project

October 22, 2019

1 NLP with ML

This project compines NLP techniques and ML supervised algorithms to find the sentiment of the famous movie review dataset. This project was part of class work at physical meetings of the NTL intiative.

1.1 Loading the dataset

First the a sample file of the dataset is loaded to check the dataset and the loading process. Then the whole dataset will be loaded.

[1]:
```python
# loading one review from the negative file, we can see that the text is not clean and require cleaning

import os

path='reviews/neg/cv000_29416.txt'
def load_file(path):

    file = open(path,'r')
    text = file.read()
    file.close()
    return text

load_file(path)
```

[1]:
```
'plot : two teen couples go to a church party , drink and then drive . \nthey
get into an accident . \none of the guys dies , but his girlfriend continues to
see him in her life , and has nightmares . \nwhat\'s the deal ? \nwatch the
movie and " sorta " find out . . . \ncritique : a mind-fuck movie for the teen
generation that touches on a very cool idea , but presents it in a very bad
package . \nwhich is what makes this review an even harder one to write , since
i generally applaud films which attempt to break the mold , mess with your head
and such ( lost highway & memento ) , but there are good and bad ways of making
all types of films , and these folks just didn\'t snag this one correctly .
\nthey seem to have taken this pretty neat concept , but executed it terribly .
\nso what are the problems with the movie ? \nwell , its main problem is that
```

```
it\'s simply too jumbled . \nit starts off " normal " but then downshifts into
this " fantasy " world in which you , as an audience member , have no idea
what\'s going on . \nthere are dreams , there are characters coming back from
the dead , there are others who look like the dead , there are strange
apparitions , there are disappearances , there are a looooot of chase scenes ,
there are tons of weird things that happen , and most of it is simply not
explained . \nnow i personally don\'t mind trying to unravel a film every now
and then , but when all it does is give me the same clue over and over again , i
get kind of fed up after a while , which is this film\'s biggest problem .
\nit\'s obviously got this big secret to hide , but it seems to want to hide it
completely until its final five minutes . \nand do they make things entertaining
, thrilling or even engaging , in the meantime ? \nnot really . \nthe sad part
is that the arrow and i both dig on flicks like this , so we actually figured
most of it out by the half-way point , so all of the strangeness after that did
start to make a little bit of sense , but it still didn\'t the make the film all
that more entertaining . \ni guess the bottom line with movies like this is that
you should always make sure that the audience is " into it " even before they
are given the secret password to enter your world of understanding . \ni mean ,
showing melissa sagemiller running away from visions for about 20 minutes
throughout the movie is just plain lazy ! ! \nokay , we get it . . . there \nare
people chasing her and we don\'t know who they are . \ndo we really need to see
it over and over again ? \nhow about giving us different scenes offering further
insight into all of the strangeness going down in the movie ? \napparently , the
studio took this film away from its director and chopped it up themselves , and
it shows . \nthere might\'ve been a pretty decent teen mind-fuck movie in here
somewhere , but i guess " the suits " decided that turning it into a music video
with little edge , would make more sense . \nthe actors are pretty good for the
most part , although wes bentley just seemed to be playing the exact same
character that he did in american beauty , only in a new neighborhood . \nbut my
biggest kudos go out to sagemiller , who holds her own throughout the entire
film , and actually has you feeling her character\'s unraveling . \noverall ,
the film doesn\'t stick because it doesn\'t entertain , it\'s confusing , it
rarely excites and it feels pretty redundant for most of its runtime , despite a
pretty cool ending and explanation to all of the craziness that came before it .
\noh , and by the way , this is not a horror or teen slasher flick . . . it\'s
\njust packaged to look that way because someone is apparently assuming that the
genre is still hot with the kids . \nit also wrapped production two years ago
and has been sitting on the shelves ever since . \nwhatever . . . skip \nit !
\nwhere\'s joblo coming from ? \na nightmare of elm street 3 ( 7/10 ) - blair
witch 2 ( 7/10 ) - the crow ( 9/10 ) - the crow : salvation ( 4/10 ) - lost
highway ( 10/10 ) - memento ( 10/10 ) - the others ( 9/10 ) - stir of echoes (
8/10 ) \n'
```

[2]:

```
#loading the whole dataset.
#adding all the files in the negative folder first then the postive to form 1 list of reviews
```

```python
folder='reviews'
rev=[]
for sent in os.listdir(folder):
    sentpath = folder+'/'+sent
    for fileid in os.listdir(sentpath):
        path = sentpath+'/'+fileid
        text=load_file(path)
        rev.append(text)
len(rev)
```

[2]:
```
2000
```

[3]:
```python
rev[100]
```

[3]:
```
"what do you get when you rip-off good movies like woody allen's bananas and
martin scorsese's after hours ? \nyou'd think you'd get the best of both films .
\ninstead you get woo . \nfalling in somewhere between def jam's how to be a
player ( which was awful ) and booty call ( which was ok ) , woo is yet another
in the embarassing genre of showing african-americans to be nothing more than
sexual buffoons . \nthe whole film plays out as a black version of after hours ,
as wild woman woo ( jada pinkett smith ) goes out on a blind date with straight-
laced tim ( tommy davidson ) . \nmayhem follows them . \nfor some unknown reason
( read : contrived screenplay ) davidson puts up with all of woo's antics for
the entire night , which include her destroying his bathroom mirror , stealing
things from his house , violently questioning him ( accusing and belittling him
actually ) about previous girlfriends , causing a riot in an elegant restaurant
, and other various infuriating things that any normal person wouldn't tolerate
. \nbut for the sake of this bad movie ? \nsure , why not ? \nthere are a few
chuckles in the film , the best being the scene swiped directly from bananas .
\nin this case , davidson is running from thugs , gets into a subway car as the
doors are closing , starts to taunt the thugs , then the doors open back up
again . \na good joke , but a stolen one . \nanother chuckle is provided by
billy dee williams' cameo as himself . \nmovies like woo are seemingly released
every three months or so , and not one of them has ever been a hit . \nwoo won't
be one either . \nso why was it made ? \nand more importantly , isn't there
anyone else besides me who thinks these films are offensive ? \neveryone
involved should really reconsider their careers at this point . \n[r] \n"
```

1.2 NLP

For this section, different nlp techniuqes are introduced to process the dataset. The dataset is normalized by cleaning unwanted charachters and removing numbers and special charchters, then the stop words will be removed and finally lemmatization will be applied to reduce the words to their stem/roots. Wordnet lemmatizer is used because it has a varaity of rules and steps compared with simpler stemmers. Finally the dataset will be vectorized using tf-idf technique, which maps the corpus words to vectors based on the frequncy of the word and its inverse frequncy to the dataset.

This method is better than the usuall bag of words method that relies on the count frequency alone.

[4]:

```
#cleaning the dataset, the order of this cleaning is improtant:
#removing special charachters first will leave \n as n which will overlap with other words.

import re
def get_clean(doc):
    #cleaning the new line \n
    doc=[re.sub('\n','',review) for review in doc]
    #remove the numbers
    doc=[re.sub('\d','',review) for review in doc]
    # remove any non alphabetical charchter
    doc=[re.sub('[^\w\s\']','',review) for review in doc]
    # remove empty spaces ruslted from previous cleaning stesp
    doc=[re.sub(r'\s\s+',' ',review) for review in doc]
    return doc
rev=get_clean(rev)
```

[5]:

```

rev[100]
```

[5]:

```
"what do you get when you ripoff good movies like woody allen's bananas and
martin scorsese's after hours you'd think you'd get the best of both films
instead you get woo falling in somewhere between def jam's how to be a player
which was awful and booty call which was ok woo is yet another in the
embarassing genre of showing africanamericans to be nothing more than sexual
buffoons the whole film plays out as a black version of after hours as wild
woman woo jada pinkett smith goes out on a blind date with straightlaced tim
tommy davidson mayhem follows them for some unknown reason read contrived
screenplay davidson puts up with all of woo's antics for the entire night which
include her destroying his bathroom mirror stealing things from his house
violently questioning him accusing and belittling him actually about previous
girlfriends causing a riot in an elegant restaurant and other various
infuriating things that any normal person wouldn't tolerate but for the sake of
this bad movie sure why not there are a few chuckles in the film the best being
the scene swiped directly from bananas in this case davidson is running from
thugs gets into a subway car as the doors are closing starts to taunt the thugs
then the doors open back up again a good joke but a stolen one another chuckle
is provided by billy dee williams' cameo as himself movies like woo are
seemingly released every three months or so and not one of them has ever been a
hit woo won't be one either so why was it made and more importantly isn't there
anyone else besides me who thinks these films are offensive everyone involved
should really reconsider their careers at this point r "
```

[6]:

```
#removing stop words as they don't transfer well with lemmatization
from nltk.corpus import stopwords
```

```python
stop = stopwords.words("english")

#the original list is too agressive, these words are important to show sentiment
stop=list(set(stop)-set(['did', 'didn', "didn't", 'do', 'don',␣
 ↪"don't","isn't"]))

#removing the stopwords from the whole corpus
for i in range(len(rev)):
    rev[i]=' '.join([word for word in rev[i].split() if word not in stop])
```

[7]:

```python
# using nltk to import the wordnet lemmatizer to reduce the words to their␣
 ↪roots.

import nltk
from nltk.stem import WordNetLemmatizer

lem = WordNetLemmatizer()

def rooting(doc):
# lemmataizing/rooting/stemming the words in each review
    for i in range(len(doc)):
        doc[i]=' '.join([lem.lemmatize(word) for word in doc[i].split(' ')])
    return doc

rev=rooting(rev)
rev[0]
```

[7]:

```
"original sin road screen rocky initially slated release last november film
bumped twice finally landing dog day summer advance screening film denied critic
generally sign studio realizes dud hand original sin really bad yes melodrama
offer reward location setting gorgeous healthy sprinkling ta angelina jolie
providing antonio banderas importantly movie entertainingly bad veteran reader
know rule don't encourage people patronize lousy film time plenty quality
offering marketplace deserving money besides let's go laugh failing others
mindset reflects elitism make uncomfortable thing different summer quality film
put mildly far far i'm concerned fair find kick may original sin never join
treasure valley doll road house showgirl bad movie hall fame it'll do something
worse come along film adapted director michael cristofer cornell woolrich novel
waltz darkness also source francois truffaut film mississippi mermaid open
turnofthecentury prison jolie's character slated dawn execution tell lurid tale
priest appears desperately horny freshman writing class tone quickly established
say thing like love story story love wary local golddiggers cuban coffee dealer
luis antonio vargas banderas make arrangement secure mail order bride america
listing mere clerk dissuade foreign golddiggers practical man luis chooses
frumpy looking lady hoping loyal mate able provide child imagine surprise fianc
e julia russell jolie turn infinitely attractive woman photo julia explains sent
different woman's image didn't want selected solely pretty face luis confesses
```

```
deception leading julie state great significance something common trusted
wedding luis julia retire glorious night carefully choreographed lovemaking body
positioned display breast bottom erotically possible jolie banderas attractive
people watching naked fun although filmmakers' insistence using one banderas'
leg cover jolie's crotch make look like he's trying climb luis stupidest man
ever lived immediately instructs bank make personal business account available
julia despite fact seems nothing like woman corresponded blissful ignorance
continues warning sign mount luis must force julia write sister emily frantic
lack communication shortly julia complains chirping pet bird found floor cage
broken neck finally clean account disappears luis begin suspect something might
wrong incidentally afraid i'm giving much away rest assured happens first minute
movie leaving plenty time numerous dopey plot twist great deal operatic acting
footage tit as along way private detective walter down played thomas jane
terrific mickey mantle hbo movie turn hired frumpy woman's sister find happened
real julia luis also eager detective track con artist decided can't kill oh
pathos cast appears recognize trashiness story adjusting performance accordingly
banderas suitably impassioned jolie alternate vamping pouting lip really pout
thomas jane start acting suspicious cagey accelerates snidely whiplash level
nastiness startling moment come prove power humiliate force luis wall verbally
taunt rubbing cheek luis finish establishing dominance fullon kiss anyone ever
question difference sex rape show chilling scene anyone ever question difference
real drama laughable potboiler show original sin"
```

[8]:

```
rev[100]
```

[8]:

```
"do get ripoff good movie like woody allen's banana martin scorsese's hour think
get best film instead get woo falling somewhere def jam's player awful booty
call ok woo yet another embarassing genre showing africanamericans nothing
sexual buffoon whole film play black version hour wild woman woo jada pinkett
smith go blind date straightlaced tim tommy davidson mayhem follows unknown
reason read contrived screenplay davidson put woo's antic entire night include
destroying bathroom mirror stealing thing house violently questioning accusing
belittling actually previous girlfriend causing riot elegant restaurant various
infuriating thing normal person tolerate sake bad movie sure chuckle film best
scene swiped directly banana case davidson running thug get subway car door
closing start taunt thug door open back good joke stolen one another chuckle
provided billy dee williams' cameo movie like woo seemingly released every three
month one ever hit woo one either made importantly isn't anyone else besides
think film offensive everyone involved really reconsider career point r"
```

[9]:

```
#vectorization

from sklearn.feature_extraction.text import TfidfVectorizer
vec=TfidfVectorizer()

data=vec.fit_transform(rev).toarray()
```

1.3 ML

In this section we will train a simple ml modle `MultinomialNB` on the training set and then compare its performance with a more complex modle that will be optimized `RandomForest`. First the dataset is split into training and testing sets after it got cleaned. Then the simple modle is trained and tested and the complex modle will be trained and optimized (using grid search) and compared with the simple one. finally the solution modle will be choosen.

[10]:

```
#splitting the dataset

import numpy as np
import random
from sklearn.model_selection import train_test_split

labels=[0 if i<1000 else 1 for i in range(2000)]
labels = np.asarray(labels)


idx=np.arange(len(rev))
np.random.shuffle(idx)

X=data[idx]
Y=labels[idx]


x_train, x_test, y_train,  y_test =train_test_split(X,Y,test_size=0.
 ↪33,random_state=42)
print(x_train.shape[0],x_test.shape[0],y_test.shape[0],y_train.shape[0])
```

```
1340 660 660 1340
```

[11]:

```
#importing the simple modle

from sklearn.naive_bayes import MultinomialNB
clf = MultinomialNB().fit(x_train, y_train)
```

[13]:

```
#results of the simple modle

from sklearn.metrics import classification_report,accuracy_score

predicted = clf.predict(x_test)
print('classification report :\n {} \n Model Acurracy = {}'.
 ↪format(classification_report(y_test,predicted),
                                                                ␣
 ↪accuracy_score(y_test,predicted)))
```

```
classification report :
               precision    recall  f1-score   support
```

```
           0       0.64      0.94      0.76       313
           1       0.90      0.53      0.67       347

    accuracy                           0.72       660
   macro avg       0.77      0.73      0.72       660
weighted avg       0.78      0.72      0.71       660

 Model Acurracy = 0.7242424242424242
```

[14]:

```
# the complex modle development with the grid search steps and final evaluation

from sklearn.model_selection import GridSearchCV
from sklearn.metrics import make_scorer,fbeta_score
from sklearn.ensemble import RandomForestClassifier
#from sklearn.cross_validation import ShuffleSplit

clfer = RandomForestClassifier(random_state=42)

parameters = {
    'n_estimators':[100, 350]
    ,'max_depth':[5,6,7,8,9]
    ,'criterion': ['gini','entropy']}

scorer=make_scorer(fbeta_score,beta=2)

grid_obj=GridSearchCV(clfer,parameters,scoring=scorer,cv=5)

grid_fit= grid_obj.fit(x_train,y_train)

best_clf= grid_fit.best_estimator_

pred=(clfer.fit(x_train,y_train).predict(x_test))
bestpred=best_clf.predict(x_test)
```

```
/usr/local/lib64/python3.7/site-packages/sklearn/ensemble/forest.py:245:
FutureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
```

[15]:

```
# results of the complex modle

print('Unoptimized_modle \n *******')
print('Accuracy score on testing data: {:.4f}'.
 ↪format(accuracy_score(y_test,pred)))
print('f-score on testing data: {:.4f}'.format(fbeta_score(y_test,pred,beta=0.
 ↪5)))
```

```

print('Optimized_modle \n *******')
print('Accuracy score on testing data: {:.4f}'.
 ↪format(accuracy_score(y_test,bestpred)))
print('f-score on testing data: {:.4f}'.
 ↪format(fbeta_score(y_test,bestpred,beta=0.5)))
```

```
Unoptimized_modle
 *******
Accuracy score on testing data: 0.6485
f-score on testing data: 0.6806
Optimized_modle
 *******
Accuracy score on testing data: 0.7788
f-score on testing data: 0.8192
```

```
[16]: #getting the optimized parameters

      for i in parameters.keys():
          print(best_clf.get_params()[i])
```

```
350
9
gini
```

We can see that the best modle was maximizing the parameters, this means there maybe be better parameters, so trying different parameters this is the final modle configuration.

```
[17]: best = RandomForestClassifier(n_estimators=400,max_depth=15,criterion='entropy'␣
       ↪,random_state=42)
      predics=best.fit(x_train,y_train).predict(x_test)

      print('Accuracy score on testing data: {:.4f}'.
       ↪format(accuracy_score(y_test,predics)))
      print('f-score on testing data: {:.4f}'.
       ↪format(fbeta_score(y_test,predics,beta=0.5)))
```

```
Accuracy score on testing data: 0.8015
f-score on testing data: 0.8296
```

1.4 Results

Looking at the table below we can see a summry of the performance of the modles:

Modle	Accuracy (%)	Fbeta (%)
MultinomialNB	72.4	71.5
RandomForest	64.9	68.0

Modle	Accuracy (%)	Fbeta (%)
RandomForest opt	80.2	83.0

we can see that the best modle is the optimized random forest classifier, with a clear difference compared with the simple modle. Also the unoptimized random forest classifier performed very badly, this implies that optimization is critical for this modle working with nlp task. these results are close to the limits of the ml modles. In general if you have a small dataset consider using a complex ml modle after optimization.