urbnqyj9a

May 17, 2023

[1]:

```
pip install sklearn
```

```
Requirement already satisfied: sklearn in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (0.0)
Requirement already satisfied: scikit-learn in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from sklearn)
(1.0.2)
Requirement already satisfied: joblib>=0.11 in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from scikit-
learn->sklearn) (1.1.0)
Requirement already satisfied: scipy>=1.1.0 in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from scikit-
learn->sklearn) (1.7.3)
Requirement already satisfied: numpy>=1.14.6 in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from scikit-
learn->sklearn) (1.21.5)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from scikit-
learn->sklearn) (3.1.0)
Note: you may need to restart the kernel to use updated packages.
```

[2]:

```
pip install numpy
```

```
Requirement already satisfied: numpy in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (1.21.5)
Note: you may need to restart the kernel to use updated packages.
```

[3]:

```
pip install pandas
```

```
Requirement already satisfied: pandas in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (1.3.5)
Requirement already satisfied: pytz>=2017.3 in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from pandas)
(2022.1)
Requirement already satisfied: numpy>=1.17.3 in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from pandas)
(1.21.5)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\sd
```

```
pro\appdata\local\programs\python\python37\lib\site-packages (from pandas)
(2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from python-
dateutil>=2.7.3->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

[4]:
```
pip install nltk
```

```
Requirement already satisfied: nltk in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (3.7)
Requirement already satisfied: tqdm in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from nltk)
(4.64.0)
Requirement already satisfied: click in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from nltk) (8.1.2)
Requirement already satisfied: joblib in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from nltk) (1.1.0)
Requirement already satisfied: regex>=2021.8.3 in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from nltk)
(2022.3.15)
Requirement already satisfied: importlib-metadata in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from click->nltk)
(4.11.3)
Requirement already satisfied: colorama in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from click->nltk)
(0.4.4)
Requirement already satisfied: typing-extensions>=3.6.4 in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from importlib-
metadata->click->nltk) (4.1.1)
Requirement already satisfied: zipp>=0.5 in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from importlib-
metadata->click->nltk) (3.8.0)
Note: you may need to restart the kernel to use updated packages.
```

[5]:
```
pip install matplotlib
```

```
Requirement already satisfied: matplotlib in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (3.5.1)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from matplotlib)
(3.0.8)
Requirement already satisfied: pillow>=6.2.0 in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from matplotlib)
(9.1.0)
Requirement already satisfied: numpy>=1.17 in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from matplotlib)
(1.21.5)
```

```
Requirement already satisfied: packaging>=20.0 in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from matplotlib)
(21.3)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from matplotlib)
(4.32.0)
Requirement already satisfied: cycler>=0.10 in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from matplotlib)
(0.11.0)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from matplotlib)
(2.8.2)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from matplotlib)
(1.4.2)
Requirement already satisfied: typing-extensions in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from
kiwisolver>=1.0.1->matplotlib) (4.1.1)
Requirement already satisfied: six>=1.5 in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from python-
dateutil>=2.7->matplotlib) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

[6]:
```
pip install imblearn
```

```
Collecting imblearn
  Downloading imblearn-0.0-py2.py3-none-any.whl (1.9 kB)
Collecting imbalanced-learn
  Downloading imbalanced_learn-0.9.0-py3-none-any.whl (199 kB)
     -------------------------------------- 199.1/199.1 KB 3.0 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.14.6 in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from imbalanced-
learn->imblearn) (1.21.5)
Requirement already satisfied: scipy>=1.1.0 in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from imbalanced-
learn->imblearn) (1.7.3)
Requirement already satisfied: joblib>=0.11 in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from imbalanced-
learn->imblearn) (1.1.0)
Requirement already satisfied: scikit-learn>=1.0.1 in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from imbalanced-
learn->imblearn) (1.0.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\sd
pro\appdata\local\programs\python\python37\lib\site-packages (from imbalanced-
learn->imblearn) (3.1.0)
Installing collected packages: imbalanced-learn, imblearn
Successfully installed imbalanced-learn-0.9.0 imblearn-0.0
Note: you may need to restart the kernel to use updated packages.
```

1 Importing

[7]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.feature_extraction.text import TfidfTransformer, CountVectorizer,␣
 ↪TfidfVectorizer
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split

from nltk.stem.porter import PorterStemmer
import nltk
import re, string
from nltk.corpus import stopwords

from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import LinearSVC
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier

from sklearn.model_selection import cross_val_score

from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import plot_precision_recall_curve
import matplotlib.pyplot as plt
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
from sklearn.metrics import classification_report
from sklearn import metrics
```

2 Loading Data

[13]:

```
df = pd.read_json('./Dataset.json')
df.head
```

[13]:

```
<bound method NDFrame.head of
content  \
0                                  Get fucking real dude.
1       She is as dirty as they come  and that crook …
2       why did you fuck it up. I could do it all day…
```

```
3       Dude they dont finish enclosing the fucking s…
4       WTF are you talking about Men? No men thats n…
…                                                  …
19996    I dont. But what is complaining about it goi…
19997   Bahah  yeah i&;m totally just gonna&; get pis…
19998       hahahahaha >:) im evil mwahahahahahahahahaha
19999            What&;s something unique about Ohio? :)
20000              Who is the biggest gossiper you know?

                           annotation  extras
0      {'notes': '', 'label': ['1']}     NaN
1      {'notes': '', 'label': ['1']}     NaN
2      {'notes': '', 'label': ['1']}     NaN
3      {'notes': '', 'label': ['1']}     NaN
4      {'notes': '', 'label': ['1']}     NaN
…                                …     …
19996  {'notes': '', 'label': ['0']}     NaN
19997  {'notes': '', 'label': ['0']}     NaN
19998  {'notes': '', 'label': ['0']}     NaN
19999  {'notes': '', 'label': ['0']}     NaN
20000  {'notes': '', 'label': ['0']}     NaN

[20001 rows x 3 columns]>
```

[14]:

```python
for i in range(0,len(df)):
    if df.annotation[i]['label'][0] == '1':
        df.annotation[i] = 1
    else:
        df.annotation[i] = 0
```

```
C:\Users\sd pro\AppData\Local\Programs\Python\Python37\lib\site-
packages\ipykernel_launcher.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  This is separate from the ipykernel package so we can avoid doing imports
until
C:\Users\sd pro\AppData\Local\Programs\Python\Python37\lib\site-
packages\ipykernel_launcher.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  """
```

```
[15]: df.drop(['extras'],axis = 1,inplace = True)
      df
```

```
[15]:
                                                 content annotation
0                                 Get fucking real dude.          1
1       She is as dirty as they come  and that crook …          1
2       why did you fuck it up. I could do it all day…          1
3       Dude they dont finish enclosing the fucking s…          1
4       WTF are you talking about Men? No men thats n…          1
…                                                  …        …
19996    I dont. But what is complaining about it goi…          0
19997   Bahah  yeah i&;m totally just gonna&; get pis…          0
19998       hahahahaha >:) im evil mwahahahahahahahahaha          0
19999            What&;s something unique about Ohio? :)          0
20000              Who is the biggest gossiper you know?          0

[20001 rows x 2 columns]
```

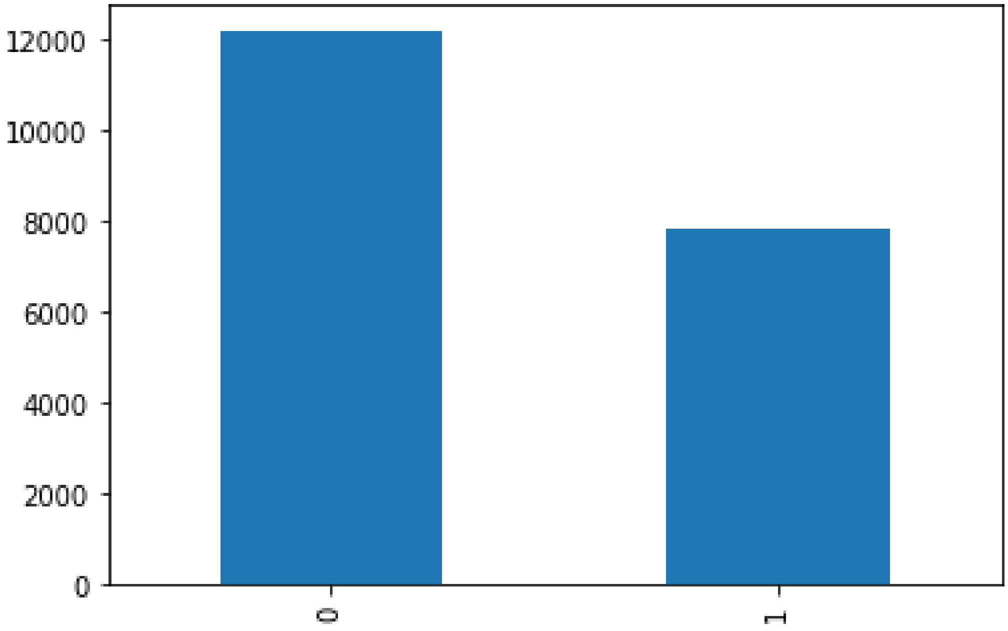
```
[16]: df.shape
```

```
[16]: (20001, 2)
```

3 Visualization

```
[17]: df['annotation'].value_counts().sort_index().plot.bar()
```

```
[17]: <AxesSubplot:>
```



[18]:

```
#Biasness
print("PosiNon cyber trollingtive: ", df.annotation.value_counts()[0]/len(df.
 ↪annotation)*100,"%")
print("Cybertrolling: ", df.annotation.value_counts()[1]/len(df.
 ↪annotation)*100,"%")
```

```
PosiNon cyber trollingtive:  60.89195540222989 %
Cybertrolling:  39.10804459777012 %
```

4 Preprocessing

[19]:

```
nltk.download('stopwords')
stop = stopwords.words('english')

regex = re.compile('[%s]' % re.escape(string.punctuation))

def test_re(s):
    return regex.sub('', s)

df ['content_without_stopwords'] = df['content'].apply(lambda x: ' '.join([word␣
 ↪for word in x.split() if word not in (stop)]))
df ['content_without_puncs'] = df['content_without_stopwords'].apply(lambda x:␣
 ↪regex.sub('',x))
del df['content_without_stopwords']
del df['content']
df
```

```
[nltk_data] Downloading package stopwords to C:\Users\sd
[nltk_data]     pro\AppData\Roaming\nltk_data…
[nltk_data]   Unzipping corpora\stopwords.zip.
```

[19]:

```
      annotation                               content_without_puncs
0              1                               Get fucking real dude
1              1  She dirty come crook Rengel Dems fucking corru…
2              1  fuck up I could day too Lets hour Ping later s…
3              1  Dude dont finish enclosing fucking showers I h…
4              1      WTF talking Men No men thats menage thats gay
…            …                                                 …
19996          0                     I dont But complaining going do
19997          0  Bahah yeah im totally gonna get pissed talking…
19998          0          hahahahaha  im evil mwahahahahahahahahaha
19999          0                          Whats something unique Ohio
20000          0                             Who biggest gossiper know
```

```
[20001 rows x 2 columns]
```

[20]:

```
#Stemming
porter_stemmer = PorterStemmer()
#punctuations
nltk.download('punkt')
tok_list = []
size = df.shape[0]

for i in range(size):
  word_data = df['content_without_puncs'][i]
  nltk_tokens = nltk.word_tokenize(word_data)
  final = ''
  for w in nltk_tokens:
    final = final + ' ' + porter_stemmer.stem(w)
  tok_list.append(final)

df['content_tokenize'] = tok_list
del df['content_without_puncs']
df
```

```
[nltk_data] Downloading package punkt to C:\Users\sd
[nltk_data]     pro\AppData\Roaming\nltk_data…
[nltk_data]   Unzipping tokenizers\punkt.zip.
```

[20]:

```
      annotation                                     content_tokenize
0              1                                   get fuck real dude
1              1    she dirti come crook rengel dem fuck corrupt …
2              1    fuck up i could day too let hour ping later s…
3              1    dude dont finish enclos fuck shower i hate ha…
4              1             wtf talk men no men that menag that gay
…            …                                                  …
19996          0                          i dont but complain go do
19997          0    bahah yeah im total gon na get piss talk you …
19998          0           hahahahaha im evil mwahahahahahahahahaha
19999          0                              what someth uniqu ohio
20000          0                             who biggest gossip know

[20001 rows x 2 columns]
```

[21]:

```
noNums = []
for i in range(len(df)):
  noNums.append(''.join([i for i in df['content_tokenize'][i] if not i.
 ↪isdigit()]))

df['content'] = noNums
df
```

[21]:

```
       annotation                                  content_tokenize  \
0               1                                get fuck real dude
1               1   she dirti come crook rengel dem fuck corrupt …
2               1   fuck up i could day too let hour ping later s…
3               1   dude dont finish enclos fuck shower i hate ha…
4               1            wtf talk men no men that menag that gay
…             …                                                 …
19996           0                          i dont but complain go do
19997           0   bahah yeah im total gon na get piss talk you …
19998           0          hahahahaha im evil mwahahahahahahahahaha
19999           0                             what someth uniqu ohio
20000           0                            who biggest gossip know

                                              content
0                                  get fuck real dude
1      she dirti come crook rengel dem fuck corrupt …
2      fuck up i could day too let hour ping later s…
3      dude dont finish enclos fuck shower i hate ha…
4               wtf talk men no men that menag that gay
…                                                 …
19996                       i dont but complain go do
19997  bahah yeah im total gon na get piss talk you …
19998        hahahahaha im evil mwahahahahahahahahaha
19999                          what someth uniqu ohio
20000                         who biggest gossip know

[20001 rows x 3 columns]
```

[22]:

```
tfIdfVectorizer=TfidfVectorizer(use_idf=True, sublinear_tf=True)
tfIdf = tfIdfVectorizer.fit_transform(df.content.tolist())
```

[23]:

```
print(tfIdf)
```

```
  (0, 3598)	0.5682792040556577
  (0, 10534)	0.6408032598619846
  (0, 4665)	0.3314842764826402
  (0, 4896)	0.3956616014132561
  (1, 7497)	0.1421522208901913
  (1, 7670)	0.18997382467613527
  (1, 10707)	0.3380770158779807
  (1, 7868)	0.17712641457020445
  (1, 6881)	0.2707206754001475
  (1, 2649)	0.3478358132370042
  (1, 3127)	0.36956626902789813
  (1, 10686)	0.36956626902789813
  (1, 2791)	0.3609013757539863
  (1, 2453)	0.20014266836955738
  (1, 3306)	0.294004579420996
  (1, 11402)	0.24231137330135857
  (1, 4665)	0.12302268120056382
  (2, 5648)	0.26264752682375
  (2, 1476)	0.2858475342270202
  (2, 14420)	0.28761927584628644
  (2, 11156)	0.4130661580674724
  (2, 7317)	0.3061308801267633
  (2, 9784)	0.38298243181872793
  (2, 5956)	0.28144866948736874
  (2, 7434)	0.24199503289435126
  :	:
  (19997, 8527)	0.362558005670761
  (19997, 14527)	0.1829917686470462
  (19997, 364)	0.2524980709313037
  (19997, 8632)	0.19487099515279527
  (19997, 5039)	0.21529577669215724
  (19997, 14639)	0.15162817445998714
  (19997, 5311)	0.2322934882970198
  (19997, 9798)	0.22212274676003707
  (19997, 13161)	0.22711912398563924
  (19997, 6367)	0.1396437116782225
  (19997, 12782)	0.14437044050700218
  (19997, 12583)	0.21638447818263024
  (19997, 4896)	0.14751463907596812
  (19998, 8599)	0.6474267500657062
  (19998, 5355)	0.5240398795250955
  (19998, 4014)	0.5046761457592059
  (19998, 6367)	0.22698633410034566
  (19999, 13559)	0.6577171835959204
  (19999, 9144)	0.5711145182804813
  (19999, 11870)	0.38585942493978787
  (19999, 14101)	0.30388948253771536
  (20000, 5085)	0.7029240479741253
  (20000, 1246)	0.5142345992116426
  (20000, 14161)	0.4012493121480635
  (20000, 7153)	0.28365392515178917
```

[24]:
```
print(tfIdf.shape) # means total rows  20001 with 14783 features
```

```
(20001, 14783)
```

[25]:
```
df2 = pd.DataFrame(tfIdf[2].T.todense(), index=tfIdfVectorizer.
 ↪get_feature_names(), columns=["TF-IDF"]) #for second entry only(just to␣
 ↪check if working)
df2 = df2.sort_values('TF-IDF', ascending=False)
print (df2.head(10))
```

```
         TF-IDF
sched  0.413066
ping   0.382982
later  0.306131
write  0.287619
book   0.285848
hour   0.281449
here   0.262648
let    0.241995
up     0.237401
could  0.223151
```

```
C:\Users\sd pro\AppData\Local\Programs\Python\Python37\lib\site-
packages\sklearn\utils\deprecation.py:87: FutureWarning: Function
get_feature_names is deprecated; get_feature_names is deprecated in 1.0 and will
be removed in 1.2. Please use get_feature_names_out instead.
  warnings.warn(msg, category=FutureWarning)
```

[26]:
```
dfx = pd.DataFrame(tfIdf.toarray(), columns = tfIdfVectorizer.
 ↪get_feature_names())
print(dfx)
```

```
        aa  aaaaaaaaaa  aaaaaanndgummi  aaaagh  aaaawwwww  aaand  \
0      0.0         0.0             0.0     0.0        0.0    0.0
1      0.0         0.0             0.0     0.0        0.0    0.0
2      0.0         0.0             0.0     0.0        0.0    0.0
3      0.0         0.0             0.0     0.0        0.0    0.0
4      0.0         0.0             0.0     0.0        0.0    0.0
…    …         …             …     …        …    …
19996  0.0         0.0             0.0     0.0        0.0    0.0
19997  0.0         0.0             0.0     0.0        0.0    0.0
19998  0.0         0.0             0.0     0.0        0.0    0.0
19999  0.0         0.0             0.0     0.0        0.0    0.0
20000  0.0         0.0             0.0     0.0        0.0    0.0

       aaanyyywhoooooooo   aaargh  aaarrrg  aah  …  zon  zone  zoo  zoom   \
0                    0.0      0.0      0.0  0.0  …  0.0   0.0  0.0   0.0
1                    0.0      0.0      0.0  0.0  …  0.0   0.0  0.0   0.0
2                    0.0      0.0      0.0  0.0  …  0.0   0.0  0.0   0.0
3                    0.0      0.0      0.0  0.0  …  0.0   0.0  0.0   0.0
4                    0.0      0.0      0.0  0.0  …  0.0   0.0  0.0   0.0
…                   …     …      …  …  …  …    …  …   …
19996                0.0      0.0      0.0  0.0  …  0.0   0.0  0.0   0.0
19997                0.0      0.0      0.0  0.0  …  0.0   0.0  0.0   0.0
19998                0.0      0.0      0.0  0.0  …  0.0   0.0  0.0   0.0
19999                0.0      0.0      0.0  0.0  …  0.0   0.0  0.0   0.0
20000                0.0      0.0      0.0  0.0  …  0.0   0.0  0.0   0.0
```

```
       zro  zucker  zune  zzzz  zzzzzz  zzzzzzzz
0      0.0     0.0   0.0   0.0     0.0       0.0
1      0.0     0.0   0.0   0.0     0.0       0.0
2      0.0     0.0   0.0   0.0     0.0       0.0
3      0.0     0.0   0.0   0.0     0.0       0.0
4      0.0     0.0   0.0   0.0     0.0       0.0
…    …     …    …   …     …       …
19996  0.0     0.0   0.0   0.0     0.0       0.0
19997  0.0     0.0   0.0   0.0     0.0       0.0
19998  0.0     0.0   0.0   0.0     0.0       0.0
19999  0.0     0.0   0.0   0.0     0.0       0.0
20000  0.0     0.0   0.0   0.0     0.0       0.0

[20001 rows x 14783 columns]
```

```
C:\Users\sd pro\AppData\Local\Programs\Python\Python37\lib\site-
packages\sklearn\utils\deprecation.py:87: FutureWarning: Function
get_feature_names is deprecated; get_feature_names is deprecated in 1.0 and will
be removed in 1.2. Please use get_feature_names_out instead.
  warnings.warn(msg, category=FutureWarning)
```

[27]:

```python
def display_scores(vectorizer, tfidf_result):
    scores = zip(vectorizer.get_feature_names(),
                 np.asarray(tfidf_result.sum(axis=0)).ravel())
    sorted_scores = sorted(scores, key=lambda x: x[1], reverse=True)
    i=0
    for item in sorted_scores:
        print ("{0:50} Score: {1}".format(item[0], item[1]))
        i = i+1
        if (i > 25):
          break
```

[28]:

```python
#top 25 words
display_scores(tfIdfVectorizer, tfIdf)
```

```
hate                                               Score: 533.8157298036014
fuck                                               Score: 503.76150769255435
damn                                               Score: 482.3875012051478
suck                                               Score: 407.37790877127185
ass                                                Score: 337.54089621427744
that                                               Score: 311.6250930420745
lol                                                Score: 298.0085779872157
im                                                 Score: 296.0216055277791
like                                               Score: 287.8183474868775
you                                                Score: 284.7850587424088
it                                                 Score: 254.75722294501585
get                                                Score: 253.19747902607998
what                                               Score: 221.43673623523864
```

```
know                                               Score:  211.53595900888456
would                                              Score:  202.5073882820925
bitch                                              Score:  193.08800391463464
ye                                                 Score:  182.22364463196365
love                                               Score:  181.49014270754344
go                                                 Score:  180.2588319545915
haha                                               Score:  179.29466045019018
think                                              Score:  178.9039058038677
one                                                Score:  174.16019276608517
do                                                 Score:  160.57524593088053
time                                               Score:  160.1100301847739
gay                                                Score:  159.5820454915121
peopl                                              Score:  151.04499856119287

C:\Users\sd pro\AppData\Local\Programs\Python\Python37\lib\site-
packages\sklearn\utils\deprecation.py:87: FutureWarning: Function
get_feature_names is deprecated; get_feature_names is deprecated in 1.0 and will
be removed in 1.2. Please use get_feature_names_out instead.
  warnings.warn(msg, category=FutureWarning)
```

[29]:

```
X=tfIdf.toarray()
y = np.array(df.annotation.tolist())
#Spltting
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=0)

print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)
```

```
(16000, 14783)
(16000,)
(4001, 14783)
(4001,)
```

[30]:

```
#Training data biasness
unique_elements, counts_elements = np.unique(y_train, return_counts=True)
print(np.asarray((unique_elements, counts_elements)))
```

```
[[   0    1]
 [9750 6250]]
```

[31]:

```
#Test Data
unique_elements, counts_elements = np.unique(y_test, return_counts=True)
print(np.asarray((unique_elements, counts_elements)))
```

```
[[   0    1]
```

```
 [2429 1572]]
```

[32]:

```
#Random oversampling on training data
from imblearn.over_sampling import RandomOverSampler

oversample = RandomOverSampler(sampling_strategy='not majority')
X_over, y_over = oversample.fit_resample(X_train, y_train)
```

[33]:

```
print(X_over.shape)
print(y_over.shape)
```

```
(19500, 14783)
(19500,)
```

[34]:

```
unique_elements, counts_elements = np.unique(y_over, return_counts=True)
print(np.asarray((unique_elements, counts_elements)))
```

```
[[   0    1]
 [9750 9750]]
```

5 Training and Calculating Scores

[35]:

```
def getStatsFromModel(model):
  print(classification_report(y_test, y_pred))
  disp = plot_precision_recall_curve(model, X_test, y_test)
  disp.ax_.set_title('2-class Precision-Recall curve: '
                 'AP={0:0.2f}')

  logit_roc_auc = roc_auc_score(y_test, model.predict(X_test))
  fpr, tpr, thresholds = roc_curve(y_test, model.predict_proba(X_test)[:,1])
  plt.figure()
  plt.plot(fpr, tpr, label='(area = %0.2f)' % logit_roc_auc)
  plt.plot([0, 1], [0, 1],'r--')
  plt.xlim([0.0, 1.0])
  plt.ylim([0.0, 1.05])
  plt.xlabel('False Positive Rate')
  plt.ylabel('True Positive Rate')
  plt.title('Receiver operating characteristic')
  plt.legend(loc="lower right")
  plt.savefig('Log_ROC')
  # plt.show()
```

5.1 Normal Methods

[36]:

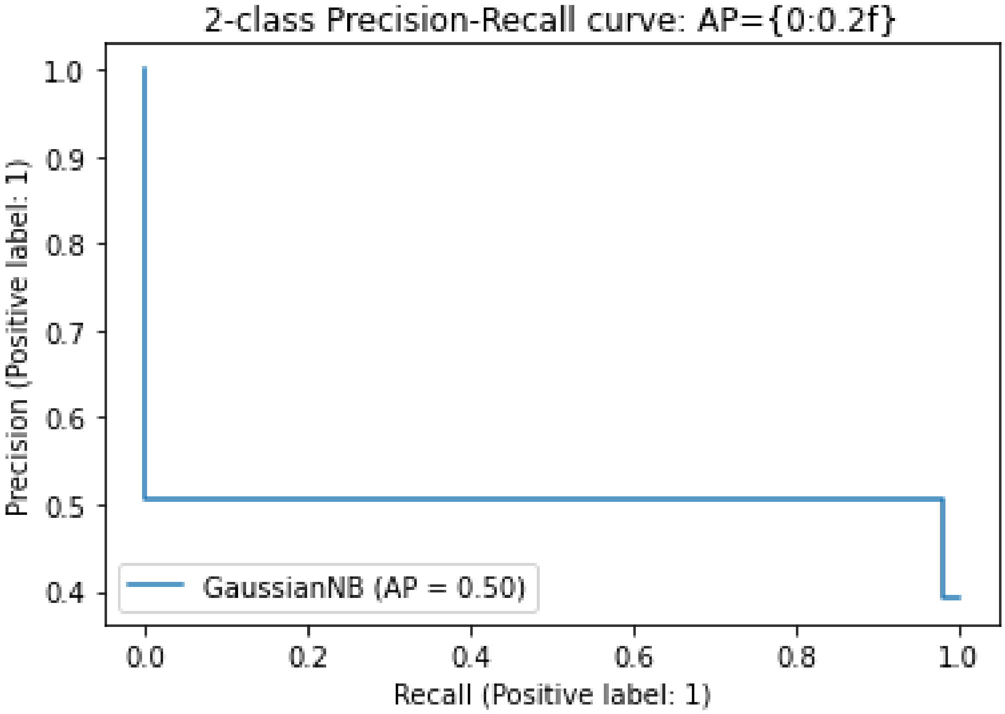
```
#Supervised Methods
# 3 normal methods
# 2 ensemble methods
gnb = GaussianNB()
gnbmodel = gnb.fit(X_over, y_over)
y_pred = gnbmodel.predict(X_test)
print ("Score:", gnbmodel.score(X_test, y_test))
print("Confusion Matrix: \n", confusion_matrix(y_test, y_pred))
getStatsFromModel(gnb)
```

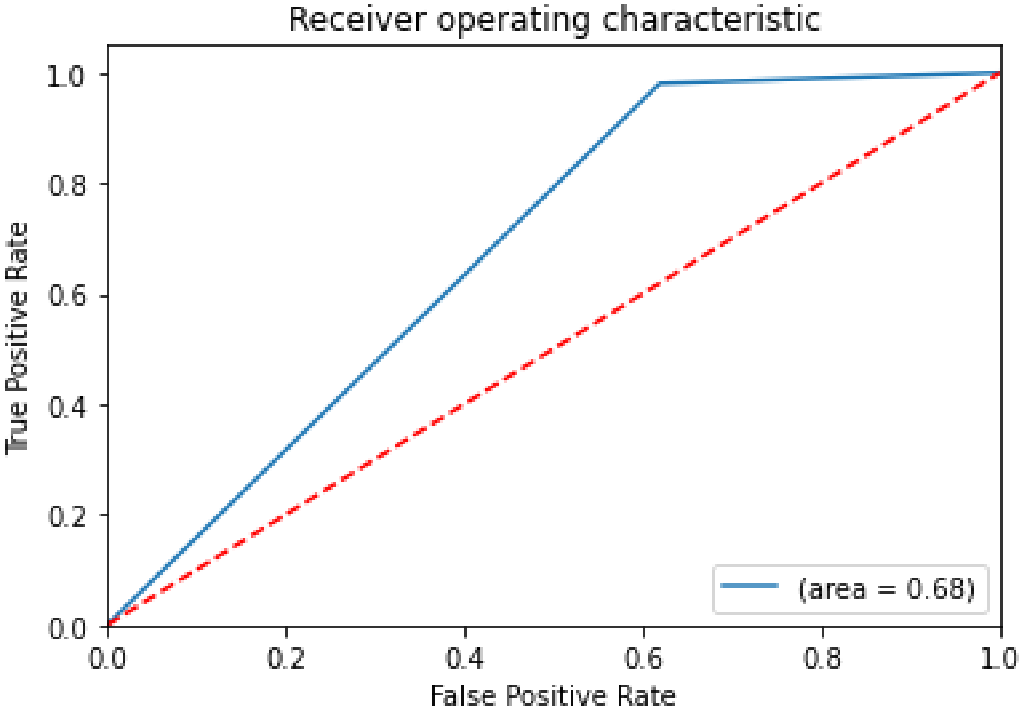
```
Score: 0.6163459135216196
Confusion Matrix:
 [[ 925 1504]
 [  31 1541]]
              precision    recall  f1-score   support

           0       0.97      0.38      0.55      2429
           1       0.51      0.98      0.67      1572

    accuracy                           0.62      4001
   macro avg       0.74      0.68      0.61      4001
weighted avg       0.79      0.62      0.59      4001


C:\Users\sd pro\AppData\Local\Programs\Python\Python37\lib\site-
packages\sklearn\utils\deprecation.py:87: FutureWarning: Function
plot_precision_recall_curve is deprecated; Function
`plot_precision_recall_curve` is deprecated in 1.0 and will be removed in 1.2.
Use one of the class methods: PrecisionRecallDisplay.from_predictions or
PrecisionRecallDisplay.from_estimator.
  warnings.warn(msg, category=FutureWarning)
```


2-class Precision-Recall curve: AP={0:0.2f}
Precision (Positive label: 1)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
GaussianNB (AP = 0.50)
0.0
0.2
0.4
0.6
0.8
1.0
Recall (Positive label: 1)


Receiver operating characteristic
True Positive Rate
1.0
0.8
0.6
0.4
0.2
0.0
(area = 0.68)
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate

[37]:

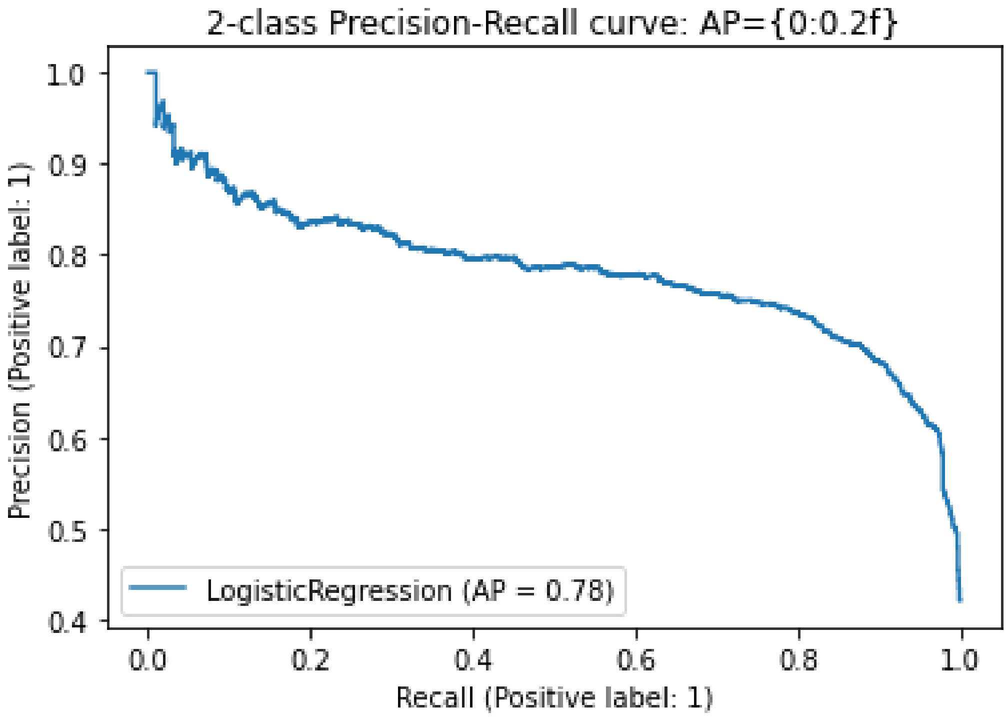
```
lgr = LogisticRegression()
lgr.fit(X_over, y_over)
y_pred = lgr.predict(X_test)
print("Accuracy: ",metrics.accuracy_score(y_test, y_pred))
print("Confusion Matrix: \n", confusion_matrix(y_test, y_pred))
getStatsFromModel(lgr)
```

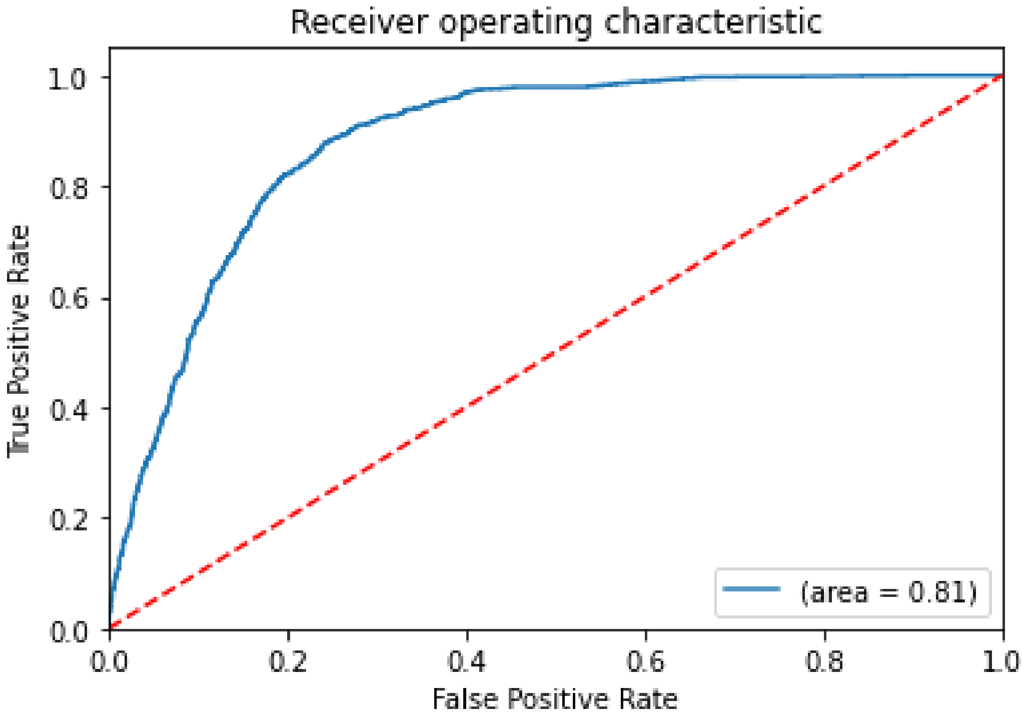
```
Accuracy:  0.8062984253936516
Confusion Matrix:
 [[1907  522]
 [ 253 1319]]
              precision    recall  f1-score   support

           0       0.88      0.79      0.83      2429
           1       0.72      0.84      0.77      1572

    accuracy                           0.81      4001
   macro avg       0.80      0.81      0.80      4001
weighted avg       0.82      0.81      0.81      4001


C:\Users\sd pro\AppData\Local\Programs\Python\Python37\lib\site-
packages\sklearn\utils\deprecation.py:87: FutureWarning: Function
plot_precision_recall_curve is deprecated; Function
`plot_precision_recall_curve` is deprecated in 1.0 and will be removed in 1.2.
Use one of the class methods: PrecisionRecallDisplay.from_predictions or
PrecisionRecallDisplay.from_estimator.
  warnings.warn(msg, category=FutureWarning)
```


2-class Precision-Recall curve: AP={0:0.2f}
Precision (Positive label: 1)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
LogisticRegression (AP = 0.78)
0.0
0.2
0.4
0.6
0.8
1.0
Recall (Positive label: 1)


Receiver operating characteristic
True Positive Rate
1.0
0.8
0.6
0.4
0.2
0.0
(area = 0.81)
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate

[38]:

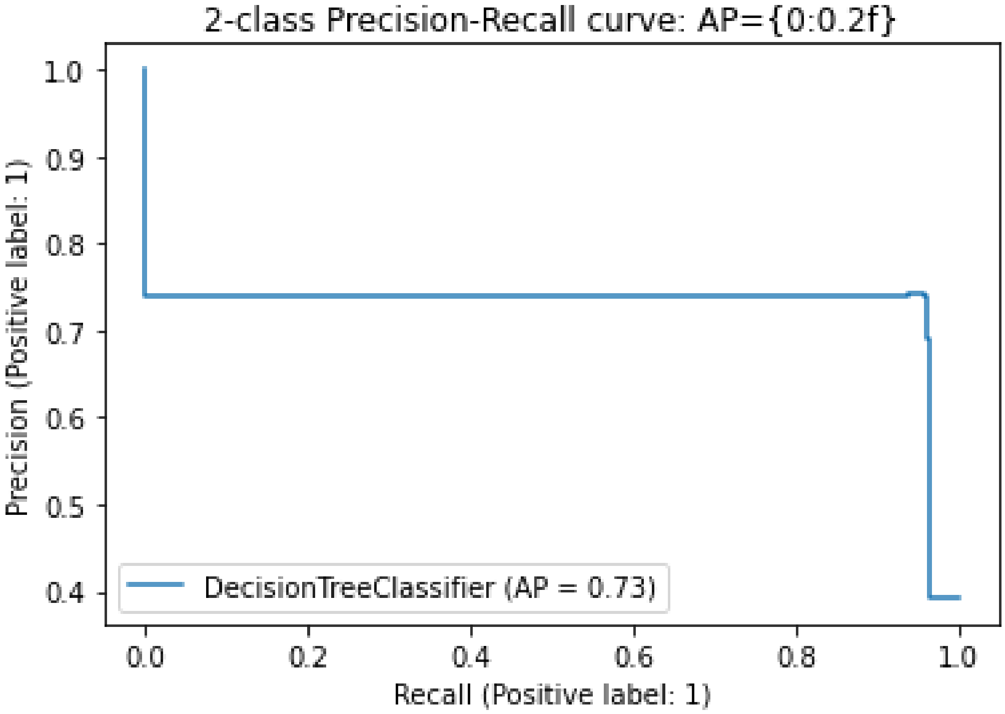
```
dtc = DecisionTreeClassifier()
dtc.fit(X_over, y_over)
y_pred = dtc.predict(X_test)
print("Accuracy: ",metrics.accuracy_score(y_test, y_pred))
print("Confusion Matrix: \n", confusion_matrix(y_test, y_pred))
getStatsFromModel(dtc)
```

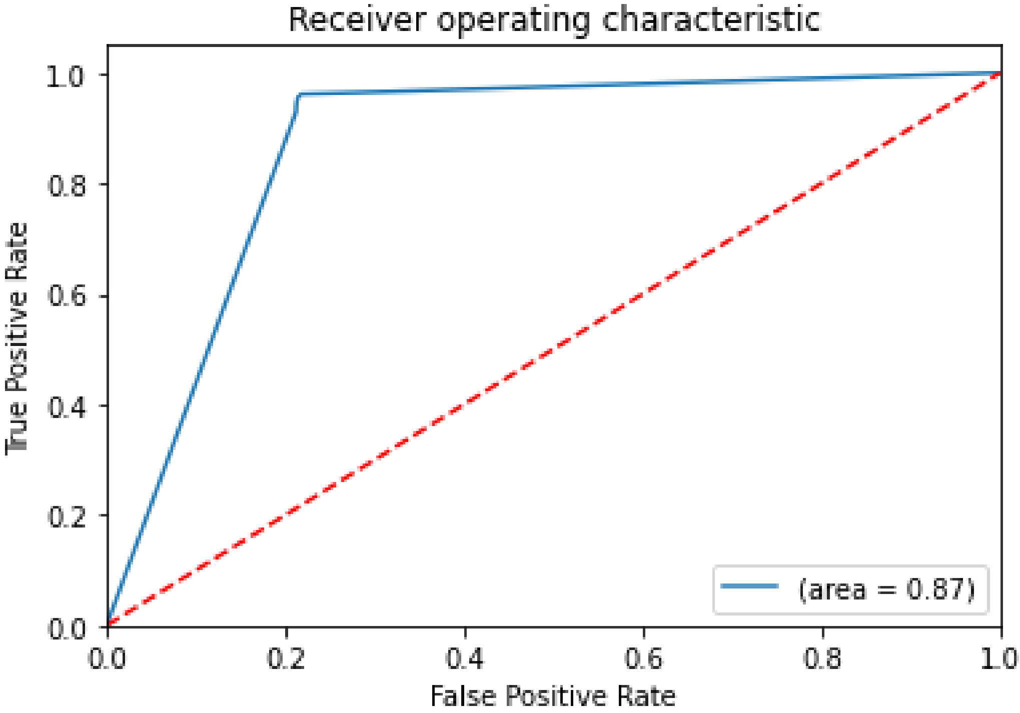
```
Accuracy:  0.8527868032991752
Confusion Matrix:
 [[1903  526]
 [  63 1509]]
              precision    recall  f1-score   support

           0       0.97      0.78      0.87      2429
           1       0.74      0.96      0.84      1572

    accuracy                           0.85      4001
   macro avg       0.85      0.87      0.85      4001
weighted avg       0.88      0.85      0.85      4001


C:\Users\sd pro\AppData\Local\Programs\Python\Python37\lib\site-
packages\sklearn\utils\deprecation.py:87: FutureWarning: Function
plot_precision_recall_curve is deprecated; Function
`plot_precision_recall_curve` is deprecated in 1.0 and will be removed in 1.2.
Use one of the class methods: PrecisionRecallDisplay.from_predictions or
PrecisionRecallDisplay.from_estimator.
  warnings.warn(msg, category=FutureWarning)
```


2-class Precision-Recall curve: AP={0:0.2f}
Precision (Positive label: 1)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
DecisionTreeClassifier (AP = 0.73)
0.0
0.2
0.4
0.6
0.8
1.0
Recall (Positive label: 1)


Receiver operating characteristic
True Positive Rate
1.0
0.8
0.6
0.4
0.2
0.0
(area = 0.87)
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate

5.2 Ensemble Methods

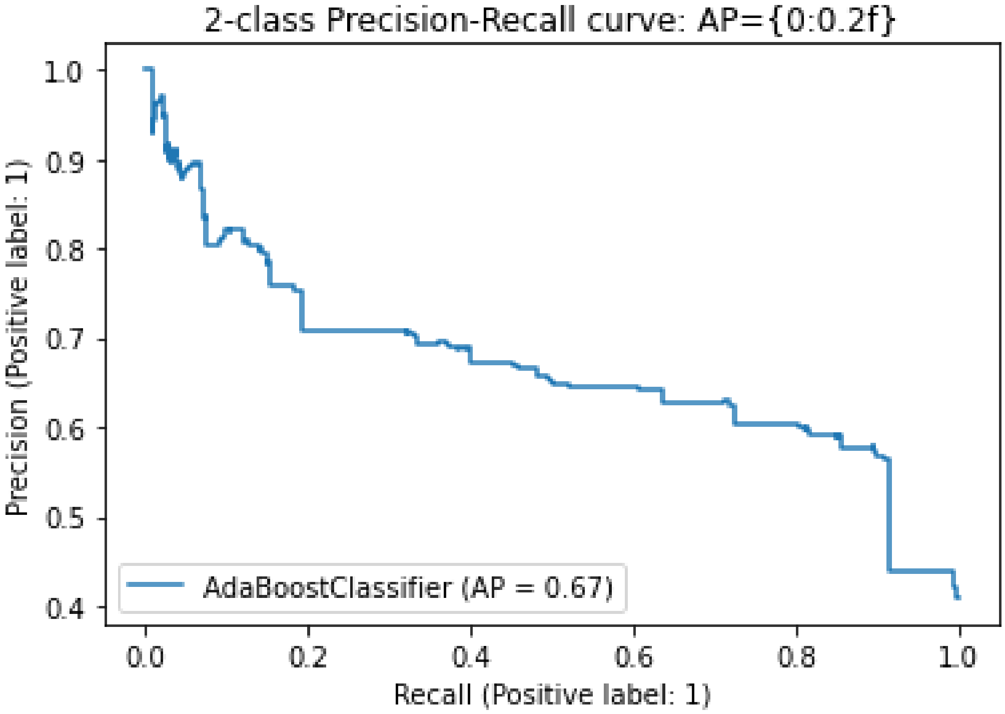
[39]:
```python
#Ensemble methods from here
abc = AdaBoostClassifier()
abc.fit(X_over, y_over)
y_pred = abc.predict(X_test)
print("Accuracy: ",metrics.accuracy_score(y_test, y_pred))
print("Confusion Matrix: \n", confusion_matrix(y_test, y_pred))
getStatsFromModel(abc)
```

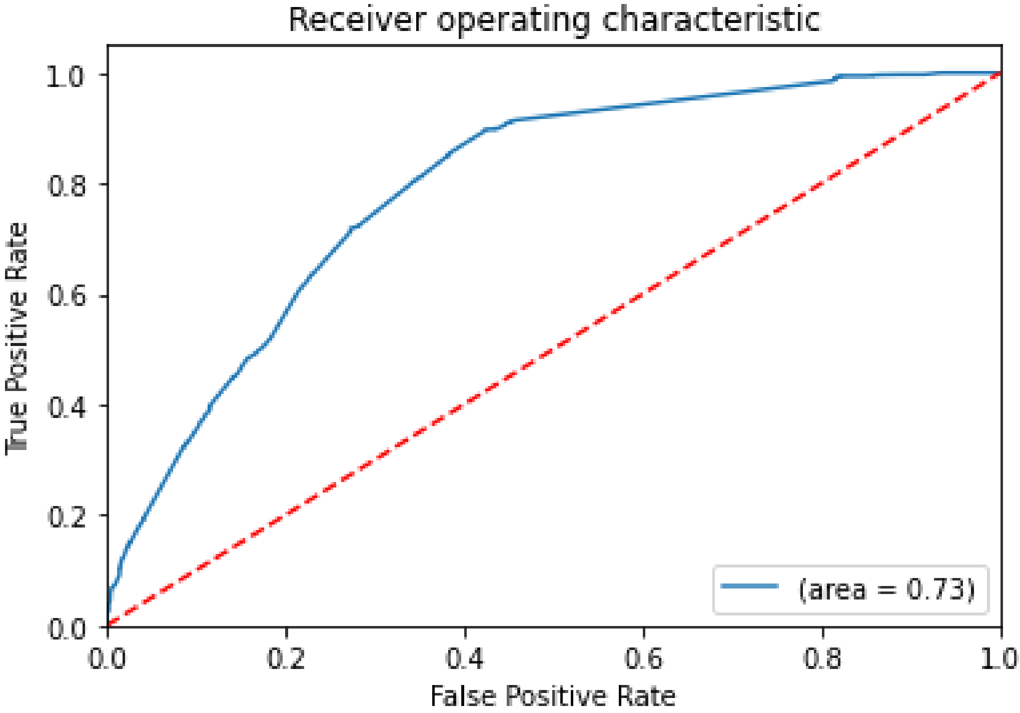
```
Accuracy:  0.7143214196450888
Confusion Matrix:
 [[1603  826]
 [ 317 1255]]
              precision    recall  f1-score   support

           0       0.83      0.66      0.74      2429
           1       0.60      0.80      0.69      1572

    accuracy                           0.71      4001
   macro avg       0.72      0.73      0.71      4001
weighted avg       0.74      0.71      0.72      4001


C:\Users\sd pro\AppData\Local\Programs\Python\Python37\lib\site-
packages\sklearn\utils\deprecation.py:87: FutureWarning: Function
plot_precision_recall_curve is deprecated; Function
`plot_precision_recall_curve` is deprecated in 1.0 and will be removed in 1.2.
Use one of the class methods: PrecisionRecallDisplay.from_predictions or
PrecisionRecallDisplay.from_estimator.
  warnings.warn(msg, category=FutureWarning)
```


2-class Precision-Recall curve: AP={0:0.2f}
Precision (Positive label: 1)
Recall (Positive label: 1)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.0
0.2
0.4
0.6
0.8
1.0
AdaBoostClassifier (AP = 0.67)


Receiver operating characteristic
True Positive Rate
False Positive Rate
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
(area = 0.73)

[40]:

```
rfc = RandomForestClassifier(verbose=True) #uses randomized decision trees
rfcmodel = rfc.fit(X_over, y_over)
y_pred = rfc.predict(X_test)
print ("Score:", rfcmodel.score(X_test, y_test))
print("Confusion Matrix: \n", confusion_matrix(y_test, y_pred))
getStatsFromModel(rfc)
```

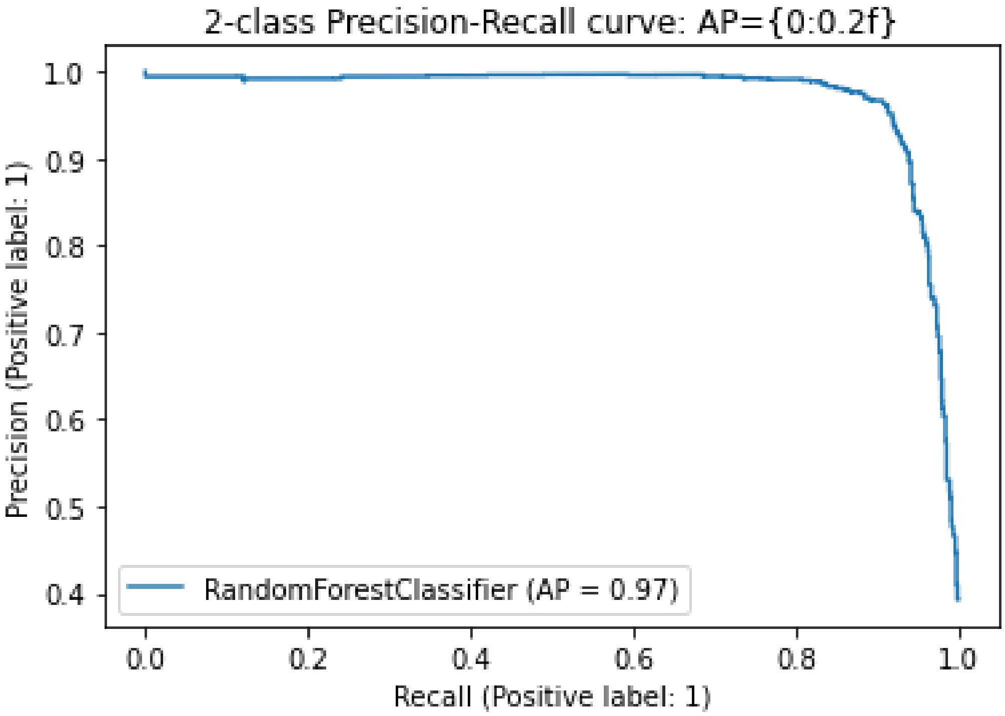
```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:  8.7min finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    1.4s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    1.1s finished
C:\Users\sd pro\AppData\Local\Programs\Python\Python37\lib\site-
packages\sklearn\utils\deprecation.py:87: FutureWarning: Function
plot_precision_recall_curve is deprecated; Function
`plot_precision_recall_curve` is deprecated in 1.0 and will be removed in 1.2.
Use one of the class methods: PrecisionRecallDisplay.from_predictions or
PrecisionRecallDisplay.from_estimator.
  warnings.warn(msg, category=FutureWarning)
```

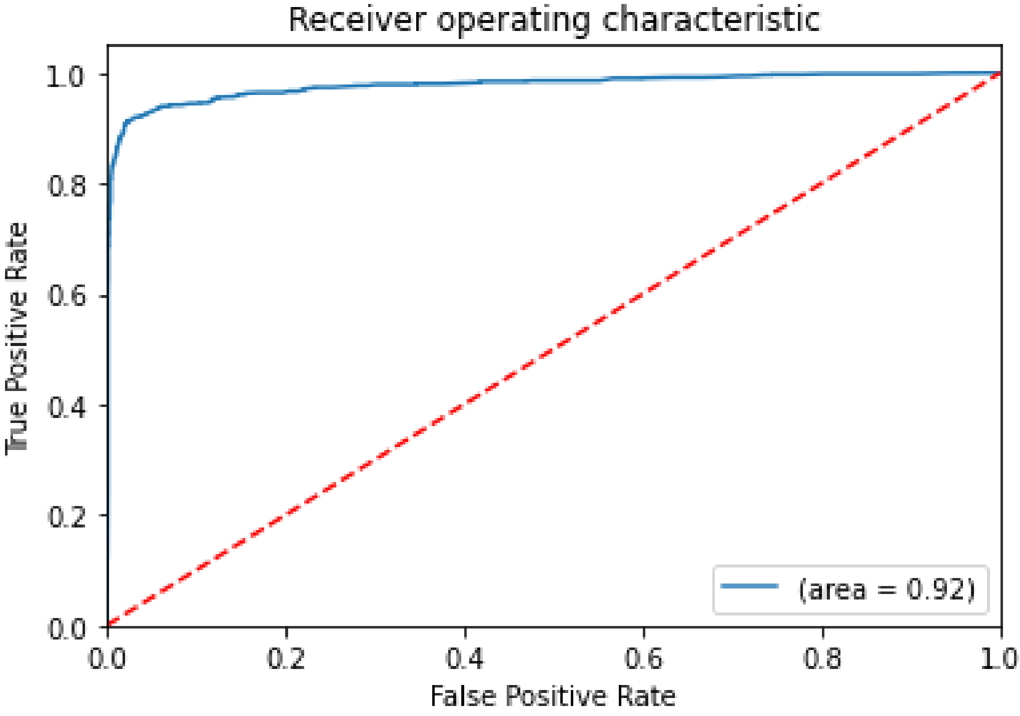
```
Score: 0.91552111972007
Confusion Matrix:
 [[2176  253]
 [  85 1487]]
              precision    recall  f1-score   support

           0       0.96      0.90      0.93      2429
           1       0.85      0.95      0.90      1572

    accuracy                           0.92      4001
   macro avg       0.91      0.92      0.91      4001
weighted avg       0.92      0.92      0.92      4001
```

```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    1.3s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    1.3s finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    1.0s finished
```


2-class Precision-Recall curve: AP={0:0.2f}
1.0
0.9
0.8
0.7
0.6
0.5
0.4
Precision (Positive label: 1)
RandomForestClassifier (AP = 0.97)
0.0
0.2
0.4
0.6
0.8
1.0
Recall (Positive label: 1)


Receiver operating characteristic
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
(area = 0.92)
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate

5.3 Requires High RAM and processing time - Not used

[41]:

```
# Model, SVM
from sklearn import svm


clf = svm.SVC(kernel='linear', verbose=True)
clf.fit(X_over, y_over)
y_pred = clf.predict(X_test)
print("Accuracy: ",metrics.accuracy_score(y_test, y_pred))
print("Confusion Matrix: \n", confusion_matrix(y_test, y_pred))
getStatsFromModel(clf)
```

```
[LibSVM]
```

[ ]:

```
from sklearn.neural_network import MLPClassifier
mlp = MLPClassifier(hidden_layer_sizes=(100,100,100,10), max_iter=200,␣
  ↪verbose=True)
mlp.fit(X_over,y_over)
print("Confusion Matrix: \n", confusion_matrix(y_test, y_pred))
getStatsFromModel(mlp)
```

[ ]: