Quora-1.png

# Quora Question Pairs

# 1. Business Problem

## 1.1 Description

Quora is a place to gain and share knowledge—about anything. It's a platform to ask questions and connect with answers. This empowers people to learn from each other and to better understand the world.

Over 100 million people visit Quora every month, so it's no surprise that many people ask similarly worded quest cause seekers to spend more time finding the best answer to their question, and make writers feel they need to Quora values canonical questions because they provide a better experience to active seekers and writers, and of term.

> Credits: Kaggle

__ Problem Statement __

- Identify which questions asked on Quora are duplicates of questions that have already been asked.
- This could be useful to instantly provide answers to questions that have already been answered.
- We are tasked with predicting whether a pair of questions are duplicates or not.

## 1.2 Sources/Useful Links

- Source : https://www.kaggle.com/c/quora-question-pairs

  **__ Useful Links __**
- Discussions : https://www.kaggle.com/anokas/data-analysis-xgboost-starter-0-35460-lb/comments
- Kaggle Winning Solution and other approaches: https://www.dropbox.com/sh/93968nfnrzh8bp5/AACZdt
- Blog 1 : https://engineering.quora.com/Semantic-Question-Matching-with-Deep-Learning
- Blog 2 : https://towardsdatascience.com/identifying-duplicate-questions-on-quora-top-12-on-kaggle-4c1c

## 1.3 Real world/Business Objectives and Constraints

1. The cost of a mis-classification can be very high.
2. You would want a probability of a pair of questions to be duplicates so that you can choose any threshold
3. No strict latency concerns.
4. Interpretability is partially important.

```
from google.colab import drive
```

```
from google.colab import drive
drive.mount('/content/drive')
%cd ./drive/My Drive
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.m
[Errno 2] No such file or directory: './drive/My Drive'
/content/drive/My Drive
```

# 2. Machine Learning Probelm

## 2.1 Data

### 2.1.1 Data Overview

- Data will be in a file Train.csv
- Train.csv contains 5 columns : qid1, qid2, question1, question2, is_duplicate
- Size of Train.csv - 60MB
- Number of rows in Train.csv = 404,290

### 2.1.2 Example Data point

```
"id","qid1","qid2","question1","question2","is_duplicate"
"0","1","2","What is the step by step guide to invest in share market in india?","What
in share market?","0"
"1","3","4","What is the story of Kohinoor (Koh-i-Noor) Diamond?","What would happen i
Kohinoor (Koh-i-Noor) diamond back?","0"
"7","15","16","How can I be a good geologist?","What should I do to be a great geologi
"11","23","24","How do I read and find my YouTube comments?","How can I see all my You
```

## 2.2 Mapping the real world problem to an ML problem

### 2.2.1 Type of Machine Leaning Problem

It is a binary classification problem, for a given pair of questions we need to predict if they are duplicate or not.

### 2.2.2 Performance Metric

Source: https://www.kaggle.com/c/quora-question-pairs#evaluation

Metric(s):

- log-loss : https://www.kaggle.com/wiki/LogarithmicLoss
- Binary Confusion Matrix

# Reading the data

```
!pip3 install fuzzywuzzy
!pip3 install distance
!pip3 install spacy
import numpy as np
import pandas as pd
from pandas import DataFrame, Series
import seaborn as sns
import matplotlib.pyplot as plt
from subprocess import check_output
%matplotlib inline
import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.tools as tls
import os
import gc

import re
from nltk.corpus import stopwords
import distance
from nltk.stem import PorterStemmer
from bs4 import BeautifulSoup

import warnings
warnings.filterwarnings("ignore")
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from subprocess import check_output
%matplotlib inline
import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.tools as tls
import os
import gc

import pandas as pd
import matplotlib.pyplot as plt
import re
import time
import warnings
import sqlite3
from sqlalchemy import create_engine # database connection
import csv
import os
warnings.filterwarnings("ignore")
import datetime as dt
import numpy as np
from nltk.corpus import stopwords
```

```
from nltk.corpus import stopwords
from sklearn.decomposition import TruncatedSVD
from sklearn.preprocessing import normalize
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.manifold import TSNE
import seaborn as sns
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix
from sklearn.metrics.classification import accuracy_score, log_loss
from sklearn.feature_extraction.text import TfidfVectorizer
from collections import Counter
from scipy.sparse import hstack
from sklearn.multiclass import OneVsRestClassifier
from sklearn.svm import SVC
#from sklearn.cross_validation import StratifiedKFold
from collections import Counter, defaultdict
from sklearn.calibration import CalibratedClassifierCV
from sklearn.naive_bayes import MultinomialNB
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV
import math
from sklearn.metrics import normalized_mutual_info_score
from sklearn.ensemble import RandomForestClassifier



from sklearn.model_selection import cross_val_score
from sklearn.linear_model import SGDClassifier
from mlxtend.classifier import StackingClassifier

from sklearn import model_selection
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import precision_recall_curve, auc, roc_curve

from fuzzywuzzy import fuzz
from sklearn.manifold import TSNE
# Import the Required lib packages for WORD-Cloud generation
# https://stackoverflow.com/questions/45625434/how-to-install-wordcloud-in-python3-6
from wordcloud import WordCloud, STOPWORDS
from os import path
from PIL import Image

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

```
Requirement already satisfied: fuzzywuzzy in /usr/local/lib/python3.6/dist-packages (
Requirement already satisfied: distance in /usr/local/lib/python3.6/dist-packages (0.
Requirement already satisfied: spacy in /usr/local/lib/python3.6/dist-packages (2.1.8
Requirement already satisfied: wasabi<1.1.0,>=0.2.0 in /usr/local/lib/python3.6/dist-
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.6/
Requirement already satisfied: preshed<2.1.0,>=2.0.1 in /usr/local/lib/python3.6/dist
Requirement already satisfied: blis<0.3.0,>=0.2.2 in /usr/local/lib/python3.6/dist-pa
Requirement already satisfied: srsly<1.1.0,>=0.0.6 in /usr/local/lib/python3.6/dist-p
Requirement already satisfied: plac<1.0.0,>=0.9.6 in /usr/local/lib/python3.6/dist-pa
Requirement already satisfied: thinc<7.1.0,>=7.0.8 in /usr/local/lib/python3.6/dist-p
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.6/dist-p
Requirement already satisfied: numpy>=1.15.0 in /usr/local/lib/python3.6/dist-package
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.6/di
Requirement already satisfied: tqdm<5.0.0,>=4.10.0 in /usr/local/lib/python3.6/dist-p
Requirement already satisfied: idna<2.9,>=2.5 in /usr/local/lib/python3.6/dist-packag
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.6/dist-pa
Requirement already satisfied: urllib3<1.25,>=1.21.1 in /usr/local/lib/python3.6/dist
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /usr/local/lib/python3.6/dist
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
True
```

```
# This function plots the confusion matrices given y_i, y_i_hat.
def plot_confusion_matrix(test_y, predict_y):
    C = confusion_matrix(test_y, predict_y)
    # C = 9,9 matrix, each cell (i,j) represents number of points of class i are predicted

    A =(((C.T)/(C.sum(axis=1))).T)
    #divid each element of the confusion matrix with the sum of elements in that column

    # C = [[1, 2],
    #     [3, 4]]
    # C.T = [[1, 3],
    #        [2, 4]]
    # C.sum(axis = 1)  axis=0 corresonds to columns and axis=1 corresponds to rows in two
    # C.sum(axix =1) = [[3, 7]]
    # ((C.T)/(C.sum(axis=1))) = [[1/3, 3/7]
    #                            [2/3, 4/7]]

    # ((C.T)/(C.sum(axis=1))).T = [[1/3, 2/3]
    #                            [3/7, 4/7]]
    # sum of row elements = 1

    B =(C/C.sum(axis=0))
    #divid each element of the confusion matrix with the sum of elements in that row
    # C = [[1, 2],
    #     [3, 4]]
    # C.sum(axis = 0)  axis=0 corresonds to columns and axis=1 corresponds to rows in two
    # C.sum(axix =0) = [[4, 6]]
    # (C/C.sum(axis=0)) = [[1/4, 2/6],
    #                       [3/4, 4/6]]
    plt.figure(figsize=(20,4))

    labels = [1,2]
    # representing A in heatmap format
    cmap=sns.light_palette("blue")
```

```
    cmap=sns.light_palette("blue")
    plt.subplot(1, 3, 1)
    sns.heatmap(C, annot=True, cmap=cmap, fmt=".3f", xticklabels=labels, yticklabels=label
    plt.xlabel('Predicted Class')
    plt.ylabel('Original Class')
    plt.title("Confusion matrix")

    plt.subplot(1, 3, 2)
    sns.heatmap(B, annot=True, cmap=cmap, fmt=".3f", xticklabels=labels, yticklabels=label
    plt.xlabel('Predicted Class')
    plt.ylabel('Original Class')
    plt.title("Precision matrix")

    plt.subplot(1, 3, 3)
    # representing B in heatmap format
    sns.heatmap(A, annot=True, cmap=cmap, fmt=".3f", xticklabels=labels, yticklabels=label
    plt.xlabel('Predicted Class')
    plt.ylabel('Original Class')
    plt.title("Recall matrix")

    plt.show()
```

```
data = pd.read_csv("train.csv")
```

```
data=data[0:50000:5]
```

```
data.head()
```

| | id | qid1 | qid2 | question1 | |
|---|---|---|---|---|---|
| **0** | 0 | 1 | 2 | What is the step by step guide to invest in sh... | What is the step by step |
| **5** | 5 | 11 | 12 | Astrology: I am a Capricorn Sun Cap moon and c... | I'm a triple Capricorn (Sun |
| **10** | 10 | 21 | 22 | Method to find separation of slits using fresn... | What are some of the thir |
| **15** | 15 | 31 | 32 | What would a Trump presidency mean for current... | How will a Trump preside |
| **20** | 20 | 41 | 42 | Why do rockets look white? | Why are rockets and b |

```
data.shape[0],data.shape[1]
```

```
(10000, 6)
```

## 2.3 Train and Test Construction

We build train and test by randomly splitting in the ratio of 70:30 or 80:20 whatever we choose as we have suffic

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

```
df_train,df_test=train_test_split(data,test_size=0.25)
```

```
print(df_train.shape[0])
print(df_test.shape[0])
```

```
7500
2500
```

```
df_train.head()
```

| | id | qid1 | qid2 | question1 | |
|---|---|---|---|---|---|
| **6075** | 6075 | 11913 | 11914 | What is Artificial Intelligence? | What all doe |
| **1205** | 1205 | 2402 | 2403 | Which processor is faster and better for batte... | Which one is a be |
| **16030** | 16030 | 30586 | 25457 | What should be the most important thing in you... | Life Advice: What a |
| **37040** | 37040 | 67461 | 67462 | What are the largest veins and arteries in the... | What are the maj |
| **44110** | 44110 | 79241 | 79242 | How do bladeless fans work? | H |

```
#Checking whether there are any rows with null values
nan_rows = df_train[df_train.isnull().any(1)]
print (nan_rows)
# Filling the null values with ' '
df_train = df_train.fillna('')
nan_rows = df_train[df_train.isnull().any(1)]
print (nan_rows)
```

```
Empty DataFrame
Columns: [id, qid1, qid2, question1, question2, is_duplicate]
Index: []
Empty DataFrame
Columns: [id, qid1, qid2, question1, question2, is_duplicate]
Index: []
```

```
#Test
#Checking whether there are any rows with null values
nan_rows = df_test[df_test.isnull().any(1)]
print (nan_rows)
# Filling the null values with ' '
df_test = df_test.fillna('')
nan_rows = df_test[df_test.isnull().any(1)]
print (nan_rows)
```

```
Empty DataFrame
Columns: [id, qid1, qid2, question1, question2, is_duplicate]
Index: []
Empty DataFrame
Columns: [id, qid1, qid2, question1, question2, is_duplicate]
Index: []
```

## 3.3 Basic Feature Extraction (before cleaning)

Let us now construct a few features like:

- **freq_qid1** = Frequency of qid1's
- **freq_qid2** = Frequency of qid2's
- **q1len** = Length of q1
- **q2len** = Length of q2
- **q1_n_words** = Number of words in Question 1
- **q2_n_words** = Number of words in Question 2
- **word_Common** = (Number of common unique words in Question 1 and Question 2)
- **word_Total** =(Total num of words in Question 1 + Total num of words in Question 2)
- **word_share** = (word_common)/(word_Total)
- **freq_q1+freq_q2** = sum total of frequency of qid1 and qid2
- **freq_q1-freq_q2** = absolute difference of frequency of qid1 and qid2

## 3.4 Preprocessing of Text

- Preprocessing:
  - Removing html tags
  - Removing Punctuations
  - Performing stemming
  - Removing Stopwords
  - Expanding contractions etc.

```
# To get the results in 4 decemal points
SAFE_DIV = 0.0001

STOP_WORDS = stopwords.words("english")


def preprocess(x):
    x = str(x).lower()
    x = x.replace(",000,000", "m").replace(",000", "k").replace("′", "'").replace("’", "'"
                           .replace("won't", "will not").replace("cannot", "can not").repl
                           .replace("n't", " not").replace("what's", "what is").replace("i
                           .replace("'ve", " have").replace("i'm", "i am").replace("'re",
                           .replace("he's", "he is").replace("she's", "she is").replace("'
                           .replace("%", " percent ").replace("₹", " rupee ").replace("$",
                           .replace("€", " euro ").replace("'ll", " will")
    x = re.sub(r"([0-9]+)000000", r"\1m", x)
    x = re.sub(r"([0-9]+)000", r"\1k", x)


    porter = PorterStemmer()
    pattern = re.compile('\W')

    if type(x) == type(''):
        x = re.sub(pattern, ' ', x)


    if type(x) == type(''):
        x = porter.stem(x)
```

```
        x = porter.stem(x)
        example1 = BeautifulSoup(x)
        x = example1.get_text()


    return x
```

## 3.5 Advanced Feature Extraction (NLP and Fuzzy Features)

Definition:

- **Token**: You get a token by splitting sentence a space
- **Stop_Word** : stop words as per NLTK.
- **Word** : A token that is not a stop_word

Features:

- **cwc_min** : Ratio of common_word_count to min lenghth of word count of Q1 and Q2
  cwc_min = common_word_count / (min(len(q1_words), len(q2_words))

- **cwc_max** : Ratio of common_word_count to max lenghth of word count of Q1 and Q2
  cwc_max = common_word_count / (max(len(q1_words), len(q2_words))

- **csc_min** : Ratio of common_stop_count to min lenghth of stop count of Q1 and Q2
  csc_min = common_stop_count / (min(len(q1_stops), len(q2_stops))

- **csc_max** : Ratio of common_stop_count to max lenghth of stop count of Q1 and Q2
  csc_max = common_stop_count / (max(len(q1_stops), len(q2_stops))

- **ctc_min** : Ratio of common_token_count to min lenghth of token count of Q1 and Q2
  ctc_min = common_token_count / (min(len(q1_tokens), len(q2_tokens))

- **ctc_max** : Ratio of common_token_count to max lenghth of token count of Q1 and Q2
  ctc_max = common_token_count / (max(len(q1_tokens), len(q2_tokens))

- **last_word_eq** : Check if First word of both questions is equal or not
  last_word_eq = int(q1_tokens[-1] == q2_tokens[-1])

- **first_word_eq** : Check if First word of both questions is equal or not
  first_word_eq = int(q1_tokens[0] == q2_tokens[0])

- **abs_len_diff** : Abs. length difference
  abs_len_diff = abs(len(q1_tokens) - len(q2_tokens))

- **mean_len** : Average Token Length of both Questions
  mean_len = (len(q1_tokens) + len(q2_tokens))/2

- **fuzz_ratio** : https://github.com/seatgeek/fuzzywuzzy#usage http://chairnerd.seatgeek.com/fuzzywuzzy-f

- **fuzz_partial_ratio** : https://github.com/seatgeek/fuzzywuzzy#usage http://chairnerd.seatgeek.com/fuzzy

- **token_sort_ratio** : https://github.com/seatgeek/fuzzywuzzy#usage http://chairnerd.seatgeek.com/fuzzyw

- **token_set_ratio** : https://github.com/seatgeek/fuzzywuzzy#usage http://chairnerd.seatgeek.com/fuzzyw

- **longest_substr_ratio** : Ratio of length longest common substring to min lenghth of token count of Q1 and
  longest_substr_ratio = len(longest common substring) / (min(len(q1_tokens), len(q2_tokens))

```
def get_token_features(q1, q2):
    token_features = [0.0]*10

    # Converting the Sentence into Tokens:
    q1_tokens = q1.split()
    q2_tokens = q2.split()

    if len(q1_tokens) == 0 or len(q2_tokens) == 0:
        return token_features
    # Get the non-stopwords in Questions
    q1_words = set([word for word in q1_tokens if word not in STOP_WORDS])
    q2_words = set([word for word in q2_tokens if word not in STOP_WORDS])

    #Get the stopwords in Questions
    q1_stops = set([word for word in q1_tokens if word in STOP_WORDS])
    q2_stops = set([word for word in q2_tokens if word in STOP_WORDS])

    # Get the common non-stopwords from Question pair
    common_word_count = len(q1_words.intersection(q2_words))

    # Get the common stopwords from Question pair
    common_stop_count = len(q1_stops.intersection(q2_stops))

    # Get the common Tokens from Question pair
    common_token_count = len(set(q1_tokens).intersection(set(q2_tokens)))
```

```
    token_features[0] = common_word_count / (min(len(q1_words), len(q2_words)) + SAFE_DIV)
    token_features[1] = common_word_count / (max(len(q1_words), len(q2_words)) + SAFE_DIV)
    token_features[2] = common_stop_count / (min(len(q1_stops), len(q2_stops)) + SAFE_DIV)
    token_features[3] = common_stop_count / (max(len(q1_stops), len(q2_stops)) + SAFE_DIV)
    token_features[4] = common_token_count / (min(len(q1_tokens), len(q2_tokens)) + SAFE_D
    token_features[5] = common_token_count / (max(len(q1_tokens), len(q2_tokens)) + SAFE_D

    # Last word of both question is same or not
    token_features[6] = int(q1_tokens[-1] == q2_tokens[-1])

    # First word of both question is same or not
    token_features[7] = int(q1_tokens[0] == q2_tokens[0])

    token_features[8] = abs(len(q1_tokens) - len(q2_tokens))

    #Average Token Length of both Questions
    token_features[9] = (len(q1_tokens) + len(q2_tokens))/2
    return token_features

# get the Longest Common sub string

def get_longest_substr_ratio(a, b):
    strs = list(distance.lcsubstrings(a, b))
    if len(strs) == 0:
        return 0
    else:
        return len(strs[0]) / (min(len(a), len(b)) + 1)

def extract_features(df):
    df['freq_qid1'] = df.groupby('qid1')['qid1'].transform('count')
    df['freq_qid2'] = df.groupby('qid2')['qid2'].transform('count')
    df['q1len'] = df['question1'].str.len()
    df['q2len'] = df['question2'].str.len()
    df['q1_n_words'] = df['question1'].apply(lambda row: len(row.split(" ")))
    df['q2_n_words'] = df['question2'].apply(lambda row: len(row.split(" ")))

    def normalized_word_Common(row):
        w1 = set(map(lambda word: word.lower().strip(), row['question1'].split(" ")))
        w2 = set(map(lambda word: word.lower().strip(), row['question2'].split(" ")))
        return 1.0 * len(w1 & w2)
    df['word_Common'] = df.apply(normalized_word_Common, axis=1)

    def normalized_word_Total(row):
        w1 = set(map(lambda word: word.lower().strip(), row['question1'].split(" ")))
        w2 = set(map(lambda word: word.lower().strip(), row['question2'].split(" ")))
        return 1.0 * (len(w1) + len(w2))
    df['word_Total'] = df.apply(normalized_word_Total, axis=1)

    def normalized_word_share(row):
        w1 = set(map(lambda word: word.lower().strip(), row['question1'].split(" ")))
        w2 = set(map(lambda word: word.lower().strip(), row['question2'].split(" ")))
        return 1.0 * len(w1 & w2)/(len(w1) + len(w2))
    df['word_share'] = df.apply(normalized_word_share, axis=1)
```

```
    df['freq_q1+q2'] = df['freq_qid1']+df['freq_qid2']
    df['freq_q1-q2'] = abs(df['freq_qid1']-df['freq_qid2'])



    # preprocessing each question
    df["question1"] = df["question1"].fillna("").apply(preprocess)
    df["question2"] = df["question2"].fillna("").apply(preprocess)

    print("token features...")

    # Merging Features with dataset

    token_features = df.apply(lambda x: get_token_features(x["question1"], x["question2"])

    df["cwc_min"]       = list(map(lambda x: x[0], token_features))
    df["cwc_max"]       = list(map(lambda x: x[1], token_features))
    df["csc_min"]       = list(map(lambda x: x[2], token_features))
    df["csc_max"]       = list(map(lambda x: x[3], token_features))
    df["ctc_min"]       = list(map(lambda x: x[4], token_features))
    df["ctc_max"]       = list(map(lambda x: x[5], token_features))
    df["last_word_eq"]  = list(map(lambda x: x[6], token_features))
    df["first_word_eq"] = list(map(lambda x: x[7], token_features))
    df["abs_len_diff"]  = list(map(lambda x: x[8], token_features))
    df["mean_len"]      = list(map(lambda x: x[9], token_features))

    #Computing Fuzzy Features and Merging with Dataset

    # do read this blog: http://chairnerd.seatgeek.com/fuzzywuzzy-fuzzy-string-matching-in
    # https://stackoverflow.com/questions/31806695/when-to-use-which-fuzz-function-to-comp
    # https://github.com/seatgeek/fuzzywuzzy
    print("fuzzy features..")

    df["token_set_ratio"]       = df.apply(lambda x: fuzz.token_set_ratio(x["question1"],
    # The token sort approach involves tokenizing the string in question, sorting the toke
    # then joining them back into a string We then compare the transformed strings with a
    df["token_sort_ratio"]      = df.apply(lambda x: fuzz.token_sort_ratio(x["question1"],
    df["fuzz_ratio"]            = df.apply(lambda x: fuzz.QRatio(x["question1"], x["questi
    df["fuzz_partial_ratio"]    = df.apply(lambda x: fuzz.partial_ratio(x["question1"], x[
    df["longest_substr_ratio"]  = df.apply(lambda x: get_longest_substr_ratio(x["question1
    return df


df_train_afe = extract_features(df_train)
df_test_afe=extract_features(df_test)
```

```
token features...
fuzzy features..
token features...
fuzzy features..
```

```
df_train_afe.shape
```

```
(7500, 32)
```

## 3.6 Featurizing text data with tfidf word-vectors

```
df_train_afe['question1'] = df_train_afe['question1']
df_train_afe['question2'] = df_train_afe['question2']
df_test_afe['question1'] = df_test_afe['question1']
df_test_afe['question2'] = df_test_afe['question2']
```

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import CountVectorizer
# merge texts
questions_train = list(df_train_afe['question1']+df_train_afe['question2'])

tfidf = TfidfVectorizer(lowercase=False )
tfidf.fit(questions_train)

word2tfidf = dict(zip(tfidf.get_feature_names(), tfidf.idf_))
```

```
df_train_vec=DataFrame()
df_test_vec=DataFrame()
```

```
# en_vectors_web_lg, which includes over 1 million unique vectors.
import spacy
nlp=spacy.load('en_core_web_sm')
from spacy.lang.en import English
from tqdm import tqdm

vecs1 = []
# https://github.com/noamraph/tqdm
# tqdm is used to print the progress bar
for qu1 in tqdm(list(df_train_afe['question1'])):
    doc1 = nlp(qu1)
    # 384 is the number of dimensions of vectors
    mean_vec1 = np.zeros([len(doc1), len(doc1[0].vector)])
    for word1 in doc1:
        # word2vec
        vec1 = word1.vector
        # fetch df score
        try:
            idf = word2tfidf[str(word1)]
        except:
            idf = 0
        # compute final vec
        mean_vec1 += vec1 * idf
    mean_vec1 = mean_vec1.mean(axis=0)
    vecs1.append(mean_vec1)
df_train_vec['feats_1'] = list(vecs1)
```

```
vecs1 = []
# https://github.com/noamraph/tqdm
# tqdm is used to print the progress bar
for qu1 in tqdm(list(df_train_afe['question2'])):
    doc1 = nlp(qu1)
    # 384 is the number of dimensions of vectors
    mean_vec1 = np.zeros([len(doc1), len(doc1[0].vector)])
    for word1 in doc1:
        # word2vec
        vec1 = word1.vector
        # fetch df score
        try:
            idf = word2tfidf[str(word1)]
        except:
            idf = 0
        # compute final vec
        mean_vec1 += vec1 * idf
    mean_vec1 = mean_vec1.mean(axis=0)
    vecs1.append(mean_vec1)
df_train_vec['feats_2'] = list(vecs1)
```

```
df_train_vec.head
```

```
df_train_ave = pd.DataFrame(df_train_vec.feats_1.values.tolist())
df_train_ave2 = pd.DataFrame(df_train_vec.feats_2.values.tolist())
```

```
df_train_ave2.shape
```

```
(7500, 96)
```

```
# en_vectors_web_lg, which includes over 1 million unique vectors.
import spacy
nlp=spacy.load('en_core_web_sm')
from spacy.lang.en import English
from tqdm import tqdm

vecs1 = []
# https://github.com/noamraph/tqdm
# tqdm is used to print the progress bar
for qu1 in tqdm(list(df_test_afe['question1'])):
    doc1 = nlp(qu1)
    # 384 is the number of dimensions of vectors
    mean_vec1 = np.zeros([len(doc1), len(doc1[0].vector)])
    for word1 in doc1:
        # word2vec
        vec1 = word1.vector
        # fetch df score
        try:
            idf = word2tfidf[str(word1)]
        except:
```

```
            idf = 0
        # compute final vec
        mean_vec1 += vec1 * idf
    mean_vec1 = mean_vec1.mean(axis=0)
    vecs1.append(mean_vec1)
df_test_vec['feats_1'] = list(vecs1)
```

```
# en_vectors_web_lg, which includes over 1 million unique vectors.
import spacy
nlp=spacy.load('en_core_web_sm')
from spacy.lang.en import English
from tqdm import tqdm

vecs1 = []
# https://github.com/noamraph/tqdm
# tqdm is used to print the progress bar
for qu1 in tqdm(list(df_test_afe['question2'])):
    doc1 = nlp(qu1)
    # 384 is the number of dimensions of vectors
    mean_vec1 = np.zeros([len(doc1), len(doc1[0].vector)])
    for word1 in doc1:
        # word2vec
        vec1 = word1.vector
        # fetch df score
        try:
            idf = word2tfidf[str(word1)]
        except:
            idf = 0
        # compute final vec
        mean_vec1 += vec1 * idf
    mean_vec1 = mean_vec1.mean(axis=0)
    vecs1.append(mean_vec1)
df_test_vec['feats_2'] = list(vecs1)
```

```
df_test_ave = pd.DataFrame(df_test_vec.feats_1.values.tolist())
df_test_ave2 = pd.DataFrame(df_test_vec.feats_2.values.tolist())
```

```
df_test_ave.head
```

```
<bound method NDFrame.head of                0           1           2  ...
0      55.676373  -69.293416 -100.540564  ...   41.045996  -91.459483   69.550669
1      -7.907188   -6.214972    3.941869  ...    0.504844   -6.143659   -5.250861
2      35.881435  -42.216787  -75.993772  ...  -16.962134  -45.838620  -30.204897
3      72.407315  -23.723768    2.321285  ...   19.039772  -35.330183   16.345512
4      36.982321  -40.711484   -5.168114  ...  -15.637145    2.201329   21.626318
5      22.250418  -16.032876    2.767762  ...   -2.982986  -27.121021   22.771539
6      62.354329   -5.084054  -36.258385  ...   42.733077   -5.455346   33.438268
7       8.620701   15.300774   -2.220484  ...    7.660635  -44.645203  -31.399163
8     104.106118  -40.426530  -33.817928  ...   34.909616  -96.405473   95.134950
9     141.026199 -205.958752 -167.517763  ...  -11.290497 -198.038393  195.080426
10     10.660709   39.323036  -68.805159  ...  -11.851143  -19.750072    7.464219
11     38.843623  -93.889069 -135.287322  ...   44.508487  -93.576323   70.656907
12     -4.834304  -85.792970  -55.040799  ...    4.701767    5.912030  -24.194110
13     18.683698  -51.256946 -228.841520  ...  135.629103   -5.626777  -60.612180
14     49.599904  -40.392448  -75.525421  ...   37.488694  -38.629653   20.849172
15     37.340766   -3.054777  -18.891218  ...   26.184507  -36.947177   -1.605292
16     78.022017  -63.706884  -33.306264  ...   -9.073442  -51.721654   27.806188
17     68.766195  -47.223478 -113.362426  ...  -11.724828  -34.219909   41.850717
18     90.220120    6.973669  -10.451894  ...   31.038971  -70.810240   88.520393
19     95.346914   14.732527   -2.255951  ...   72.048934  -60.521912   31.041605
20     82.390392  -79.445982  -29.916504  ...  -45.100557 -107.764785  122.015432
21     90.649263  -21.952502  -50.155744  ...    3.753665  -17.144196   97.253853
22     64.767079    5.663171  -35.545964  ...   37.105841  -54.664683    9.588223
23    117.807390  -68.328536   65.950119  ...  -34.531788  -65.135938   60.546552
24    110.513616   -9.600875  -74.257225  ...   10.070497  -41.901476    4.042290
25     31.310685  -72.617664   30.649680  ...   34.221035  -76.649137   24.679297
26     42.263461   -3.958726  -12.539535  ...    4.773146   -2.250153   37.986240
27    125.272094 -126.115292    1.011468  ...   40.685774 -120.146563  214.729094
28     75.065707   -0.004583  -34.447115  ...   15.612866  -13.876423  -68.387914
29     34.622145    4.981720  -56.687831  ...   -8.392697    0.364838   31.156752
...          ...         ...         ...  ...         ...         ...         ...
2470   14.799084  -58.328190 -190.735688  ...  113.437533 -177.336194   -9.639469
2471   84.991693  -97.351934  -46.572250  ...    2.293382  -32.795754   37.178331
2472   21.805755   -7.721070   16.017695  ...   46.751081  -38.501283    8.714302
2473   51.544233    4.406096   26.530095  ...  175.222727 -111.555785  109.235822
2474   71.227361  -58.693734  -44.713443  ...   33.974938 -110.099683  -50.220170
2475   -0.519860  -29.082537   -5.154664  ...  -10.932487    3.165286   11.719324
2476   17.340218  -87.628361   -9.814847  ...  -35.574717  -18.992875   73.325820
2477   20.470117  -18.883738  -38.004566  ...   47.927115  -72.226341   56.899275
2478   16.430224  -16.572828  -73.911566  ...    9.812139  -69.849385   20.171175
2479   38.468527   49.134721  -84.843538  ...   22.714533    2.719276   21.809345
2480   54.538454  -49.042756  -33.118759  ...  -46.915401  -16.558754   28.971948
2481   -4.902040  -40.643476 -138.363354  ...   22.563435  -94.684880   52.753951
2482   79.829255   31.979129  -97.884765  ...   38.888568   -1.203058    5.223269
2483   64.009650   28.377054  -63.304777  ...   31.546822  -81.850965  -17.450640
2484   75.222693   56.825661  -21.486143  ...   30.308598    8.032567  -17.623145
2485  142.730869  -60.522212 -209.085222  ...   -1.238353  -10.283261   92.580828
2486   12.223931  -24.808723  -36.779121  ...   -8.371537   10.710338   -3.891283
2487  104.214400 -136.537583 -115.838203  ...  -61.049260  -77.313349   75.104673
2488   37.325945  -43.314250   24.613589  ...    8.937196  -62.454403   29.705684
2489   26.383438  -56.074443   -7.186858  ...   27.555743  -56.422489   97.199866
2490   13.705673   45.025310  -79.784357  ...   20.728342  -30.259561   21.810825
2491   48.237680 -107.792107  -22.860636  ...    1.850410 -104.820573   76.297591
2492   90.703906 -151.341943  -86.859648  ...   11.677919 -136.023148   91.796758
2493   55.197067  -29.355392  -79.955732  ...  106.971037  -26.384290   74.792705
2494  111.371377 -143.463387  -68.685551  ...  -24.824650  -51.910212   50.207569
2495   40.700258 -100.986655  -60.567452  ...   56.510690  -78.969259   60.899952
2496   49.035355  -32.063289  -81.977765  ...  -38.247449  -49.895811   89.846364
2497   91.952072   -6.241876  -37.745051  ...  -26.058092  -61.371228   71.622765
2498   72.003094  -57.647030  -66.112287  ...  -20.329215 -123.427877   25.257072
```

```
2499   15.237106   -9.080193  -89.208226  ...  -10.946389   34.018420  -39.885777

[2500 rows x 96 columns]>
```

```
df_train_afe
```

| | id | qid1 | qid2 | question1 | question2 | is_duplicate | freq_qid1 | freq_ |
|---|---|---|---|---|---|---|---|---|
| **6075** | 6075 | 11913 | 11914 | what is artificial intelligence | what all does artificial intelligence include | 0 | 1 | |
| **1205** | 1205 | 2402 | 2403 | which processor is faster and better for batte... | which one is a better processor 1 8 ghz intel... | 0 | 1 | |
| **16030** | 16030 | 30586 | 25457 | what should be the most important thing in you... | life advice what are some of the most importa... | 0 | 1 | |
| **37040** | 37040 | 67461 | 67462 | what are the largest veins and arteries in the... | what are the major arteries of the human body | 0 | 1 | |
| **44110** | 44110 | 79241 | 79242 | how do bladeless fans work | how does bladeless fan works | 1 | 1 | |
| **29645** | 29645 | 39425 | 54825 | what is meant by surgical strike | what is the meaning of surgical strike | 1 | 1 | |
| **23200** | 23200 | 43491 | 43492 | i leveraged 100k to secure a loan for a startu... | does amalgam filing dangerous | 0 | 1 | |
| **6875** | 6875 | 13455 | 13456 | does china has prime minister | is there prime minister in china | 1 | 1 | |
| **3855** | 3855 | 7635 | 7636 | what is travis kalanick like on investor confe... | what ethnicity is travis kalanick | 0 | 1 | |
| **45765** | 45765 | 49437 | 1215 | is world war 3 coming | is world war 3 more imminent than expected | 1 | 1 | |
| **39590** | 39590 | 46356 | 19540 | how can you cope with loneliness | what are the ways to end loneliness | 1 | 1 | |
| **17900** | 17900 | 33951 | 33952 | why will not richard muller answer my question | how do i get richard muller answer my questions | 0 | 1 | |
| **47020** | 47020 | 84007 | 84008 | what does iq exactly means | what does actually iq | 1 | 1 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | exactly means | mean | | |
| **17660** | 17660 | 33522 | 33523 | are there any substantial way to quit meth | what is the best way to quit meth | 1 | 1 |
| **23675** | 23675 | 44319 | 44320 | what are the future methodology changes in the... | what are the examinations i can appear for aft... | 0 | 1 |
| **20610** | 20610 | 38870 | 38871 | what kind of jobs are byu computer science bi... | is quora a better realization of google own vi... | 0 | 1 |
| **41995** | 41995 | 75736 | 75737 | what can we study after pursuing graduation in... | what are the fields of study after graduating ... | 1 | 1 |
| **32365** | 32365 | 59595 | 59596 | why does my wrist hurt when i cry | why do my wrist hurt when squatting | 0 | 1 |
| **6125** | 6125 | 12008 | 12009 | how do i make green tea | what is the right procedure to make green tea | 1 | 1 |
| **32145** | 32145 | 59207 | 59208 | does electricity travel at the speed of light | is the speed of electricity a synonym for the ... | 1 | 1 |
| **42100** | 42100 | 75910 | 75911 | what is the best strategy to prepare for cat i... | how do i prepare for cat in one month | 1 | 1 |
| **42330** | 42330 | 23143 | 76297 | i am financially stuck in a half baked relatio... | i am looking for a job change but i am unable... | 0 | 2 |
| **42200** | 42200 | 76080 | 76081 | what is the difference between pitch and tar | what are the best react js repositories that f... | 0 | 1 |
| **10805** | 10805 | 20905 | 20906 | which is the best institute in mumbai for doin... | which is the best institute in mumbai from whe... | 1 | 1 |
| | | | | who is the best | what are the | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **28775** | 28775 | 53318 | 53319 | best keyboardist on bits pilani ca... | what are the best pop punk bands | 0 | 1 |
| **19985** | 19985 | 37741 | 37742 | what is the difference between hardware techno... | what is the difference between software and ha... | 0 | 1 |
| **15460** | 15460 | 29534 | 29535 | which type of css js framework used paytm | what js framework should i use on a site with ... | 0 | 1 |
| **4070** | 4070 | 8056 | 8057 | what do you do when you are upset | what you do when you get upset | 1 | 1 |
| **27515** | 27515 | 51113 | 49664 | does house baratheon have any future | is house baratheon extinct | 1 | 1 |
| **29685** | 29685 | 54892 | 54893 | what is the coolest thing or task that you hav... | what were the coolest things you have automated | 1 | 1 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **37395** | 37395 | 68055 | 68056 | what are the most common barriers that affect ... | what are the most common barriers that affect ... | 0 | 1 |
| **22640** | 22640 | 42469 | 42470 | what are the best ways to fake your own death | what are the worst ways to fake one own own de... | 0 | 1 |
| **37365** | 37365 | 67999 | 68000 | what were the books studied by aiims topper 2016 | what books should i study for my pg entrance i... | 0 | 1 |
| **17720** | 17720 | 33624 | 28584 | what happen actually after we die where does ... | what will happen after we die does nothing ha... | 1 | 1 |
| **39365** | 39365 | 71368 | 71369 | how is the march 2 success asvab practice test | my kaplan own practice tests average score is ... | 0 | 1 |
| **8455** | 8455 | 16483 | 16484 | how do i wear red lipstick without sending a ... | how should i convince my son to not wear lipst... | 0 | 1 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **39275** | 39275 | 71223 | 71224 | what advice would you give to someone that giv... | what kind of a person is someone who does not ... | 0 | 1 |
| **7450** | 7450 | 14555 | 14556 | what has been the economic impact from brexit ... | what have been the economic effects of brexit | 1 | 1 |
| **17380** | 17380 | 33031 | 33032 | what are some successful ways to quit smoking | how do you quit smoking | 1 | 1 |
| **33075** | 33075 | 60815 | 60816 | will deafness or blindness be cured | will blindness and deafness be cured | 1 | 1 |
| **6000** | 6000 | 5534 | 11770 | does masturbation cause infertility | does masturbation in males lead to sexual infe... | 1 | 1 |
| **33035** | 33035 | 60746 | 60747 | which is the best and reasonable web hosting s... | which is the best web hosting service provider... | 1 | 1 |
| **9310** | 9310 | 18094 | 18095 | who would win in a fight goku or the hulk | who would win in a fight the hulk or the marv... | 0 | 1 |
| **14590** | 14590 | 27929 | 27930 | why do people want to earn more money | why do people want to earn more money | 1 | 1 |
| **1320** | 1320 | 2632 | 2633 | presently 2015 how many articles parts and ... | how many pages are there in the indian constit... | 0 | 1 |
| **32720** | 32720 | 5297 | 38545 | how do you control your anger | how do i control my anger and have patience | 1 | 2 |
| **45970** | 45970 | 82285 | 82286 | why do women have so much sex | why do women have sex with men | 0 | 1 |
| **43525** | 43525 | 78275 | 78276 | how do i find out someone location through mob... | is there any mobile app through which i can do... | 0 | 1 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **37330** | 37330 | 67946 | 67947 | what is the atomic mass of methane how is it ... | what is relative atomic mass and how is it det... | 0 | 1 |
| **18955** | 18955 | 35862 | 35863 | what is the relationship between power and the... | what is the relationship between power and time | 0 | 1 |
| **27675** | 27675 | 51389 | 51390 | what is the scope for mba operations managemen... | how good is the future of operations managemen... | 1 | 1 |
| **7375** | 7375 | 14410 | 14411 | were any major party candidates as problematic... | i am now 7 sem be mech can i crack gate exam | 0 | 1 |
| **10705** | 10705 | 20715 | 20716 | is deep web really that dangerous | how unsafe the deep web is | 1 | 1 |
| **27765** | 27765 | 51552 | 51553 | what is best average worst case time complex... | what is best algorithm run time complexity | 0 | 1 |
| **38185** | 38185 | 69390 | 69391 | can i use html5 video for backgrounds with the... | how do i use a child theme in wordpress | 0 | 1 |
| **35485** | 35485 | 64833 | 64834 | ball mill ball mill manufacture | do you know ball mill | 0 | 1 |
| **25495** | 25495 | 47518 | 47519 | what is the current ongoing research related t... | what kind of studies are currently ongoing wit... | 0 | 1 |
| **15220** | 15220 | 29094 | 29095 | what is it like to switch to a macbook after ... | why do some people still use windows laptops w... | 0 | 1 |
| **21775** | 21775 | 40957 | 40958 | how big can the iss get | how big is the iss | 0 | 1 |
| **36935** | 36935 | 37176 | 67284 | what is the best perfume under rs 500 for men ... | what are the best perfumes for men that are av... | 0 | 1 |

7500 rows × 32 columns

```
df_train_afe=df_train_afe.reset_index(drop=True)
df_test_afe=df_test_afe.reset_index(drop=True)
```

```
df_train_afe
```

| | id | qid1 | qid2 | question1 | question2 | is_duplicate | freq_qid1 | freq_c |
|---|---|---|---|---|---|---|---|---|
| **0** | 6075 | 11913 | 11914 | what is artificial intelligence | what all does artificial intelligence include | 0 | 1 | |
| **1** | 1205 | 2402 | 2403 | which processor is faster and better for batte... | which one is a better processor 1 8 ghz intel... | 0 | 1 | |
| **2** | 16030 | 30586 | 25457 | what should be the most important thing in you... | life advice what are some of the most importa... | 0 | 1 | |
| **3** | 37040 | 67461 | 67462 | what are the largest veins and arteries in the... | what are the major arteries of the human body | 0 | 1 | |
| **4** | 44110 | 79241 | 79242 | how do bladeless fans work | how does bladeless fan works | 1 | 1 | |
| **5** | 29645 | 39425 | 54825 | what is meant by surgical strike | what is the meaning of surgical strike | 1 | 1 | |
| **6** | 23200 | 43491 | 43492 | i leveraged 100k to secure a loan for a startu... | does amalgam filing dangerous | 0 | 1 | |
| **7** | 6875 | 13455 | 13456 | does china has prime minister | is there prime minister in china | 1 | 1 | |
| **8** | 3855 | 7635 | 7636 | what is travis kalanick like on investor confe... | what ethnicity is travis kalanick | 0 | 1 | |
| **9** | 45765 | 49437 | 1215 | is world war 3 coming | is world war 3 more imminent than expected | 1 | 1 | |
| **10** | 39590 | 46356 | 19540 | how can you cope with loneliness | what are the ways to end loneliness | 1 | 1 | |
| **11** | 17900 | 33951 | 33952 | why will not richard muller answer my question | how do i get richard muller answer my questions | 0 | 1 | |
| **12** | 47020 | 84007 | 84008 | what does iq exactly means | what does actually iq | 1 | 1 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | exactly means | mean | | |
| **13** | 17660 | 33522 | 33523 | are there any substantial way to quit meth | what is the best way to quit meth | 1 | 1 |
| **14** | 23675 | 44319 | 44320 | what are the future methodology changes in the... | what are the examinations i can appear for aft... | 0 | 1 |
| **15** | 20610 | 38870 | 38871 | what kind of jobs are byu computer science bi... | is quora a better realization of google own vi... | 0 | 1 |
| **16** | 41995 | 75736 | 75737 | what can we study after pursuing graduation in... | what are the fields of study after graduating ... | 1 | 1 |
| **17** | 32365 | 59595 | 59596 | why does my wrist hurt when i cry | why do my wrist hurt when squatting | 0 | 1 |
| **18** | 6125 | 12008 | 12009 | how do i make green tea | what is the right procedure to make green tea | 1 | 1 |
| **19** | 32145 | 59207 | 59208 | does electricity travel at the speed of light | is the speed of electricity a synonym for the ... | 1 | 1 |
| **20** | 42100 | 75910 | 75911 | what is the best strategy to prepare for cat i... | how do i prepare for cat in one month | 1 | 1 |
| **21** | 42330 | 23143 | 76297 | i am financially stuck in a half baked relatio... | i am looking for a job change but i am unable... | 0 | 2 |
| **22** | 42200 | 76080 | 76081 | what is the difference between pitch and tar | what are the best react js repositories that f... | 0 | 1 |
| **23** | 10805 | 20905 | 20906 | which is the best institute in mumbai for doin... | which is the best institute in mumbai from whe... | 1 | 1 |
| | | | | who is the best | what are the | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **24** | 28775 | 53318 | 53319 | best keyboardist on bits pilani ca... | what are the best pop punk bands | 0 | 1 |
| **25** | 19985 | 37741 | 37742 | what is the difference between hardware techno... | what is the difference between software and ha... | 0 | 1 |
| **26** | 15460 | 29534 | 29535 | which type of css js framework used paytm | what js framework should i use on a site with ... | 0 | 1 |
| **27** | 4070 | 8056 | 8057 | what do you do when you are upset | what you do when you get upset | 1 | 1 |
| **28** | 27515 | 51113 | 49664 | does house baratheon have any future | is house baratheon extinct | 1 | 1 |
| **29** | 29685 | 54892 | 54893 | what is the coolest thing or task that you hav... | what were the coolest things you have automated | 1 | 1 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **7470** | 37395 | 68055 | 68056 | what are the most common barriers that affect ... | what are the most common barriers that affect ... | 0 | 1 |
| **7471** | 22640 | 42469 | 42470 | what are the best ways to fake your own death | what are the worst ways to fake one own own de... | 0 | 1 |
| **7472** | 37365 | 67999 | 68000 | what were the books studied by aiims topper 2016 | what books should i study for my pg entrance i... | 0 | 1 |
| **7473** | 17720 | 33624 | 28584 | what happen actually after we die where does ... | what will happen after we die does nothing ha... | 1 | 1 |
| **7474** | 39365 | 71368 | 71369 | how is the march 2 success asvab practice test | my kaplan own practice tests average score is ... | 0 | 1 |
| **7475** | 8455 | 16483 | 16484 | how do i wear red lipstick without sending a ... | how should i convince my son to not wear lipst... | 0 | 1 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **7476** | 39275 | 71223 | 71224 | what advice would you give to someone that giv... | what kind of a person is someone who does not ... | 0 | 1 |
| **7477** | 7450 | 14555 | 14556 | what has been the economic impact from brexit ... | what have been the economic effects of brexit | 1 | 1 |
| **7478** | 17380 | 33031 | 33032 | what are some successful ways to quit smoking | how do you quit smoking | 1 | 1 |
| **7479** | 33075 | 60815 | 60816 | will deafness or blindness be cured | will blindness and deafness be cured | 1 | 1 |
| **7480** | 6000 | 5534 | 11770 | does masturbation cause infertility | does masturbation in males lead to sexual infe... | 1 | 1 |
| **7481** | 33035 | 60746 | 60747 | which is the best and reasonable web hosting s... | which is the best web hosting service provider... | 1 | 1 |
| **7482** | 9310 | 18094 | 18095 | who would win in a fight goku or the hulk | who would win in a fight the hulk or the marv... | 0 | 1 |
| **7483** | 14590 | 27929 | 27930 | why do people want to earn more money | why do people want to earn more money | 1 | 1 |
| **7484** | 1320 | 2632 | 2633 | presently 2015 how many articles parts and ... | how many pages are there in the indian constit... | 0 | 1 |
| **7485** | 32720 | 5297 | 38545 | how do you control your anger | how do i control my anger and have patience | 1 | 2 |
| **7486** | 45970 | 82285 | 82286 | why do women have so much sex | why do women have sex with men | 0 | 1 |
| **7487** | 43525 | 78275 | 78276 | how do i find out someone location through mob... | is there any mobile app through which i can do... | 0 | 1 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **7488** | 37330 | 67946 | 67947 | what is the atomic mass of methane how is it ... | what is relative atomic mass and how is it det... | 0 | 1 |
| **7489** | 18955 | 35862 | 35863 | what is the relationship between power and the... | what is the relationship between power and time | 0 | 1 |
| **7490** | 27675 | 51389 | 51390 | what is the scope for mba operations managemen... | how good is the future of operations managemen... | 1 | 1 |
| **7491** | 7375 | 14410 | 14411 | were any major party candidates as problematic... | i am now 7 sem be mech can i crack gate exam | 0 | 1 |
| **7492** | 10705 | 20715 | 20716 | is deep web really that dangerous | how unsafe the deep web is | 1 | 1 |
| **7493** | 27765 | 51552 | 51553 | what is best average worst case time complex... | what is best algorithm run time complexity | 0 | 1 |
| **7494** | 38185 | 69390 | 69391 | can i use html5 video for backgrounds with the... | how do i use a child theme in wordpress | 0 | 1 |
| **7495** | 35485 | 64833 | 64834 | ball mill ball mill manufacture | do you know ball mill | 0 | 1 |
| **7496** | 25495 | 47518 | 47519 | what is the current ongoing research related t... | what kind of studies are currently ongoing wit... | 0 | 1 |
| **7497** | 15220 | 29094 | 29095 | what is it like to switch to a macbook after ... | why do some people still use windows laptops w... | 0 | 1 |
| **7498** | 21775 | 40957 | 40958 | how big can the iss get | how big is the iss | 0 | 1 |
| **7499** | 36935 | 37176 | 67284 | what is the best perfume under rs 500 for men ... | what are the best perfumes for men that are av... | 0 | 1 |

7500 rows × 32 columns

```
df_train_ave_final=pd.concat([df_train_ave,df_train_ave2],axis=1)
df_test_ave_final=pd.concat([df_test_ave,df_test_ave2],axis=1)
```

```
X_train=pd.concat([df_train_afe,df_train_ave_final],axis=1)
X_test=pd.concat([df_test_afe,df_test_ave_final],axis=1)
```

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

```
(7500, 224)
(2500, 224)
```

```
y_train= X_train['is_duplicate']
y_test=X_test['is_duplicate']
```

```
X_train.drop([ 'id','qid1','qid2','question1','question2','is_duplicate'], axis=1, inplace
X_test.drop([ 'id','qid1','qid2','question1','question2','is_duplicate'], axis=1, inplace=
```

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

```
(7500, 218)
(7500,)
(2500, 218)
(2500,)
```

```
print(X_train.columns)
```

```
Index([  'freq_qid1',   'freq_qid2',       'q1len',       'q2len',
        'q1_n_words',  'q2_n_words', 'word_Common',  'word_Total',
        'word_share',  'freq_q1+q2',
       ...
                  86,            87,            88,            89,
                  90,            91,            92,            93,
                  94,            95],
      dtype='object', length=218)
```

```
   X_train.columns =([ 'freq_qid1','freq_qid2','q1len','q2len','q1_n_words','q2_n_words','
 'cwc_min','cwc_max','csc_min','csc_max','ctc_min','ctc_max','last_word_eq','first_word_eq'
      'mean_len','token_set_ratio','token_sort_ratio','fuzz_ratio','fuzz_partial_ratio','l
      '0_x','1_x','2_x','3_x','4_x','5_x','6_x','7_x','8_x','9_x','10_x','11_x','12_x','13
      '21_x','22_x','23_x','24_x','25_x','26_x','27_x','28_x','29_x','30_x','31_x','32_x',
      '41_x','42_x','43_x','44_x','45_x','46_x','47_x','48_x','49_x','50_x','51_x','52_x',
      '61_x','62_x','63_x','64_x','65_x','66_x','67_x','68_x','69_x','70_x','71_x','72_x',
      '81_x','82_x','83_x','84_x','85_x','86_x','87_x','88_x','89_x','90_x','91_x','92_x',
      '0_y','1_y','2_y','3_y','4_y','5_y','6_y','7_y','8_y','9_y','10_y','11_y','12_y','13
      '19_y','20_y','21_y','22_y','23_y','24_y','25_y','26_y','27_y','28_y','29_y','30_y',
      '37_y','38_y','39_y','40_y','41_y','42_y','43_y','44_y','45_y','46_y','47_y','48_y',
```

```
        '55_y','56_y','57_y','58_y','59_y','60_y','61_y','62_y','63_y','64_y','65_y','66_y',
        '73_y','74_y','75_y','76_y','77_y','78_y','79_y','80_y','81_y','82_y','83_y','84_y',
        '91_y','92_y','93_y','94_y','95_y'])
```

```
   X_test.columns =([ 'freq_qid1','freq_qid2','q1len','q2len','q1_n_words','q2_n_words','w
'cwc_min','cwc_max','csc_min','csc_max','ctc_min','ctc_max','last_word_eq','first_word_eq'
      'mean_len','token_set_ratio','token_sort_ratio','fuzz_ratio','fuzz_partial_ratio','l
      '0_x','1_x','2_x','3_x','4_x','5_x','6_x','7_x','8_x','9_x','10_x','11_x','12_x','13
      '21_x','22_x','23_x','24_x','25_x','26_x','27_x','28_x','29_x','30_x','31_x','32_x',
      '41_x','42_x','43_x','44_x','45_x','46_x','47_x','48_x','49_x','50_x','51_x','52_x',
      '61_x','62_x','63_x','64_x','65_x','66_x','67_x','68_x','69_x','70_x','71_x','72_x',
      '81_x','82_x','83_x','84_x','85_x','86_x','87_x','88_x','89_x','90_x','91_x','92_x',
      '0_y','1_y','2_y','3_y','4_y','5_y','6_y','7_y','8_y','9_y','10_y','11_y','12_y','13
      '19_y','20_y','21_y','22_y','23_y','24_y','25_y','26_y','27_y','28_y','29_y','30_y',
      '37_y','38_y','39_y','40_y','41_y','42_y','43_y','44_y','45_y','46_y','47_y','48_y',
      '55_y','56_y','57_y','58_y','59_y','60_y','61_y','62_y','63_y','64_y','65_y','66_y',
      '73_y','74_y','75_y','76_y','77_y','78_y','79_y','80_y','81_y','82_y','83_y','84_y',
      '91_y','92_y','93_y','94_y','95_y'])
```

# XGBOOST

```
#https://www.kaggle.com/tilii7/hyperparameter-grid-search-with-xgboost
from xgboost import XGBClassifier
from sklearn.model_selection import RandomizedSearchCV
xgb = XGBClassifier(learning_rate=0.02, n_estimators=600, objective='binary:logistic')
params = {
'min_child_weight': [1, 5, 10],
'gamma': [0.5, 1, 1.5, 2, 5],
'subsample': [0.6, 0.8, 1.0],
'colsample_bytree': [0.6, 0.8, 1.0],
'max_depth': [3, 4, 5]
}
random_search = RandomizedSearchCV(xgb, param_distributions=params, scoring='roc_auc', n_j


random_search.fit(X_train, y_train)
```

```
Fitting 5 folds for each of 10 candidates, totalling 50 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 2 concurrent workers.
[Parallel(n_jobs=-1)]: Done  28 tasks      | elapsed: 20.5min
[Parallel(n_jobs=-1)]: Done  50 out of  50 | elapsed: 33.3min finished
RandomizedSearchCV(cv=5, error_score='raise-deprecating',
                   estimator=XGBClassifier(base_score=0.5, booster='gbtree',
                                           colsample_bylevel=1,
                                           colsample_bynode=1,
                                           colsample_bytree=1, gamma=0,
                                           learning_rate=0.02, max_delta_step=0,
                                           max_depth=3, min_child_weight=1,
                                           missing=None, n_estimators=600,
                                           n_jobs=1, nthread=None,
                                           objective='binary:logistic',
                                           random_state=0, reg_alpha=0,
                                           reg_lambda=1, scale_pos_weight=1,
                                           seed=None, silent=None, subsample=1,
                                           verbosity=1),
                   iid='warn', n_iter=10, n_jobs=-1,
                   param_distributions={'colsample_bytree': [0.6, 0.8, 1.0],
                                        'gamma': [0.5, 1, 1.5, 2, 5],
                                        'max_depth': [3, 4, 5],
                                        'min_child_weight': [1, 5, 10],
                                        'subsample': [0.6, 0.8, 1.0]},
                   pre_dispatch='2*n_jobs', random_state=1001, refit=True,
                   return_train_score=False, scoring='roc_auc', verbose=3)
```

```
print(random_search.best_estimator_)
```

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=0.8, gamma=1,
              learning_rate=0.02, max_delta_step=0, max_depth=5,
              min_child_weight=5, missing=None, n_estimators=600, n_jobs=1,
              nthread=None, objective='binary:logistic', random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
              silent=None, subsample=0.8, verbosity=1)
```

```
xgb = XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=0.8, gamma=1,
              learning_rate=0.02, max_delta_step=0, max_depth=5,
              min_child_weight=5, missing=None, n_estimators=600, n_jobs=1,
              nthread=None, objective='binary:logistic', random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
              silent=None, subsample=0.8, verbosity=1)

xgb.fit(X_train, y_train)

predict_y = xgb.predict(X_test)
```

```
predicted_y =np.array(predict_y>0.5,dtype=int)
print("Total number of data points :", len(predicted_y))
plot_confusion_matrix(y_test, predicted_y)
```

```
Total number of data points : 2500
```