Wiki_Toxicity

July 24, 2021

[1]:
```
# Course | PG DS- Natural Language Processing
# Author: Bakul Purohit # Date: 24th July 2021

# Domain : Wikipedia Toxicity Analysis
# Project 2: Using NLP and machine learning, make a model to identify toxic␣
 ↪comments
# from the Talk edit pages on Wikipedia.Help identify the words that make a␣
 ↪comment toxic.
```

[2]:
```
## Wiki reviews Toxicity Analysis with NLP
## Techniques deployed in this code
## Tokenization: breaking text into tokens (words, sentences, n-grams)
## Stopword removal: a/an/the
## Stemming and lemmatization: root word
## TF-IDF: word importance--> Term Frequency and Inverse Document Frequency
```

0.1 Step 1: Start by reading in the libraries and also Input Reviews File (train.csv)

[3]:
```
# Filter warnings which are not necessarily errors/ exceptions
import warnings
warnings.filterwarnings('ignore')
```

[4]:
```
# Import requisite libraries

import pandas as pd
import numpy as np
import re
```

0.2 Step 2: Load and observe the data using Describe function and extract comments from the data in a list

[5]:
```
# Q.1. --> Load the data using read_csv function from pandas package
Wiki = pd.read_csv('train.csv')
Wiki.head()
```

[5]:

	id	comment_text	toxic
0	e617e2489abe9bca	"\r\n\r\n A barnstar for you! \r\n\r\n The De…	0
1	9250cf637294e09d	"\r\n\r\nThis seems unbalanced. whatever I ha…	0
2	ce1aa4592d5240ca	Marya Dzmitruk was born in Minsk, Belarus in M…	0
3	48105766ff7f075b	"\r\n\r\nTalkback\r\n\r\n Dear Celestia… "	0
4	0543d4f82e5470b6	New Categories \r\n\r\nI honestly think that w…	0

[6]:
```
Wiki. shape
```

[6]:
```
(5000, 3)
```

[7]:
```
Wiki.describe
```

[7]:
```
<bound method NDFrame.describe of                     id
comment_text  \
0     e617e2489abe9bca  "\r\n\r\n A barnstar for you! \r\n\r\n  The De…
1     9250cf637294e09d  "\r\n\r\nThis seems unbalanced.  whatever I ha…
2     ce1aa4592d5240ca  Marya Dzmitruk was born in Minsk, Belarus in M…
3     48105766ff7f075b      "\r\n\r\nTalkback\r\n\r\n Dear Celestia…  "
4     0543d4f82e5470b6  New Categories \r\n\r\nI honestly think that w…
…                  …                                                …
4995  60229df7b48ba6ff  "\r\n\r\n Dildo, if you read my response corre…
4996  36a645227572ec5c         CALM DOWN, CALM DOWN, DON'T GET A BIG DICK
4997  6d47fa39945ed6f5  In my opinion Dougweller is using his privileg…
4998  de2e4c0d38db6e30  The style section has been expanded too. I did…
4999  4cda24210a33ac35  ANY ONE THAT IS NOT AGREEMENT WITH YOU OR IS A…

      toxic
0         0
1         0
2         0
3         0
4         0
…     …
4995      0
4996      1
4997      0
4998      0
4999      0
```

```
[5000 rows x 3 columns]>
```

[8]:
```
#Q.2 Get the comments into a list, for easy text cleanup and manipulation
# my_list will contain all comments
my_list = Wiki.comment_text.values
```

[9]:
```
my_list
```

[9]:
```
array(['"\r\n\r\n A barnstar for you! \r\n\r\n  The Defender of the Wiki
Barnstar I like your edit on the Kayastha page. Lets form a solidarity group
against those who malign the article and its subject matter. I propose the
folloing name for the group.\r\n\r\nUnited intellectuals\' front of Kayastha
ethinicty against racist or castist abuse (UIFKEARCA)   "',
       '"\r\n\r\nThis seems unbalanced.  whatever I have said about Mathsci, he
has said far more extreme and unpleasant things about me (not to mention
others), and with much greater frequency.  I\'m more than happy to reign myself
in, if that\'s what you\'d like (ruth be told, I was just trying to get Mathsci
to pay attention and stop being uncivil).  I would expect you to issue the same
request to Mathsci.  \r\n\r\n If this is intentionally unbalanced (for whatever
reason), please let me know, and I will voluntarily close this account and move
on to other things.  I like wikipedia, and I have a lot to contribute in my own
way, but there is no point contributing to the project if some editors have
administrative leave to be aggressively rude.  I\'m a good editor, and I don\'t
really deserve to have people riding my ass every time I try to do certain
things.  I\'ll happily leave it in the hands of the drama-prone, if that\'s what
you think is best.  Ludwigs2 "',
       'Marya Dzmitruk was born in Minsk, Belarus in March 19, 1992. Her mother,
Olga Nikolaevna Moroz was born in Baranovichi, Belarus and her father was born
in Brest, Belarus. She is second child in the family. Her parents divorced in
1998 and soon after her father remarried and had two more children. \r\nMarya,
at the age of 4 began doing gymnastics, but quit two years later because she was
denied a medal in a competition, where her age was incorrectly marked. When she
turned 6 years old, she got admitted to Music School #4 in Minsk, class of
violin, and to Public School #66 with piano classes as a main course. At the age
of 11, Marya starred in Belarusfilm movie called "Dunechka". Soon after she
started to play in theatre and was featured in television shows. By 2005 her
mother decided to move to United States. In September of 2005 Marya went to her
first American school, Ingrid B. Lacy Middle School. She graduated in Spring
2006 and traveled back to Belarus for 2 months. In August 2006 she went to
Oceana High School, from which she will graduate in 2010. \r\n\r\nMarya Dzmitruk
is a member of ISAR (International Society for Astrological Research), also a
member of a non-profit government organization which deals with human rights
abuse throughout the world (also known as Helsinki Committee). Marya holds two
diplomas from music schools, four scholarships for Lisa Spector's Music school
and several awards from YLI (Youth Leadership Institute). \r\n\r\nMarya has a
very close relationship with her mother. Her personal life is "as happy as it
could possibly get" a source says. She is currently dating Alex K., from Odessa,
```

```
Ukraine. \r\n\r\nMarya currently attends school and works for ISAR and Helsinki
Committee at her own free time.',
       …,
       'In my opinion Dougweller is using his privileges poorly, for personal
attack, or to play games and make a point.  It is my opinio that he should be
blocked for abusing these priveleges',
       "The style section has been expanded too. I didn't remember, but this was
how I placed the tag.",
       "ANY ONE THAT IS NOT AGREEMENT WITH YOU OR IS A REPULICAN IS jOE
HAZELTON.\r\n\r\nIT'S WACK A MOLE TIME… AND REMEMBER YOUR EDITS ARE BEING
LOOKED AT.."],
      dtype=object)
```

0.3 Step 3: Data Cleanup

[10]:
```
# Q.3 Cleanup Tasks
```

[11]:
```
# Q.3 a: Using regular expressions, remove IP addresses
```

[12]:
```
comments_ip_drop = [re.sub('[\d+\.{3}]\d+',"",txt) for txt in my_list]
```

[13]:
```
# Q.3 b: Using regular expressions, remove URLs
```

[14]:
```
comments_url_drop = [re.sub("\w+://\S+","", txt) for txt in comments_ip_drop]
```

[15]:
```
# Q.3 c: Normalize the Casing
# Here we have converted comments to lower case as a part of Normalizing
```

[16]:
```
comments_lower = [txt.lower() for txt in comments_url_drop]
```

[17]:
```
comments_lower[1:3]
```

[17]:
```
['"\r\n\r\nthis seems unbalanced.  whatever i have said about mathsci, he has
said far more extreme and unpleasant things about me (not to mention others),
and with much greater frequency.  i\'m more than happy to reign myself in, if
that\'s what you\'d like (ruth be told, i was just trying to get mathsci to pay
attention and stop being uncivil).  i would expect you to issue the same request
to mathsci.  \r\n\r\n if this is intentionally unbalanced (for whatever reason),
please let me know, and i will voluntarily close this account and move on to
other things.  i like wikipedia, and i have a lot to contribute in my own way,
but there is no point contributing to the project if some editors have
administrative leave to be aggressively rude.  i\'m a good editor, and i don\'t
really deserve to have people riding my ass every time i try to do certain
things.  i\'ll happily leave it in the hands of the drama-prone, if that\'s what
you think is best.  ludwigs2 "',
```

```
 'marya dzmitruk was born in minsk, belarus in march , . her mother, olga
nikolaevna moroz was born in baranovichi, belarus and her father was born in
brest, belarus. she is second child in the family. her parents divorced in  and
soon after her father remarried and had two more children. \r\nmarya, at the age
of 4 began doing gymnastics, but quit two years later because she was denied a
medal in a competition, where her age was incorrectly marked. when she turned 6
years old, she got admitted to music school #4 in minsk, class of violin, and to
public school # with piano classes as a main course. at the age of , marya
starred in belarusfilm movie called "dunechka". soon after she started to play
in theatre and was featured in television shows. by  her mother decided to move
to united states. in september of  marya went to her first american school,
ingrid b. lacy middle school. she graduated in spring  and traveled back to
belarus for 2 months. in august  she went to oceana high school, from which she
will graduate in . \r\n\r\nmarya dzmitruk is a member of isar (international
society for astrological research), also a member of a non-profit government
organization which deals with human rights abuse throughout the world (also
known as helsinki committee). marya holds two diplomas from music schools, four
scholarships for lisa spector's music school and several awards from yli (youth
leadership institute). \r\n\r\nmarya has a very close relationship with her
mother. her personal life is "as happy as it could possibly get" a source says.
she is currently dating alex k., from odessa, ukraine. \r\n\r\nmarya currently
attends school and works for isar and helsinki committee at her own free time.']
```

[18]:
```
# Final Cleanup Task
# Remove Extra Line Breaks

comments_cleaned = [txt.replace("\'","") for txt in comments_lower]
```

[19]:
```
# Q.3 d: Tokenization
```

[20]:
```
from nltk.tokenize import word_tokenize
```

[21]:
```
print(word_tokenize(comments_cleaned[0]))
```

```
['``', 'a', 'barnstar', 'for', 'you', '!', 'the', 'defender', 'of', 'the',
'wiki', 'barnstar', 'i', 'like', 'your', 'edit', 'on', 'the', 'kayastha',
'page', '.', 'lets', 'form', 'a', 'solidarity', 'group', 'against', 'those',
'who', 'malign', 'the', 'article', 'and', 'its', 'subject', 'matter', '.', 'i',
'propose', 'the', 'folloing', 'name', 'for', 'the', 'group', '.', 'united',
'intellectuals', 'front', 'of', 'kayastha', 'ethinicty', 'against', 'racist',
'or', 'castist', 'abuse', '(', 'uifkearca', ')', '``']
```

[22]:
```
comment_tokenize = [word_tokenize(sent) for sent in comments_cleaned]
print(comment_tokenize[0])
```

```
['``', 'a', 'barnstar', 'for', 'you', '!', 'the', 'defender', 'of', 'the',
'wiki', 'barnstar', 'i', 'like', 'your', 'edit', 'on', 'the', 'kayastha',
```

```
'page', '.', 'lets', 'form', 'a', 'solidarity', 'group', 'against', 'those',
'who', 'malign', 'the', 'article', 'and', 'its', 'subject', 'matter', '.', 'i',
'propose', 'the', 'folloing', 'name', 'for', 'the', 'group', '.', 'united',
'intellectuals', 'front', 'of', 'kayastha', 'ethinicty', 'against', 'racist',
'or', 'castist', 'abuse', '(', 'uifkearca', ')', '``']
```

[23]:
```
# Q.3 d: Remove stop words and punctuations
```

[24]:
```
from nltk.corpus import stopwords
from string import punctuation
```

[25]:
```
stop_nltk = stopwords.words("english")
stop_punct = list(punctuation)
```

[26]:
```
# This is a final list of Final Stop Words which will be run after next step on␣
 ↪comments list
stop_final = stop_nltk + stop_punct + ["...", "``","''", "====", "must"]
```

[27]:
```
# Q.3 e Define a function to perform all these steps, you'll use this later on␣
 ↪the actual test set
# Here a function named del_stop is being declared

def del_stop(sent):
    return [term for term in sent if term not in stop_final]
```

[28]:
```
# Pass our tokenized comment list's first comment through this function

del_stop(comment_tokenize[1])
```

[28]:
```
['seems',
 'unbalanced',
 'whatever',
 'said',
 'mathsci',
 'said',
 'far',
 'extreme',
 'unpleasant',
 'things',
 'mention',
 'others',
 'much',
 'greater',
 'frequency',
 'im',
 'happy',
 'reign',
```

```
'thats',
'youd',
'like',
'ruth',
'told',
'trying',
'get',
'mathsci',
'pay',
'attention',
'stop',
'uncivil',
'would',
'expect',
'issue',
'request',
'mathsci',
'intentionally',
'unbalanced',
'whatever',
'reason',
'please',
'let',
'know',
'voluntarily',
'close',
'account',
'move',
'things',
'like',
'wikipedia',
'lot',
'contribute',
'way',
'point',
'contributing',
'project',
'editors',
'administrative',
'leave',
'aggressively',
'rude',
'im',
'good',
'editor',
'dont',
'really',
```

```
 'deserve',
 'people',
 'riding',
 'ass',
 'every',
 'time',
 'try',
 'certain',
 'things',
 'ill',
 'happily',
 'leave',
 'hands',
 'drama-prone',
 'thats',
 'think',
 'best',
 'ludwigs2']
```

[29]:
```
# Pass entire comments list to fetch a ready list for further Data Processing

comments_final = [del_stop(sent) for sent in comment_tokenize]
```

0.4 Step 4: Analyze the Top Terms in Dataset

[30]:
```
### Q. 4--> Part 1 Checking out the top terms in the data
```

[31]:
```
from collections import Counter
```

[32]:
```
term_list = []
for sent in comments_final:
    term_list.extend(sent)
```

[33]:
```
res = Counter(term_list)
res.most_common(20)
```

[33]:
```
[('article', 1655),
 ('page', 1495),
 ('wikipedia', 1339),
 ('talk', 1171),
 ('please', 1038),
 ('ass', 986),
 ('would', 964),
 ('fuck', 907),
 ('one', 858),
```

```
 ('like', 836),
 ('dont', 780),
 ('also', 657),
 ('think', 630),
 ('see', 630),
 ('know', 595),
 ('im', 562),
 ('edit', 560),
 ('use', 549),
 ('articles', 549),
 ('people', 538)]
```

[34]:
```
### Q. 4--> Part 2 Identifying and dropping contextual stop wordss from the data
stop_context = ["article", "page", "wikipedia", "talk", "articles", "pages"]
```

[35]:
```
# Final Stop Word pool becomes the prior Stopw words and punctuations along␣
 ↪with these contextual words
stop_final = stop_final + stop_context
```

[36]:
```
comments_final = [del_stop(sent) for sent in comment_tokenize]
```

[37]:
```
comments_final = [" ".join(sent) for sent in comments_final]
comments_final[:2]
```

[37]:
```
['barnstar defender wiki barnstar like edit kayastha lets form solidarity group
malign subject matter propose folloing name group united intellectuals front
kayastha ethinicty racist castist abuse uifkearca',
 'seems unbalanced whatever said mathsci said far extreme unpleasant things
mention others much greater frequency im happy reign thats youd like ruth told
trying get mathsci pay attention stop uncivil would expect issue request mathsci
intentionally unbalanced whatever reason please let know voluntarily close
account move things like lot contribute way point contributing project editors
administrative leave aggressively rude im good editor dont really deserve people
riding ass every time try certain things ill happily leave hands drama-prone
thats think best ludwigs2']
```

0.5 Step 5: Model Building, Evaluation and Optimization

[38]:
```
#Q. 5 Separate X and Y and perform train test split, 70-30
```

[39]:
```
len(comments_final)
```

[39]:
```
5000
```

[40]:
```
# define X and Y
X = comments_final
y = Wiki.toxic
```

[41]:
```
# split the new DataFrame into training and testing sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =0.
↪3,random_state=1)
```

[42]:
```
# Q. 6 Use TF-IDF values for the terms as feature to get into a vector space␣
↪model
# Import TF-IDF vectorizer from sklearn
# Instantiate with a maximum of 4000 terms in your vocabulary
# Fit and apply on the train set
# Apply on the test set
```

[43]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
```

[44]:
```
vectorizer = TfidfVectorizer(max_features = 4000)
```

len(X_train), len(X_test)

[45]:
```
X_train_bow = vectorizer.fit_transform(X_train)
```

[46]:
```
X_test_bow = vectorizer.transform(X_test)
```

[47]:
```
X_train_bow.shape, X_test_bow.shape
```

[47]:
```
((3500, 4000), (1500, 4000))
```

[48]:
```
# Q.7 through 10 --> Below section includes Model Building, Model Evaluation␣
↪and then adjustments of class
#imbalance
```

[49]:
```
# Q. 7, 8 Model building: Support Vector Machine
# Instantiate SVC from sklearn with a linear kernel
# Fit on the train data
# Make predictions for the train and the test set
```

[50]:
```
from sklearn import svm
```

[51]:
```
# Build an SVM Linear Classifier
classifier_linear = svm.SVC(kernel='linear')
```

[52]:
```
%%time
classifier_linear.fit(X_train_bow, y_train)
```

```
CPU times: user 616 ms, sys: 16 ms, total: 632 ms
Wall time: 633 ms
```

[52]:
```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='linear',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

[53]:
```
y_train_pred = classifier_linear.predict(X_train_bow)
```

[54]:
```
y_train_pred[:5]
```

[54]:
```
array([0, 0, 0, 1, 0])
```

[55]:
```
from sklearn.metrics import classification_report
```

[56]:
```
print(classification_report(y_train, y_train_pred))
```

```
              precision    recall  f1-score   support

           0       0.97      1.00      0.99      3194
           1       1.00      0.70      0.82       306

    accuracy                           0.97      3500
   macro avg       0.98      0.85      0.90      3500
weighted avg       0.97      0.97      0.97      3500
```

[57]:
```
# Q. 9, 10 Adjust Class Imbalance and re-evaluate
```

[58]:
```
# Adjusting the class weights to improve the recall for the label
```

[59]:
```
classifier_linear = svm.SVC(kernel='linear', class_weight="balanced")
```

[60]:
```
%%time
classifier_linear.fit(X_train_bow, y_train)
```

```
CPU times: user 796 ms, sys: 8 ms, total: 804 ms
Wall time: 865 ms
```

[60]:
```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight='balanced', coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='linear',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

[61]:
```
y_train_pred = classifier_linear.predict(X_train_bow)
```

[62]:
```
print(classification_report(y_train, y_train_pred))
```

```
                precision    recall  f1-score   support

           0       1.00      0.99      0.99      3194
           1       0.90      0.99      0.94       306

    accuracy                           0.99      3500
   macro avg       0.95      0.99      0.97      3500
weighted avg       0.99      0.99      0.99      3500
```

0.6 Step 6: Hyper Param Tuning

[63]:
```
### Q.11 through 13 Hyper-parameter tuning
```

[64]:
```
from sklearn.model_selection import GridSearchCV, StratifiedKFold
```

[65]:
```
# Create the parameter grid based on the results of random search
param_grid = {
    'C': [0.1, 1, 10,1000, 10000, 100000]
}
```

[66]:
```
classifier_svm = svm.SVC(random_state=42, class_weight="balanced",␣
↪kernel="linear")
```

[67]:
```
# Instantiate the grid search model
grid_search = GridSearchCV(estimator = classifier_svm, param_grid = param_grid,
                           cv = StratifiedKFold(5), n_jobs = -1, verbose = 1,␣
↪scoring = "recall" )
```

[68]:
```
grid_search.fit(X_train_bow, y_train)
```

```
Fitting 5 folds for each of 6 candidates, totalling 30 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 2 concurrent workers.
[Parallel(n_jobs=-1)]: Done  30 out of  30 | elapsed:   11.1s finished
```

[68]:
```
GridSearchCV(cv=StratifiedKFold(n_splits=5, random_state=None, shuffle=False),
             error_score=nan,
             estimator=SVC(C=1.0, break_ties=False, cache_size=200,
                           class_weight='balanced', coef0=0.0,
                           decision_function_shape='ovr', degree=3,
                           gamma='scale', kernel='linear', max_iter=-1,
                           probability=False, random_state=42, shrinking=True,
                           tol=0.001, verbose=False),
             iid='deprecated', n_jobs=-1,
             param_grid={'C': [0.1, 1, 10, 1000, 10000, 100000]},
             pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
```

```
                   scoring='recall', verbose=1)
```

[69]:
```
grid_search.best_estimator_
```

[69]:
```
SVC(C=1000, break_ties=False, cache_size=200, class_weight='balanced',
    coef0=0.0, decision_function_shape='ovr', degree=3, gamma='scale',
    kernel='linear', max_iter=-1, probability=False, random_state=42,
    shrinking=True, tol=0.001, verbose=False)
```

[70]:
```
### Q.14 Using the best estimator to make predictions on the test set
```

[71]:
```
y_test_pred = grid_search.best_estimator_.predict(X_test_bow)
```

[72]:
```
print(classification_report(y_test, y_test_pred))
```

```
              precision    recall  f1-score   support

           0       0.96      0.88      0.92      1369
           1       0.32      0.58      0.41       131

    accuracy                           0.86      1500
   macro avg       0.64      0.73      0.66      1500
weighted avg       0.90      0.86      0.87      1500
```

[73]:
```
## Q 15 What are the most prominent terms in the toxic comments?
# Separate the comments from the test set that the model identified as toxic
# Make one large list of the terms
# Get the top 15 terms
```

[74]:
```
y_test_pred = grid_search.best_estimator_.predict(X_test_bow)
```

[75]:
```
toxic_comments = pd.Series(X_test)[y_test_pred == 1].values
```

[76]:
```
term_list = []
for comment in toxic_comments:
    term_list.extend(word_tokenize(comment))
```

[77]:
```
cts = Counter(term_list)
```

0.7 Step 7: Most Common Toxic Terms

[78]:
```
cts.most_common(15)
```

[78]:
```
[('fuck', 292),
 ('ass', 285),
 ('gay', 215),
 ('cuntbag', 126),
 ('fucking', 90),
 ('hate', 88),
 ('jews', 80),
 ('niggers', 79),
 ('spics', 79),
 ('minorities', 79),
 ('like', 25),
 ('go', 24),
 ('youre', 14),
 ('real', 11),
 ('well', 10)]
```

[79]:
```
### End of Program
```

[ ]:
```

```