracy', cv

tuned_parameters = {'n_estimators' : randint(low=1, high=101)}
rscv = RandomizedSearchCV(RandomForestClassifier(n_jobs=-1), tuned_parameters, scoring = 'accu
```

1.0.1 Bow

In [16]:
```
gscv.fit(bow_train_resampled, y_train_resampled)

  predictions=gscv.best_estimator_.predict(bow_test)
  print(classification_report(y_test, predictions))
  print(confusion_matrix(y_test, predictions).T)
  tn, fp, fn, tp = confusion_matrix(y_test, predictions).ravel()

  print("TPR = {}\n TNR = {}\n FPR = {}\n FNR = {}".format(tp/(fn+tp), tn/(tn+fp), fp/(tn+fp), fn/(fn+tp)))
```

	precision	recall	f1-score	support
negative	0.00	0.00	0.00	87
positive	0.78	1.00	0.88	313
avg / total	0.61	0.78	0.69	400

```
[[  0   0]
 [ 87 313]]
TPR = 1.0
 TNR = 0.0
 FPR = 1.0
 FNR = 0.0
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/metrics/classification.py:1135: UndefinedMetricWarning: Precision
  'precision', 'predicted', average, warn_for)
```

```
In [17]: gscv.best_estimator_
```

```
Out[17]: RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                                max_depth=None, max_features='auto', max_leaf_nodes=None,
                                min_impurity_decrease=0.0, min_impurity_split=None,
                                min_samples_leaf=1, min_samples_split=2,
                                min_weight_fraction_leaf=0.0, n_estimators=20, n_jobs=-1,
                                oob_score=False, random_state=None, verbose=0,
                                warm_start=False)
```

```
In [14]: rscv.fit(bow_train_resampled, y_train_resampled)
```

```
predictions=rscv.best_estimator_.predict(bow_test)
print(classification_report(y_test, predictions))
print(confusion_matrix(y_test, predictions).T)
tn, fp, fn, tp = confusion_matrix(y_test, predictions).ravel()

print("TPR = {}\n TNR = {}\n FPR = {}\n FNR = {}".format(tp/(fn+tp), tn/(tn+fp), fp/(tn+fp), fn/(fn+tp)))
```

	precision	recall	f1-score	support
negative	0.33	0.01	0.02	87
positive	0.78	0.99	0.88	313
avg / total	0.69	0.78	0.69	400

```
[[  1   2]
 [ 86 311]]
TPR = 0.9936102236421726
 TNR = 0.011494252873563218
```

```
 FPR = 0.9885057471264368
 FNR = 0.006389776357827476
```

In [15]:
```
rscv.best_estimator_
```

Out[15]:
```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                       max_depth=None, max_features='auto', max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=24, n_jobs=-1,
                       oob_score=False, random_state=None, verbose=0,
                       warm_start=False)
```

1.0.2 TF IDF

In [15]:
```
gscv.fit(tf_idf_train_resampled, y_train_resampled)

        predictions=gscv.best_estimator_.predict(tf_idf_test)
        print(classification_report(y_test, predictions))
        print(confusion_matrix(y_test, predictions).T)
        tn, fp, fn, tp = confusion_matrix(y_test, predictions).ravel()

        print("TPR = {}\n TNR = {}\n FPR = {}\n FNR = {}".format(tp/(fn+tp), tn/(tn+fp), fp/(tn+fp), fn/(fn+tp)))
```

```
                precision    recall  f1-score   support

    negative       0.42      0.11      0.18        87
    positive       0.80      0.96      0.87       313

avg / total       0.71      0.77      0.72       400

[[ 10  14]
 [ 77 299]]
TPR = 0.9552715654952076
 TNR = 0.11494252873563218
 FPR = 0.8850574712643678
 FNR = 0.04472843450479233
```

In [16]:
```
rscv.fit(tf_idf_train_resampled, y_train_resampled)

        predictions=rscv.best_estimator_.predict(tf_idf_test)
        print(classification_report(y_test, predictions))
        print(confusion_matrix(y_test, predictions).T)
        tn, fp, fn, tp = confusion_matrix(y_test, predictions).ravel()

        print("TPR = {}\n TNR = {}\n FPR = {}\n FNR = {}".format(tp/(fn+tp), tn/(tn+fp), fp/(tn+fp), fn/(fn+tp)))
```

	precision	recall	f1-score	support
negative	0.54	0.15	0.23	87
positive	0.80	0.96	0.88	313
avg / total	0.75	0.79	0.74	400

```
[[ 13  11]
 [ 74 302]]
TPR = 0.9648562300319489
 TNR = 0.14942528735632185
 FPR = 0.8505747126436781
 FNR = 0.03514376996805112
```

```
In [17]: gscv.best_estimator_
```

```
Out[17]: RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                                max_depth=None, max_features='auto', max_leaf_nodes=None,
                                min_impurity_decrease=0.0, min_impurity_split=None,
                                min_samples_leaf=1, min_samples_split=2,
                                min_weight_fraction_leaf=0.0, n_estimators=95, n_jobs=-1,
                                oob_score=False, random_state=None, verbose=0,
                                warm_start=False)
```

```
In [18]: rscv.best_estimator_
```

```
Out[18]: RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                                max_depth=None, max_features='auto', max_leaf_nodes=None,
                                min_impurity_decrease=0.0, min_impurity_split=None,
                                min_samples_leaf=1, min_samples_split=2,
                                min_weight_fraction_leaf=0.0, n_estimators=73, n_jobs=-1,
                                oob_score=False, random_state=None, verbose=0,
                                warm_start=False)
```

1.1 Classification using GBDT

1.1.1 BoW

```
In [0]: from scipy.stats import uniform

        tuned_parameters = {'n_estimators': np.arange(10,60,10),'max_depth' : np.arange(1,5,1), 'learning_rat
        gscv = GridSearchCV(XGBClassifier(), tuned_parameters, scoring = 'accuracy', cv=5)

        tuned_parameters = {'n_estimators': randint(low=10, high=61), 'max_depth' : randint(low=1, high=6), 'le
        rscv = RandomizedSearchCV(XGBClassifier(), tuned_parameters, scoring = 'accuracy', cv=5, n_iter=2
```

```
In [38]: gscv.fit(bow_train_resampled, y_train_resampled)
```

```python
predictions=gscv.best_estimator_.predict(bow_test)
print(classification_report(y_test, predictions))
print(confusion_matrix(y_test, predictions).T)
tn, fp, fn, tp = confusion_matrix(y_test, predictions).ravel()

print("TPR = {}\n TNR = {}\n FPR = {}\n FNR = {}".format(tp/(fn+tp), tn/(tn+fp), fp/(tn+fp), fn/(fn+tp)))
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
```

```
              precision    recall  f1-score   support

   negative       0.77      0.11      0.20        87
   positive       0.80      0.99      0.89       313

avg / total       0.79      0.80      0.74       400

[[ 10   3]
 [ 77 310]]
TPR = 0.9904153354632588
 TNR = 0.11494252873563218
 FPR = 0.8850574712643678
 FNR = 0.009584664536741214


/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
  if diff:
```

In [41]:
```
rscv.fit(bow_train_resampled, y_train_resampled)

predictions=rscv.best_estimator_.predict(bow_test)
print(classification_report(y_test, predictions))
print(confusion_matrix(y_test, predictions).T)
tn, fp, fn, tp = confusion_matrix(y_test, predictions).ravel()

print("TPR = {}\n TNR = {}\n FPR = {}\n FNR = {}".format(tp/(fn+tp), tn/(tn+fp), fp/(tn+fp), fn/(fn+tp)))
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
  if diff:
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:

             precision    recall  f1-score   support

    negative       0.55      0.07      0.12        87
    positive       0.79      0.98      0.88       313

avg / total       0.74      0.79      0.71       400

[[  6   5]
```

```
 [ 81 308]]
TPR = 0.9840255591054313
 TNR = 0.06896551724137931
 FPR = 0.9310344827586207
 FNR = 0.01597444089456869


/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:


In [42]: gscv.best_estimator_

Out[42]: XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
                       colsample_bytree=1, gamma=0, learning_rate=0.275, max_delta_step=0,
                       max_depth=2, min_child_weight=1, missing=None, n_estimators=50,
                       n_jobs=1, nthread=None, objective='binary:logistic', random_state=0,
                       reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
                       silent=True, subsample=1)

In [43]: rscv.best_estimator_

Out[43]: XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
                       colsample_bytree=1, gamma=0, learning_rate=0.20249593153909556,
                       max_delta_step=0, max_depth=4, min_child_weight=1, missing=None,
                       n_estimators=31, n_jobs=1, nthread=None,
                       objective='binary:logistic', random_state=0, reg_alpha=0,
                       reg_lambda=1, scale_pos_weight=1, seed=None, silent=True,
                       subsample=1)
```

1.1.2 TF IDF

```
In [44]: gscv.fit(tf_idf_train_resampled, y_train_resampled)

             predictions=gscv.best_estimator_.predict(tf_idf_test)
             print(classification_report(y_test, predictions))
             print(confusion_matrix(y_test, predictions).T)
             tn, fp, fn, tp = confusion_matrix(y_test, predictions).ravel()

             print("TPR = {}\n TNR = {}\n FPR = {}\n FNR = {}".format(tp/(fn+tp), tn/(tn+fp), fp/(tn+fp), fn/(fn+tp)))

/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value
  if diff:
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
  if diff:

             precision    recall  f1-score   support

   negative       0.57      0.32      0.41        87
   positive       0.83      0.93      0.88       313

avg / total       0.78      0.80      0.78       400

[[ 28  21]
 [ 59 292]]
TPR = 0.9329073482428115
 TNR = 0.3218390804597701
 FPR = 0.6781609195402298
 FNR = 0.0670926517571885
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value 
  if diff:
```

In [45]:
```python
rscv.fit(tf_idf_train_resampled, y_train_resampled)

        predictions=rscv.best_estimator_.predict(tf_idf_test)
        print(classification_report(y_test, predictions))
        print(confusion_matrix(y_test, predictions).T)
        tn, fp, fn, tp = confusion_matrix(y_test, predictions).ravel()

        print("TPR = {}\n TNR = {}\n FPR = {}\n FNR = {}".format(tp/(fn+tp), tn/(tn+fp), fp/(tn+fp), fn/(fn+tp)))
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value 
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value 
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value 
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value 
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value 
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value 
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value 
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value 
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value 
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value 
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value 
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value 
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value 
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value 
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value 
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value 
  if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value 
```

```
    if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
    if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
    if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
    if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
    if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
    if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
    if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
    if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
    if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
    if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
    if diff:
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
    if diff:
```

```
             precision    recall  f1-score   support

   negative       0.60      0.32      0.42        87
   positive       0.83      0.94      0.88       313

avg / total       0.78      0.81      0.78       400

[[ 28  19]
 [ 59 294]]
TPR = 0.939297124600639
 TNR = 0.3218390804597701
 FPR = 0.6781609195402298
 FNR = 0.06070287539936102
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The truth value o
    if diff:
```

In [46]:
```
gscv.best_estimator_
```

Out[46]:
```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bytree=1, gamma=0, learning_rate=0.5, max_delta_step=0,
              max_depth=4, min_child_weight=1, missing=None, n_estimators=50,
```

```
                n_jobs=1, nthread=None, objective='binary:logistic', random_state=0,
                reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
                silent=True, subsample=1)
```

In [47]:
```
rscv.best_estimator_
```

Out[47]:
```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
       colsample_bytree=1, gamma=0, learning_rate=0.5238871512238155,
       max_delta_step=0, max_depth=5, min_child_weight=1, missing=None,
       n_estimators=47, n_jobs=1, nthread=None,
       objective='binary:logistic', random_state=0, reg_alpha=0,
       reg_lambda=1, scale_pos_weight=1, seed=None, silent=True,
       subsample=1)
```

2 Conclusions

Random forest performance
Using BoW TPR = 0.99 TNR = 0.01
best n_estimators = 93
Using TF IDF TPR = 0.96 TNR = 0.14
best n_estimators = 72
GBDT performance
Using BoW TPR = 0.99 TNR = 0.11
best n_estimators = 50 best max_depth = 2 best eta = 0.275
Using TF IDF TPR = 0.93 TNR = 0.32
best n_estimators = 50 best max_depth = 4 best eta = 0.5
TF IDF representations provide better results with both classifiers. GBDT provides better TNR. GBDT combined with TF IDF provides optimal value for both TPR and TNR