- Over 100 million people visit Quora every month, so it's no surprise that many people ask similarly worded questions.
- Multiple questions with the same intent can cause seekers to spend more time finding the best answer to their question, and make writers feel they need to answer multiple versions of the same question.
- Quora values canonical questions because they provide a better experience to active seekers and writers, and offer more value to both of these groups in the long term.

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

DATA OVERVIEW

- Data will be in a file Train.csv

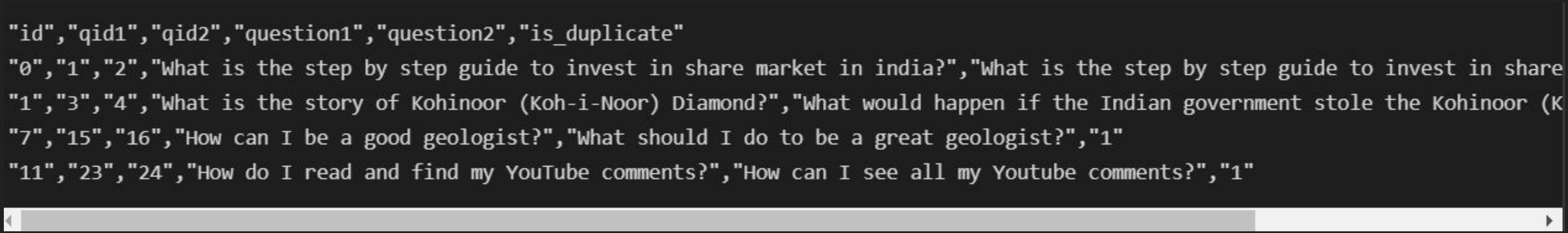
- Train.csv contains 5 columns : qid1, qid2, question1, question2, is_duplicate

- Size of Train.csv - 60MB

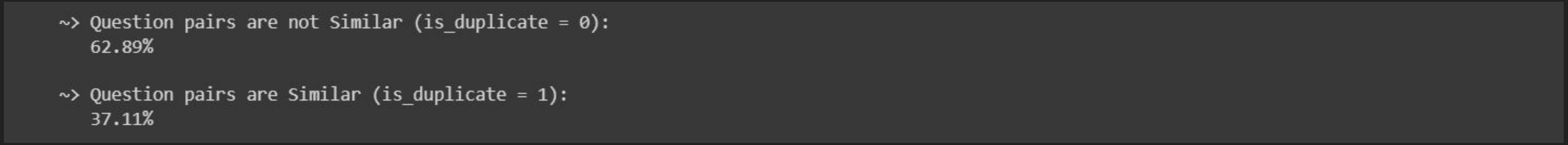
- Number of rows in Train.csv = 404,290

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

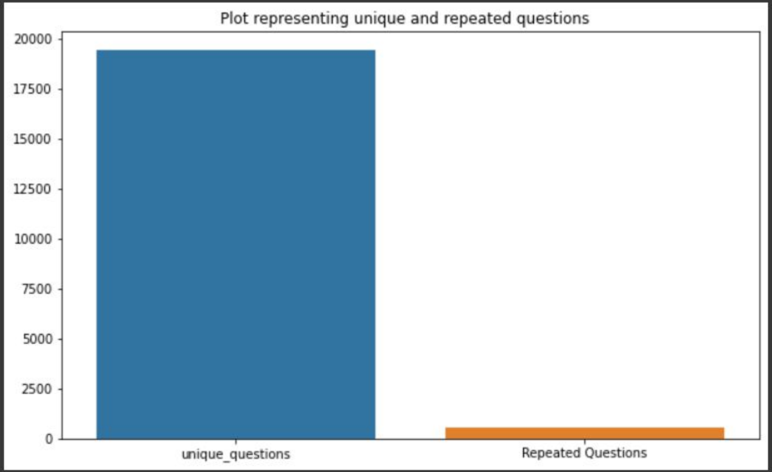
as we have sufficient points to work with

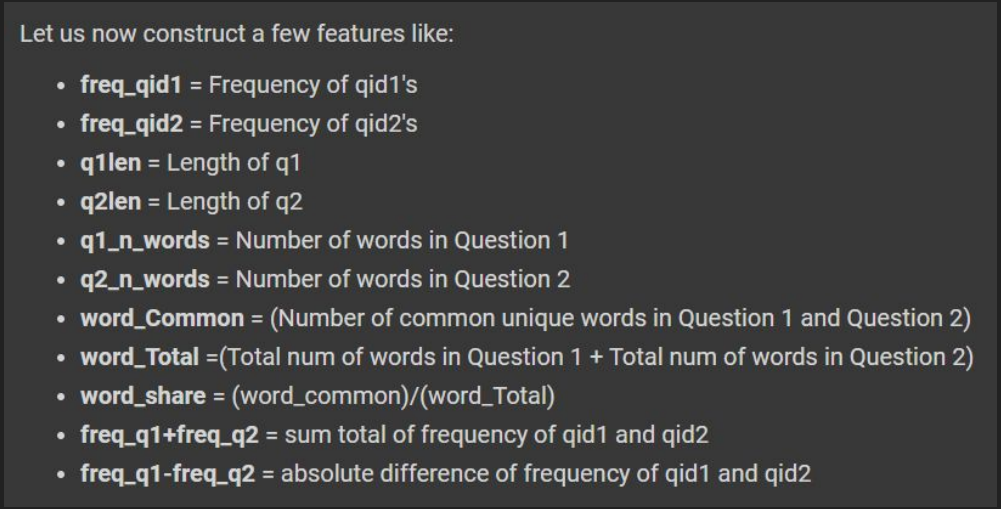


```
"id","qid1","qid2","question1","question2","is_duplicate"
"0","1","2","What is the step by step guide to invest in share market in india?","What is the step by step guide to invest in share
"1","3","4","What is the story of Kohinoor (Koh-i-Noor) Diamond?","What would happen if the Indian government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



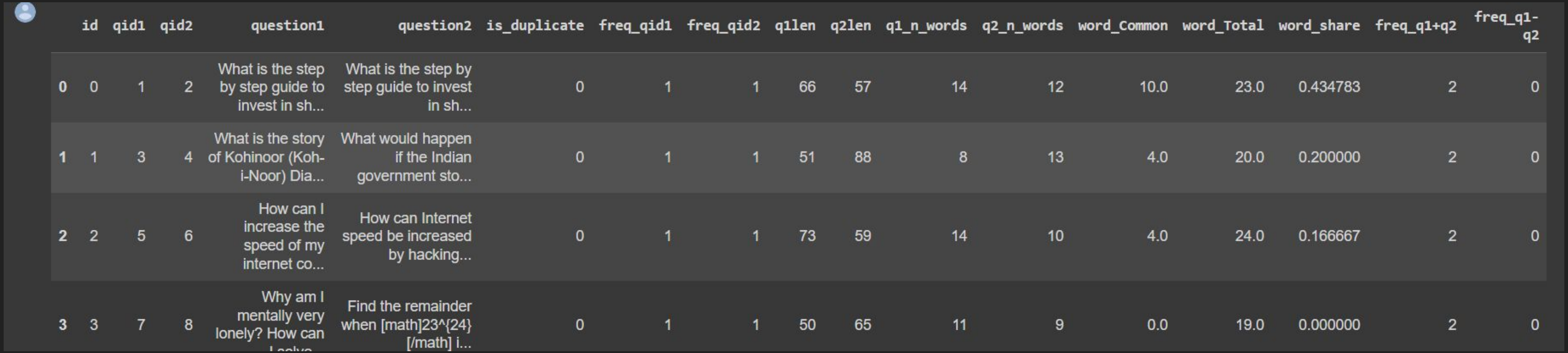
```
    ~> Question pairs are not Similar (is_duplicate = 0):
       62.89%

    ~> Question pairs are Similar (is_duplicate = 1):
       37.11%
```



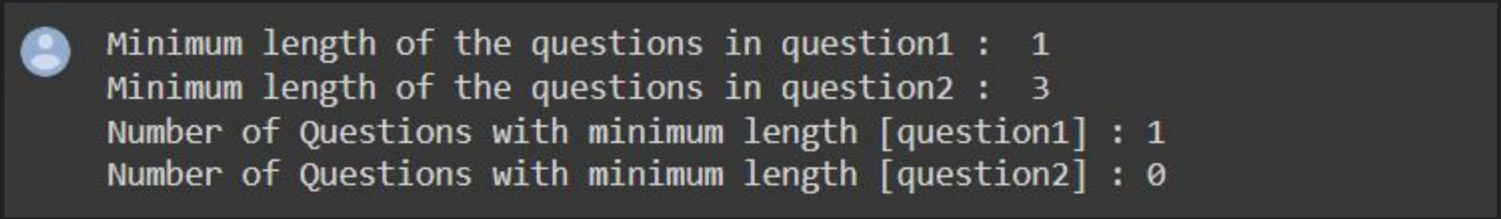


Let us now construct a few features like:

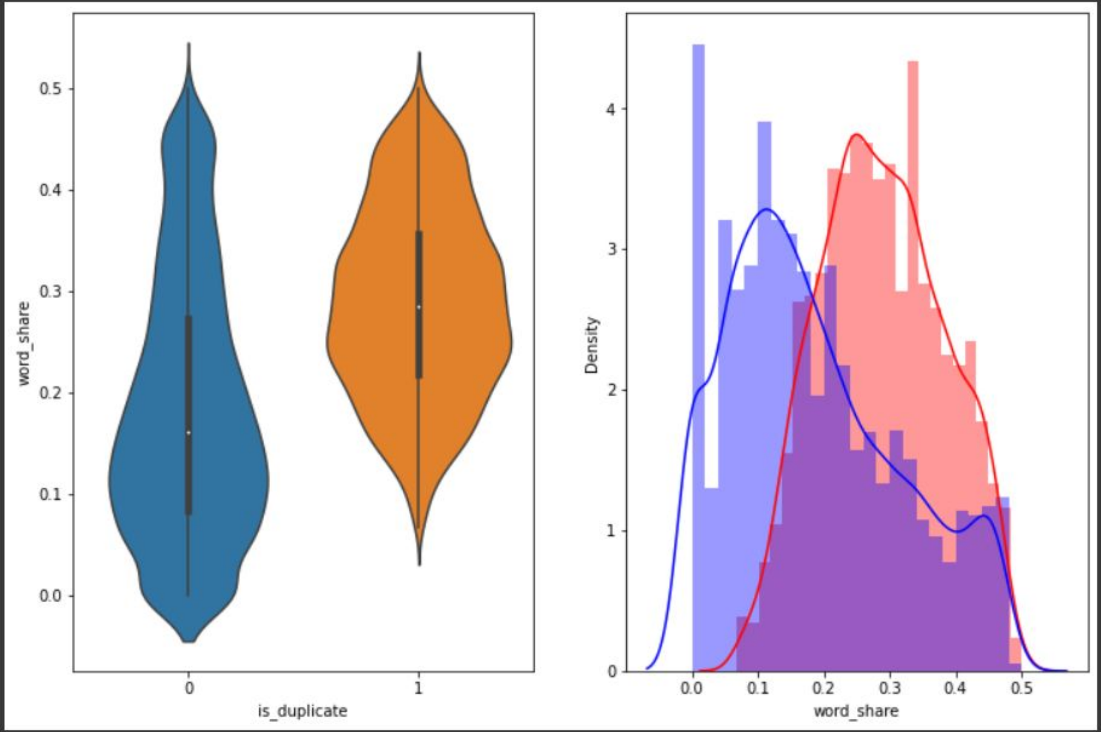
- **freq_qid1** = Frequency of qid1's
- **freq_qid2** = Frequency of qid2's
- **q1len** = Length of q1
- **q2len** = Length of q2
- **q1_n_words** = Number of words in Question 1
- **q2_n_words** = Number of words in Question 2
- **word_Common** = (Number of common unique words in Question 1 and Question 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

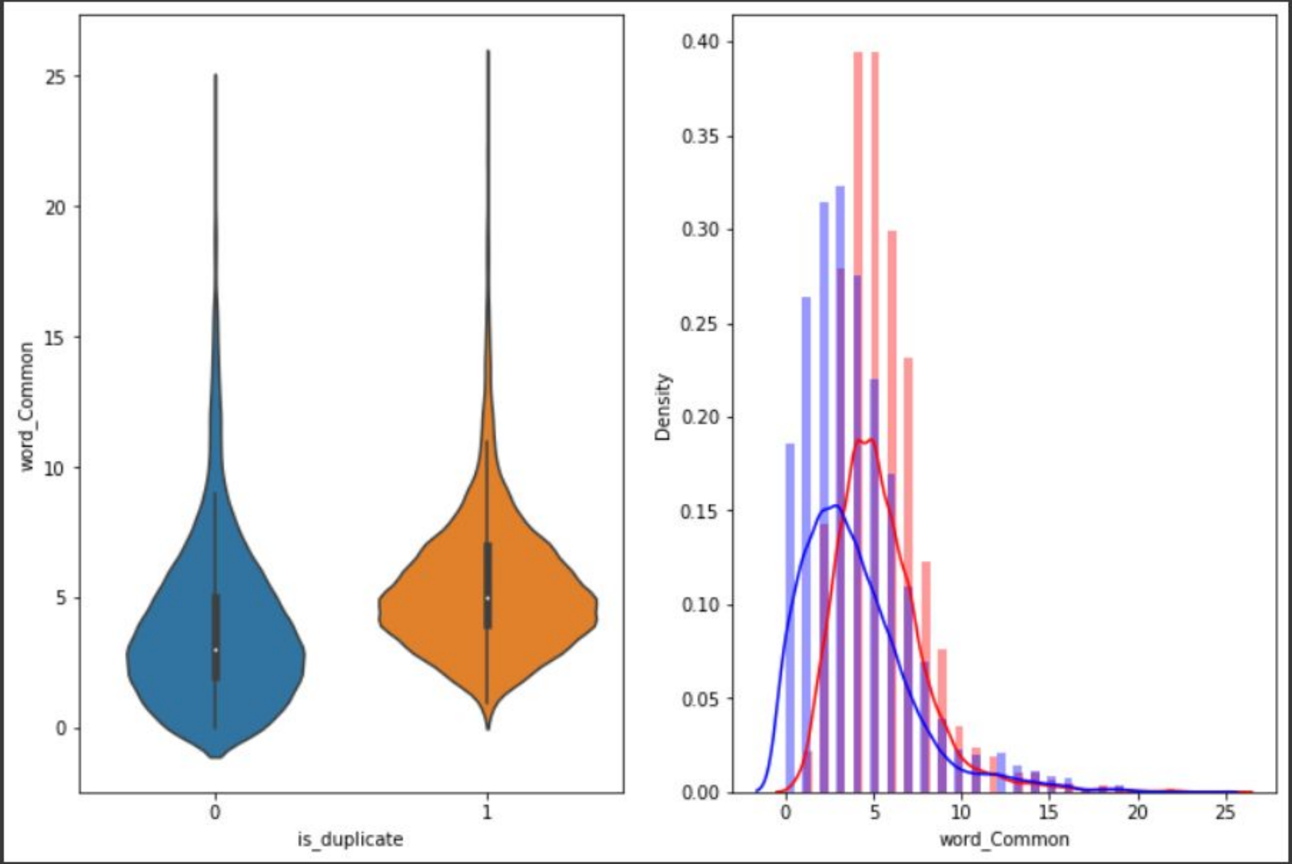


	id	qid1	qid2	question1	question2	is_duplicate	freq_qid1	freq_qid2	q1len	q2len	q1_n_words	q2_n_words	word_Common	word_Total	word_share	freq_q1+q2	freq_q1-q2
0	0	1	2	What is the step by step guide to invest in sh...	What is the step by step guide to invest in sh...	0	1	1	66	57	14	12	10.0	23.0	0.434783	2	0
1	1	3	4	What is the story of Kohinoor (Koh-i-Noor) Dia...	What would happen if the Indian government sto...	0	1	1	51	88	8	13	4.0	20.0	0.200000	2	0
2	2	5	6	How can I increase the speed of my internet co...	How can Internet speed be increased by hacking...	0	1	1	73	59	14	10	4.0	24.0	0.166667	2	0
3	3	7	8	Why am I mentally very lonely? How can I solve...	Find the remainder when [math]23^{24}[/math] i...	0	1	1	50	65	11	9	0.0	19.0	0.000000	2	0



```
Minimum length of the questions in question1 :  1
Minimum length of the questions in question2 :  3
Number of Questions with minimum length [question1] : 1
Number of Questions with minimum length [question2] : 0
```




word_Common
25
20
15
10
5
0
0
1
is_duplicate
Density
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
5
10
15
20
25
word_Common

PREPROCESSING

Preprocessing:preparing the raw data

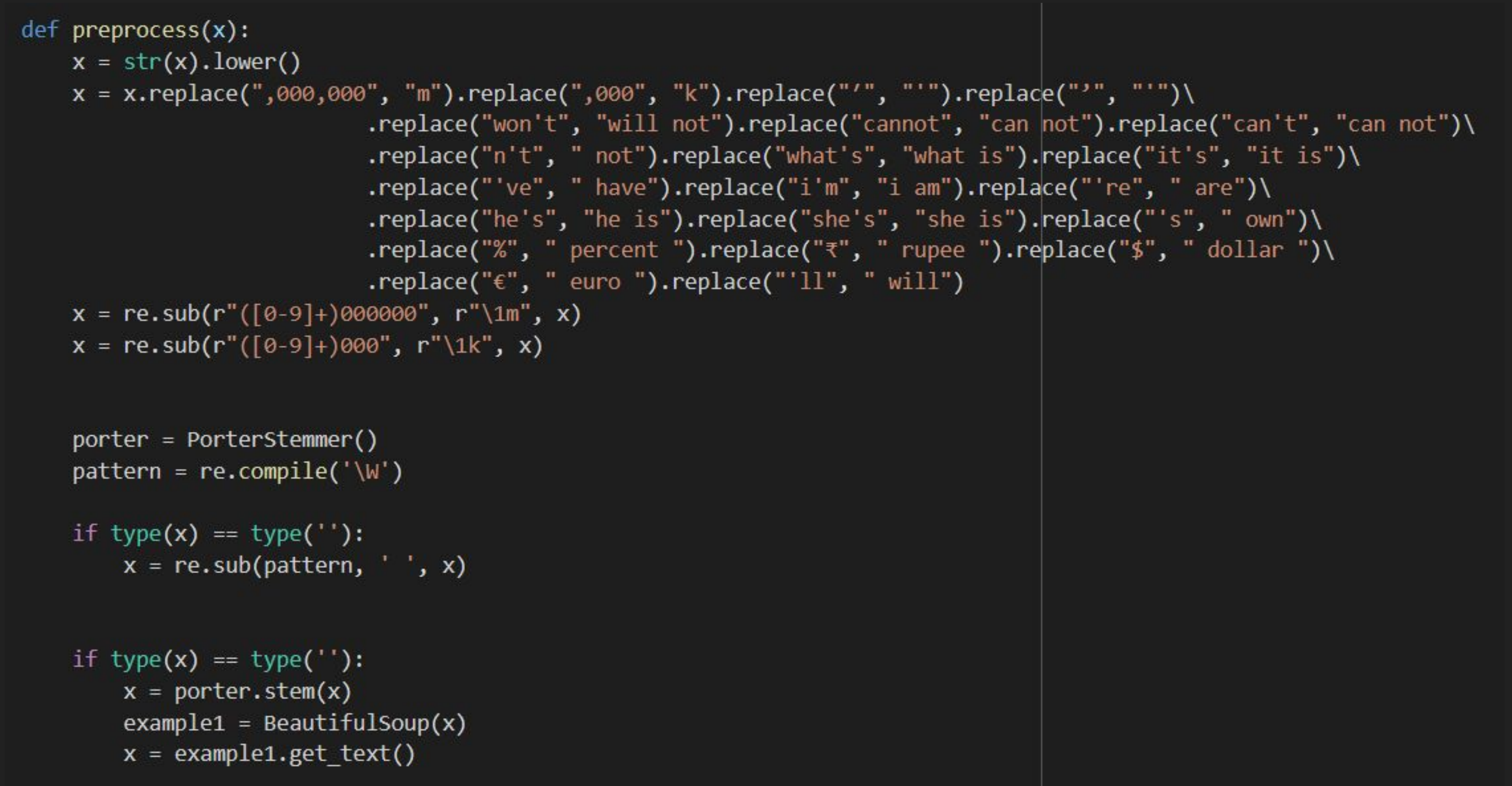
Removing html tags -------------><html> <head> <div>

Removing Punctuations ------->(, . / ; ’ “)

Performing stemming ---------->*eating, eats, eaten* is *eat*.

Removing Stopwords ----------->the , a , in , an

Expanding contractions etc-->i’ll = i will



```python
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").replace("can't", "can not")\
                           .replace("n't", " not").replace("what's", "what is").replace("it's", "it is")\
                           .replace("'ve", " have").replace("i'm", "i am").replace("'re", " are")\
                           .replace("he's", "he is").replace("she's", "she is").replace("'s", " own")\
                           .replace("%", " percent ").replace("₹", " rupee ").replace("$", " dollar ")\
                           .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)
        example1 = BeautifulSoup(x)
        x = example1.get_text()
```

FEATURE EXTRACTION

NLP AND FUZZY FEATURES

Definition:

- Token: by splitting sentence a space
- Stop_Word : stop words as per NLTK.
- Word : 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))

FEATURES

- 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 Last 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

Machine Learning Models

Reading data from file and storing into sql table

```
final_features.csv  —---> train.db
```

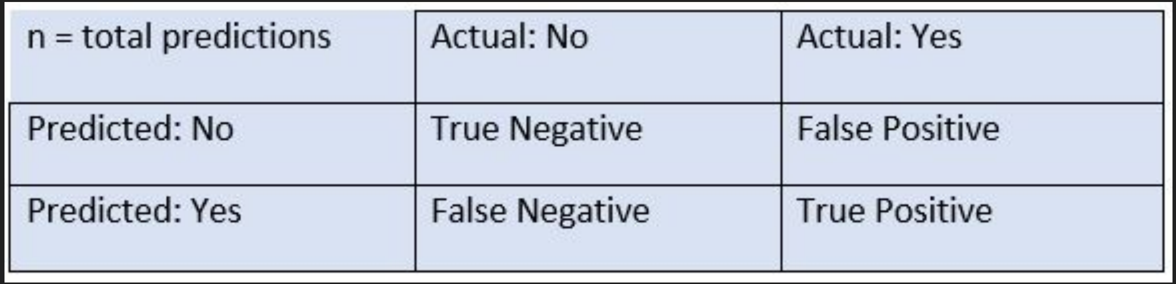
Random train test split

70:30

```
X_train,X_test, y_train, y_test = train_test_split(data, y_true, stratify=y_true,
test_size=0.3,random_state=13)
```

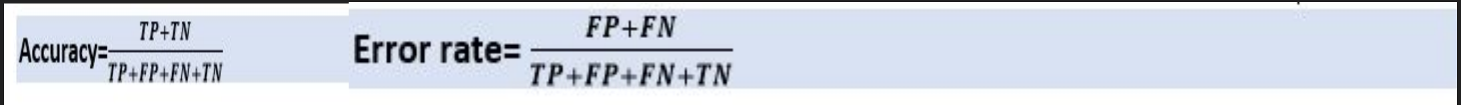
CONFUSION MATRIX

The confusion matrix is a matrix used to determine the performance of the classification models for a given set of test data. It can only be determined if the true values for test data are known. The matrix itself can be easily understood, but the related terminologies may be confusing. Since it shows the errors in the model performance in the form of a matrix, hence also known as an **error matrix.**



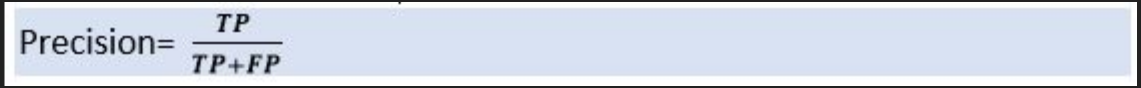
n = total predictions	Actual: No	Actual: Yes
Predicted: No	True Negative	False Positive
Predicted: Yes	False Negative	True Positive

-
- It evaluates the performance of the classification models, when they make predictions on test data, and tells how good our classification model is.
- It not only tells the error made by the classifiers but also the type of errors such as it is either type-I or type-II error.
- With the help of the confusion matrix, we can calculate the different parameters for the model, such as accuracy, precision, etc.



Accuracy= $\frac{TP+TN}{TP+FP+FN+TN}$

Error rate= $\frac{FP+FN}{TP+FP+FN+TN}$



Precision= $\frac{TP}{TP+FP}$