quora-text-classification

Use the "Run" button to execute the code.

```
!ls
```

```
kaggle.json  sample_data
```

```
!pip install kaggle --upgrade
import os
#Kaggle_config_dir should be targeted to the folder where kaggle.jason is present
#using os.environ it is set folder(. means current folder)
os.environ['KAGGLE_CONFIG_DIR']='.'
!ls
!chmod 600 kaggle.json

!kaggle competitions download -c quora-insincere-questions-classification -f train.csv
!kaggle competitions download -c quora-insincere-questions-classification -f test.csv -
!kaggle competitions download -c quora-insincere-questions-classification -f sample_sub
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: kaggle in /usr/local/lib/python3.9/dist-packages (1.5.13)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.9/dist-packages (from kaggle) (8.0.1)
Requirement already satisfied: certifi in /usr/local/lib/python3.9/dist-packages (from kaggle) (2022.12.7)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.9/dist-packages (from kaggle) (2.8.2)
Requirement already satisfied: requests in /usr/local/lib/python3.9/dist-packages (from kaggle) (2.27.1)
Requirement already satisfied: tqdm in /usr/local/lib/python3.9/dist-packages (from kaggle) (4.65.0)
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.9/dist-packages (from kaggle) (1.16.0)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.9/dist-packages (from kaggle) (1.26.15)
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.9/dist-packages (from python-slugify->kaggle) (1.3)
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.9/dist-packages (from requests->kaggle) (2.0.12)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.9/dist-packages (from requests->kaggle) (3.4)
kaggle.json  sample_data
```

```
Downloading train.csv.zip to data
100% 54.9M/54.9M [00:00<00:00, 87.0MB/s]
100% 54.9M/54.9M [00:00<00:00, 81.9MB/s]
Downloading test.csv.zip to data
 76% 12.0M/15.8M [00:00<00:00, 60.9MB/s]
100% 15.8M/15.8M [00:00<00:00, 72.3MB/s]
Downloading sample_submission.csv.zip to data
100% 4.09M/4.09M [00:00<00:00, 40.0MB/s]
100% 4.09M/4.09M [00:00<00:00, 39.7MB/s]
```

#Explore the data using Pandas

```
train_fname='data/train.csv.zip'
test_fname='data/test.csv.zip'
sample_fname='data/sample_submission.csv.zip'
```

#Exploring data using Pandas

```
import pandas as pd
```

```
raw_df=pd.read_csv(train_fname)
```

```
raw_df
```

	qid	question_text	target
0	00002165364db923c7e6	How did Quebec nationalists see their province...	0
1	000032939017120e6e44	Do you have an adopted dog, how would you enco...	0
2	0000412ca6e4628ce2cf	Why does velocity affect time? Does velocity a...	0
3	000042bf85aa498cd78e	How did Otto von Guericke used the Magdeburg h...	0
4	0000455dfa3e01eae3af	Can I convert montra helicon D to a mountain b...	0
...	...	...	...
1306117	ffffcc4e2331aaf1e41e	What other technical skills do you need as a c...	0
1306118	ffffd431801e5a2f4861	Does MS in ECE have good job prospects in USA ...	0
1306119	ffffd48fb36b63db010c	Is foam insulation toxic?	0
1306120	ffffec519fa37cf60c78	How can one start a research project based on ...	0
1306121	ffffed09fedb5088744a	Who wins in a battle between a Wolverine and a...	0

1306122 rows × 3 columns

```
#assingning no ds where target ==1
sincear_ds=raw_df[raw_df.target==0]
```

```
sincear_ds.question_text.values[:10]
```

```
array(['How did Quebec nationalists see their province as a nation in the 1960s?',
       'Do you have an adopted dog, how would you encourage people to adopt and not
shop?',
       'Why does velocity affect time? Does velocity affect space geometry?',
       'How did Otto von Guericke used the Magdeburg hemispheres?',
       'Can I convert montra helicon D to a mountain bike by just changing the tyres?',
       'Is Gaza slowly becoming Auschwitz, Dachau or Treblinka for Palestinians?',
       'Why does Quora automatically ban conservative opinions when reported, but does
not do the same for liberal views?',
       'Is it crazy if I wash or wipe my groceries off? Germs are everywhere.',
       'Is there such a thing as dressing moderately, and if so, how is that different
than dressing modestly?',
       'Is it just me or have you ever been in this phase wherein you became ignorant
to the people you once loved, completely disregarding their feelings/lives so you get
to have something go your way and feel temporarily at ease. How did things change?'],
      dtype=object)
```

```
insincear_df=raw_df[raw_df.target==1]
```

```
insincear_df.question_text.values[:10]
```

```
array(['Has the United States become the largest dictatorship in the world?',
       'Which babies are more sweeter to their parents? Dark skin babies or light skin
babies?',
       "If blacks support school choice and mandatory sentencing for criminals why
don't they vote Republican?",
       'I am gay boy and I love my cousin (boy). He is sexy, but I dont know what to
do. He is hot, and I want to see his di**. What should I do?',
       'Which races have the smallest penis?',
       'Why do females find penises ugly?',
       'How do I marry an American woman for a Green Card? How much do they charge?',
       "Why do Europeans say they're the superior race, when in fact it took them over
2,000 years until mid 19th century to surpass China's largest economy?",
       'Did Julius Caesar bring a tyrannosaurus rex on his campaigns to frighten the
Celts into submission?',
       "In what manner has Republican backing of 'states rights' been hypocritical and
what ways have they actually restricted the ability of states to make their own
laws?"],
      dtype=object)
```

```
#to take count of target
#normalise==True to get percentage
raw_df.target.value_counts()
raw_df.target.value_counts(normalize=True)
```

```
0    0.93813
```

```
1    0.06187
Name: target, dtype: float64
```

```
raw_df['question_text'].apply(len).mean()
```

```
70.67883551459971
```

```
lists=["a","about","all","also","and","as","at","be","because","but","by","can","come",
```

```
%%time
dict1={}
for text in raw_df['question_text']:
  text.upper()
  for words in text.split():
    if words not in lists:
     if words in dict1.keys():
        dict1[words]+=1
     else:
       dict1[words]=1
```

```
CPU times: user 33.2 s, sys: 176 ms, total: 33.4 s
Wall time: 40.9 s
```

```
dict(sorted(dict1.items(),key=lambda x: x[1], reverse=True))
```

```
test_df=pd.read_csv(test_fname)
```

```
test_df[:5]
```

	qid	question_text
0	0000163e3ea7c7a74cd7	Why do so many women become so rude and arroga...
1	00002bd4fb5d505b9161	When should I apply for RV college of engineer...
2	00007756b4a147d2b0b3	What is it really like to be a nurse practitio...
3	000086e4b7e1c7146103	Who are entrepreneurs?
4	0000c4c3fbe8785a3090	Is education really making good people nowadays?

```
sub_df=pd.read_csv(sample_fname)
```

```
sub_df[sub_df.prediction==1]
```

qid	prediction

#Creating working model

```
SAMPLE_SIZE=100_000
```

```
sample_df=raw_df.sample(SAMPLE_SIZE,random_state=48)
```

```
sample_df
```

	qid	question_text	target
1186167	e8742311147e40e82cc5	What are the pros and cons, if bride's father ...	0
929790	b63729dd2633ca4e16d9	How many messages does it take from Quora Mode...	0
863523	a933de9b0432d3b2c610	How I can dress to attract girl?	0
594547	7472bb69ead4a6503c29	What challenges did Eric Carle face when becom...	0
682839	85bc2ad10bb74a9bb4bd	How do I make 10k per month using a mobile app?	0
...	...	...	...
1224468	effb2cc8ead21974a148	What is Garcinia?	0
331777	410a7d072b7923ac0213	What moment made you feel like time was standi...	0
781550	9919adff8cae97dab524	Now that medical doctor is no longer the "in" ...	1
285861	37fa008451b3e38e0a12	Why is electronics being used in automotive fu...	0
465285	5b1bd2afd765d914de14	What are some natural remedies for cramps at 5...	0

100000 rows × 3 columns

#Text Preprocessing Techniques

outline

1. inderstanding the bag of model
2. tokenization
3. stop word removal
4. stemming

Bag of word intuition

1. create a list of all word across all the text document
2. convert each question/document into vector count of each word

Limitation:

1. there maybe two many words, make vector large.
2. some words way occur many times.
3. some way occur rarely.
4. A single word may have many forms. past tense

```
q0=sincear_ds.question_text.values[1]
```

```
q0
```

```
'Do you have an adopted dog, how would you encourage people to adopt and not shop?'
```

```
q1=raw_df[raw_df.target==1].question_text.values[0]
```

```
q1
```

```
'Has the United States become the largest dictatorship in the world?'
```

###Tokenization

Spliting of documents into words and seperator.

```
from nltk.tokenize import word_tokenize
import nltk
```

```
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
```

```
True
```

```
q0_toke=word_tokenize(q0)
q1_toke=word_tokenize(q1)
```

```
q0_toke
```

```
['Do',
 'you',
 'have',
 'an',
 'adopted',
 'dog',
 ',',
 'how',
 'would',
 'you',
 'encourage',
 'people',
 'to',
 'adopt',
 'and',
 'not',
 'shop',
 '?']
```

```
q1_toke
```

```
['Has',
 'the',
 'United',
 'States',
 'become',
 'the',
 'largest',
 'dictatorship',
 'in',
 'the',
 'world',
 '?']
```

###Stop word removal

Removing commonly occuring words.

```
from nltk.corpus import stopwords
```

```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

```
True
```

```
stop_word=stopwords.words('english')
```

fn to remove stopword from question

```
def remove_stopwords(token):
  return{word for word in token if word.lower() not in stop_word}
```

```
q0_stp=remove_stopwords(q0_toke)
```

```
q0_stp
```

```
q1_stp=remove_stopwords(q1_toke)
```

```
q1_stp
```

###Stemming (stemmer)

go,gone,going -> go birds,bird->bird

```
from nltk.stem import PorterStemmer
```

```
stemmer=PorterStemmer()
```

```
stemmer.stem("birds")
```

```
'bird'
```

```
q0_stemmer=[stemmer.stem(words) for words in q0_stp]
```

```
q0_stemmer
```

```
[',', 'adopt', 'encourag', 'would', 'dog', '?', 'adopt', 'peopl', 'shop']
```

```
q1_stemmer=[stemmer.stem(words) for words in q1_stp]
```

```
q1_stemmer
```

```
['world', 'dictatorship', '?', 'becom', 'state', 'unit', 'largest']
```

```
q1_stp
```

```
{'?', 'States', 'United', 'become', 'dictatorship', 'largest', 'world'}
```

##Lematisation

"love->"love "loving"->"love" "lovable"-."love" #not much used in bag of method

```
from nltk.stem import WordNetLemmatizer
```

```
lemet=WordNetLemmatizer()
```

```
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
```

```
True
```

```
q0_lemet=[lemet.lemmatize(words) for words in q0_toke]
```

```
" ".join(q0_lemet)
```

```
'Do you have an adopted dog , how would you encourage people to adopt and not shop ?'
```

```
" ".join(q0_toke)
```

```
'Do you have an adopted dog , how would you encourage people to adopt and not shop ?'
```

##Implement bag of word Model

outline:

1. create a vocabulary using count vectorizer
2. Transform text to vector using count vectoriser
3. configure text preprocessing in count vectoriser

##Create a vocabulary

```
sample_df
```

	qid	question_text	target
1186167	e8742311147e40e82cc5	What are the pros and cons, if bride's father ...	0
929790	b63729dd2633ca4e16d9	How many messages does it take from Quora Mode...	0
863523	a933de9b0432d3b2c610	How I can dress to attract girl?	0
594547	7472bb69ead4a6503c29	What challenges did Eric Carle face when becom...	0
682839	85bc2ad10bb74a9bb4bd	How do I make 10k per month using a mobile app?	0
...	...	...	...
1224468	effb2cc8ead21974a148	What is Garcinia?	0
331777	410a7d072b7923ac0213	What moment made you feel like time was standi...	0
781550	9919adff8cae97dab524	Now that medical doctor is no longer the "in" ...	1
285861	37fa008451b3e38e0a12	Why is electronics being used in automotive fu...	0
465285	5b1bd2afd765d914de14	What are some natural remedies for cramps at 5...	0

100000 rows × 3 columns

```
#take 5 question from sample df
small_df=sample_df[:5]
```

```
small_df.question_text.values
```

```
array(["What are the pros and cons, if bride's father writes a will, giving equal share
to her daughter, instead of giving dowry esp in India?",
       'How many messages does it take from Quora Moderation to result in an edit
block?',
       'How I can dress to attract girl?',
       'What challenges did Eric Carle face when becoming a designer?',
       'How do I make 10k per month using a mobile app?'], dtype=object)
```

```
from sklearn.feature_extraction.text import CountVectorizer
```

```
small_vector=CountVectorizer()
```

```
small_vector.fit_transform(small_df.question_text)
```

```
<5x56 sparse matrix of type '<class 'numpy.int64'>'
	with 62 stored elements in Compressed Sparse Row format>
```

✋ This method is only to learn the words The small_df 5 question from sample questions which has 10k questions

```
small_vector.vocabulary_
```

```
print(len(small_vector.get_feature_names_out()))
print(small_vector.get_feature_names_out())
```

###Transform documents into vectors

1. .transform is used to transorm into vector
2.

```
vectors=small_vector.transform(small_df.question_text)
```

```
vectors
```

```
<5x56 sparse matrix of type '<class 'numpy.int64'>'
	with 62 stored elements in Compressed Sparse Row format>
```

```
#to see vecto we use .toarray()
print(vectors.shape)
vectors.toarray()
```

```
(5, 56)
```

```
array([[0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1,
        0, 1, 0, 1, 0, 0, 2, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0,
        1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1],
       [0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0,
        0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0,
        0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
        0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 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, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0,
        1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0],
       [1, 0, 0, 1, 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, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1,
        0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]])
```

```
def tokeniser(text):
  return{stemmer.stem(word) for word in word_tokenize(text)}
```

```
tokeniser(" this is not a big (deal)")
```

```
{'(', ')', 'a', 'big', 'deal', 'is', 'not', 'thi'}
```

Configure count vectoriser parameter

1. max_fearure=> it only give count vector of maximum of word. Here it is 1000
2.

```
vectoriser=CountVectorizer(lowercase=True,tokenizer=tokeniser,stop_words=stop_word,max_
```

```
sample_df
```

```
%%time
vectoriser.fit(sample_df.question_text)
```

```
#To print length of vocabulary, to print the vocabulary vocabulary_  is needed
len(vectoriser.vocabulary_)
```

```
1000
```

```
#to get words of vector
vectoriser.get_feature_names_out()[:100]
```

```
%%time
input=vectoriser.fit_transform(sample_df.question_text)
```

```
CPU times: user 41 s, sys: 137 ms, total: 41.1 s
Wall time: 42.4 s
```

```
sample_df.question_text.values[0]
```

```
"What are the pros and cons, if bride's father writes a will, giving equal share to her
daughter, instead of giving dowry esp in India?"
```

```
vectoriser.get_feature_names_out()[:100]
```

```
array(['!', '$', '%', '&', "'", "''", "'m", "'re", "'s", "'ve", '(', ')',
       ',', '-', '.', '1', '10', '100', '11', '12', '12th', '2', '20',
       '2017', '2018', '3', '30', '4', '5', '50', '6', '7', '8', ':', '?',
       '``', 'abl', 'abus', 'accept', 'access', 'accomplish', 'accord',
       'account', 'achiev', 'acid', 'act', 'action', 'activ', 'actor',
       'actual', 'ad', 'add', 'admiss', 'adult', 'advanc', 'advantag',
       'advic', 'affect', 'africa', 'african', 'age', 'ago', 'air',
       'allow', 'alon', 'alreadi', 'also', 'altern', 'alway', 'amazon',
       'america', 'american', 'among', 'amount', 'android', 'ani', 'anim',
       'anoth', 'answer', 'anxieti', 'anyon', 'anyth', 'apart', 'app',
       'appear', 'appl', 'appli', 'applic', 'arab', 'area', 'armi',
       'around', 'art', 'asian', 'ask', 'atheist', 'attack', 'attend',
       'attract', 'australia'], dtype=object)
```

```
input.shape
```

```
(100000, 1000)
```

```
input[0].toarray()
```

```
test_df
```

```
%%time
test_input=vectoriser.fit_transform(test_df.question_text)
```

```
test_input.shape
```

```
(375806, 1000)
```

ML model for text classification

Outline:

- Create a training and validation set(train model with training set and validate it with validation set)
- Train a logistic regression model
- Make prediction on training, validation & test data

Split into Trainning and Validation set

```
sample_df
```

Train_split_text in sklern to split into train and validation

```
from sklearn.model_selection import train_test_split
```

```
#0.3 is thr ratio of validaton set to sample 30% used for validation other 70% training
train_inputs,val_inputs,train_targets,val_targets=train_test_split(input,sample_df.targ
```

```
print("train_input  ",train_inputs.shape)
print("train_target ",train_targets.shape)
print("val_inputs  ",val_inputs.shape)
print("val_target ",val_targets.shape)
```

```
train_input   (70000, 1000)
train_target  (70000,)
val_inputs   (30000, 1000)
val_target  (30000,)
```

Train Logistic Regression Models

1.

```
from sklearn.linear_model import LogisticRegression
```

```
max_itteration=1000
```

```
#max_ier is used to give information how many level of itteration for model, how many t
#solver used tell regresson to use which method for regression ("sag"=stocatic decent g
model=LogisticRegression(max_iter=max_itteration,solver="sag")
```

```
%%time
model.fit(train_inputs,train_targets)
```

```
CPU times: user 18.3 s, sys: 25.5 ms, total: 18.3 s
Wall time: 18.3 s
```

```
    LogisticRegression(max_iter=1000, solver='sag')
```

**In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.**
LogisticRegression

```
    LogisticRegression(max_iter=1000, solver='sag')
```

```
train_pred=model.predict(train_inputs)
```

```
train_pred.shape
```

```
(70000,)
```

```
#pd.series print the number of ellements in prediction
pd.Series(train_pred).value_counts()
```

```
0    67925
1     2075
dtype: int64
```

```
pd.Series(train_targets).value_counts()
```

```
0    65716
1     4284
Name: target, dtype: int64
```

```
#how to check accuracy ?
from sklearn.metrics import accuracy_score
```

```
accuracy_score(train_targets,train_pred)
```

```
0.9502714285714285
```

```
import numpy as np
```

```
accuracy_score(train_targets,np.zeros(len(train_targets)))
```

```
0.9388
```

it is showing a 94% accuracy when we compare train tARGet with full zero numpy. and our prediction is 95% accuracy so it cannot be called as a great model. so to calculate this sklearn has another class called f1_score from sklearn.metrics

```
from sklearn.metrics import f1_score
```

```
f1_score(train_targets,train_pred)
```

```
0.4525868847303035
```

```
f1_score(train_targets,np.random.choice((1,0),len(train_targets)))
```

```
0.10799004419159854
```

```
#orediction ov validation files which is not used for bag of word and logistic regressi
val_pred=model.predict(val_inputs)
```

```
f1_score(val_targets,val_pred)
```

```
0.40951694304253783
```

###Make prediction and upload to Kaggle

```
test_df
```

	qid	question_text
0	0000163e3ea7c7a74cd7	Why do so many women become so rude and arroga...
1	00002bd4fb5d505b9161	When should I apply for RV college of engineer...

	qid	question_text
2	00007756b4a147d2b0b3	What is it really like to be a nurse practitio...
3	000086e4b7e1c7146103	Who are entrepreneurs?
4	0000c4c3fbe8785a3090	Is education really making good people nowadays?
...	...	...
375801	ffff7fa746bd6d6197a9	How many countries listed in gold import in in...
375802	ffffa1be31c43046ab6b	Is there an alternative to dresses on formal p...
375803	ffffae173b6ca6bfa563	Where I can find best friendship quotes in Tel...
375804	ffffb1f7f1a008620287	What are the causes of refraction of light?
375805	fffff85473f4699474b0	Climate change is a worrying topic. How much t...

375806 rows × 2 columns

```
test_input
```

```
<375806x1000 sparse matrix of type '<class 'numpy.int64'>'
	with 2089244 stored elements in Compressed Sparse Row format>
```

```
test_pred=model.predict(test_input)
```

```
sub_df
```

	qid	prediction
0	0000163e3ea7c7a74cd7	0
1	00002bd4fb5d505b9161	1
2	00007756b4a147d2b0b3	0
3	000086e4b7e1c7146103	0
4	0000c4c3fbe8785a3090	0
...	...	...
375801	ffff7fa746bd6d6197a9	0
375802	ffffa1be31c43046ab6b	0
375803	ffffae173b6ca6bfa563	0
375804	ffffb1f7f1a008620287	0
375805	fffff85473f4699474b0	1

375806 rows × 2 columns

```
#submission (sub_df) had everything value as 0 then we equated it with prediction so th
sub_df.prediction=test_pred
```

```
sub_df.to_csv('submission.csv',index=None)
```