QUORA DUPLICATE QUESTION PREDICTION

MACHINE LEARNING ALGORITHM

CAPSTONE PROJECT – MILESTONE 1

VERSION HISTORY	DATE	PREPARER	APPROVER
ISSUED FOR INFORMATION	07/16/2020	JUGAL SHAH	KEVIN GLYNN

Table of Contents

List of Figures

List of Tables

1. SUMMARY

This report explores the task Natural Language Understanding (NLU) by looking at duplicate question detection in the Quora dataset. Extensive exploration was conducted of the dataset and used various machine learning models, including linear and tree-based models. This report represents initial findings and exploratory data analysis.

The system proposed in the report got an accuracy score of 78% when trained on a subset of approximately 160,000 question pairs sampled from the training data.

2. ABOUT QUORA

Quora's mission is to share and grow the world's knowledge. A vast amount of the knowledge that would be valuable to many people is currently only available to a few — either locked in people's heads, or only accessible to select groups. We want to connect the people who have knowledge to the people who need it, to bring together people with different perspectives so they can understand each other better, and to empower everyone to share their knowledge for the benefit of the rest of the world.

Quora has only one version of each question. It doesn't have a left-wing version, a right-wing version, a western version, and an eastern version. Quora brings together people from different worlds to answer the same question, in the same place — and to learn from each other. We want Quora to be the place to voice your opinion because Quora is where the debate is happening. We want the Quora answer to be the definitive answer for everybody forever.

3. INFORMATION ABOUT DATASETS

The Quora Question Pairs dataset consists of a training set of 404,290 question pairs, and a test set of 2,345,795 question pairs, and is provided as part of a Kaggle competition. Since the test set provided does not contain labels for any question pair, the only measure of performance that can be obtained with this test set is accuracy (via online submission to Kaggle). Therefore, separate test set was constructed from the training set provided, since this would allow to obtain performance metrics other than accuracy and perform further error analysis of prediction models. Thus, this data exploration only considered training set of 404,290 question pairs.

Each sample point has the following fields:

- id: unique ID of each pair
- qid1: ID of first question
- qid2: ID of second question
- question1: text of first question
- question2: text of second question
- is duplicate: are the questions duplicates of each other (0 indicates not duplicate, 1 indicates duplicate)

Of the 404,290 question pairs, 255,027 (63.08%) have a negative (0) label, and 149,263 (36.92%) have a positive (1) label, making our dataset unbalanced.

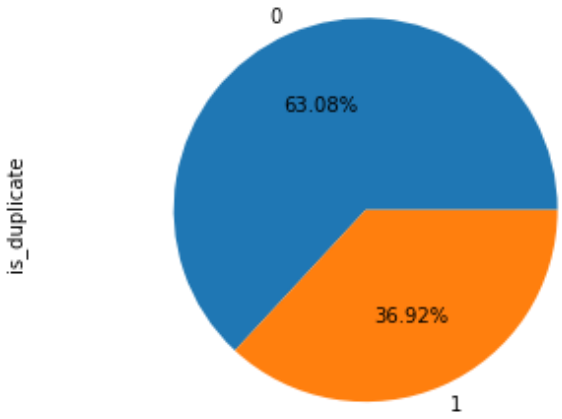


Figure 3-1 : Proportion of positive and negative data

While every question pair is unique, every question within the questions pairs is not; 79.22% of questions appear more than once, with one of the questions appearing 158 times. Across all question pairs, there are 537,933 unique questions. Of these, 111,780 questions occur across multiple pairs. Figure 1 shows the number of times a question appears against the number of questions for that many occurrences.

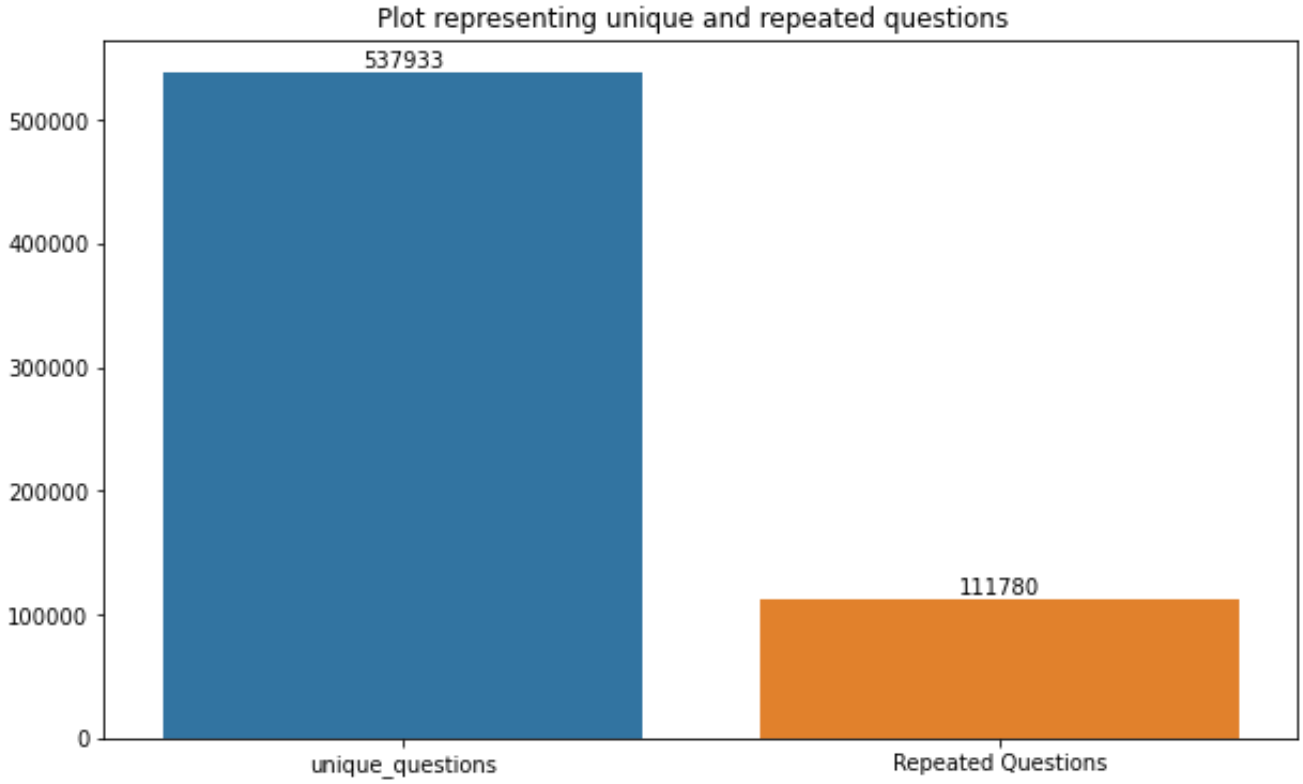


Figure 3-2 : Number of unique question and duplicate question

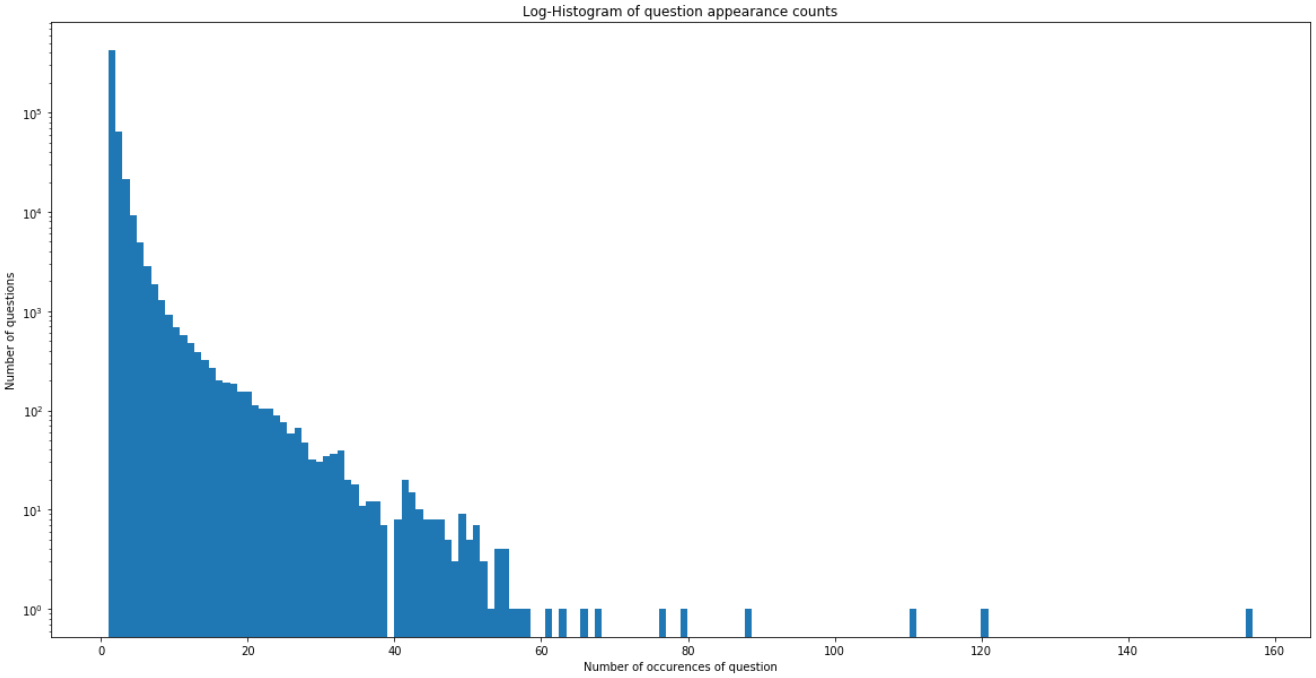


Figure 3-3 : Number of times questions appeared in dataset

4. DATA CLEANING METHODOLOGY

The data is a mass of human generated text and, unsurprisingly, contains anomalies such as nonASCII characters. Different preprocessing steps applied to eliminate different anomalies, and resultantly lowered the vocabulary size.

For this analysis, nltk.tokenizer for tokenization as a universal preprocessing step. In addition, for the linear models we removed non-ASCII characters and for the tree-based models we moved punctuation. Stop words from nltk library is used to remove stop words.

Also, set of words created from sentences in order to remove duplicate words in same sentence. That will further increase common word ratio and similarity scores.

5. EXPLORATORY DATA ANALYSIS

Based on given dataset, some new features where generated to analyze dataset.

A new data frame is created with combining question1 and qeustion2 from data set and named as 'df_all_question'.

Within this data frame, features like word count and character count is generated.

Table 5-1 : Descriptive statistics of words and characters in dataset

	Word Count	Character count
Minimum	237	1169
Average	11.07	59.82
Maximum	1	1

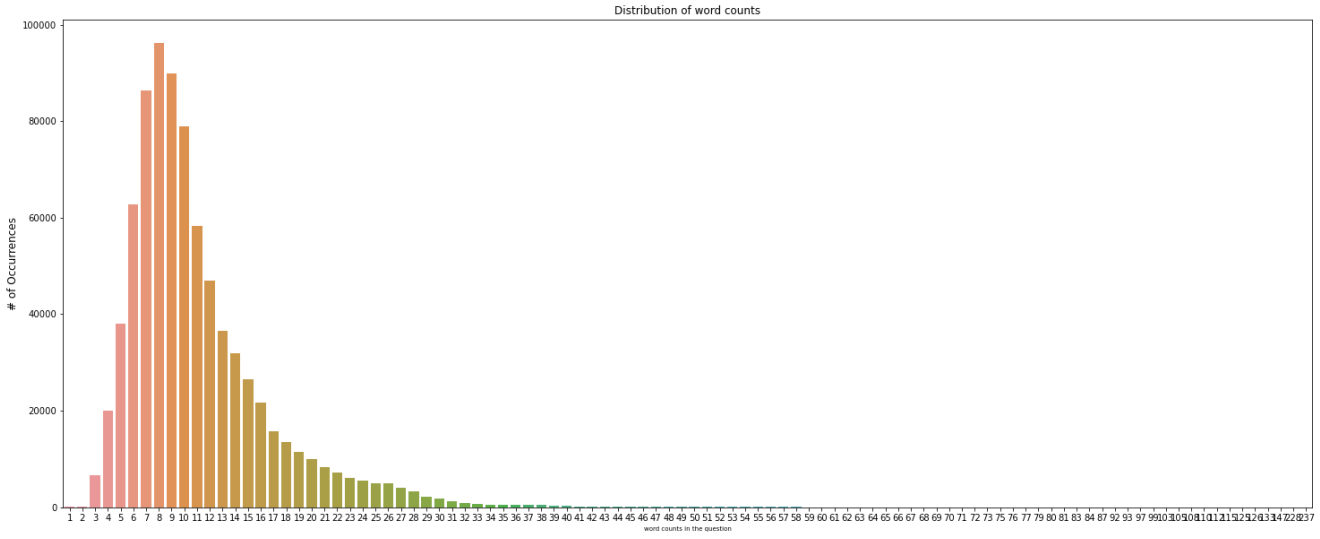


Figure 5-1 : Distribution of number of words in each question

Some questions are with only one or two charters. Most of them were '?'. It is decided to ignore questions with less than 10 character. Some questions with 10 characters are as below and it appears to be genuine questions. Questions with less than 10 characters were inspected and they were not valid questions.

- Is 30 old?
- What is ∈?
- Am I lazy?
- Can I sue?

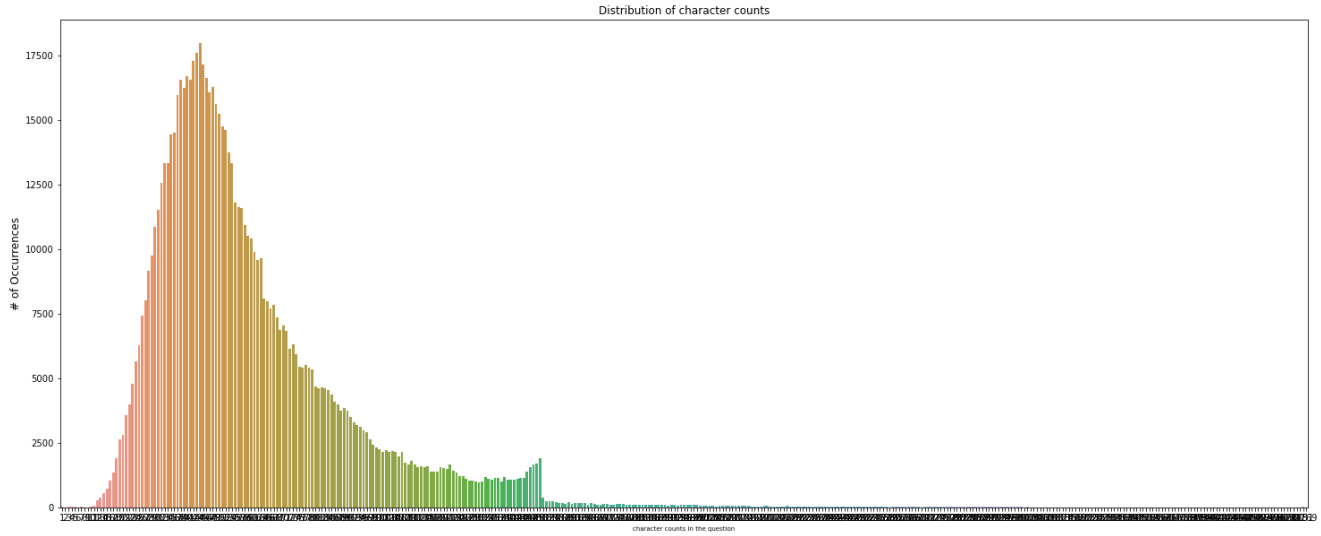


Figure 5-2 : Distribution of number of characters in each question

	question1	question2	unigram_ques1	unigram_ques2
918	i am what should i do	what can i do when i am	[]	[]
1510	if and what is	why can i not multiply fractions in python	[]	[python, multiply, fractions]
7120	is it proper to use a mma after saying thank you	what is here and not there	[saying, mma, proper, thank, use]	[]
7368	what is	what is	[]	[]
7820	why and how is	why is equal to	[]	[equal]
9581	how can i just be myself	how can i not be myself	[]	[]
9746	what is	what is	[]	[]
10614	if and what is	how do you solve base divided by base where is	[]	[base, solve, divided]
13797	what is the most visited tourist attraction in africa		[tourist, africa, visited, attraction]	[]
17486	i am neither good at studies nor at anything else what should a loser like me do to transform self	if then why	[transform, self, good, neither, else, anything, studies, like, loser]	[]

There were total of 228 rows with empty unigram. These rows also dropped from dataset.

Apart from that, from main dataset, common words between each questions pair is calculated and new feature 'common word ratio' is generated. Common word ratio is defined as ratio of total no. of common words between pair of question and total no. of words in each question. On average there are 3-4 common words in questions pairs.

Apart from applying text cleaning, also 'English Stop Words' and lemmatizing applied to text strings and set of unigrams generated for each question pairs. There were several questions where Unigram value was empty.

Those question pair was simply removed from data set.

Table 5-2 : Descriptive statistics of question pairs

	No. of Question Pairs
Has zero common words	42,398
Has At least one common word	361,558

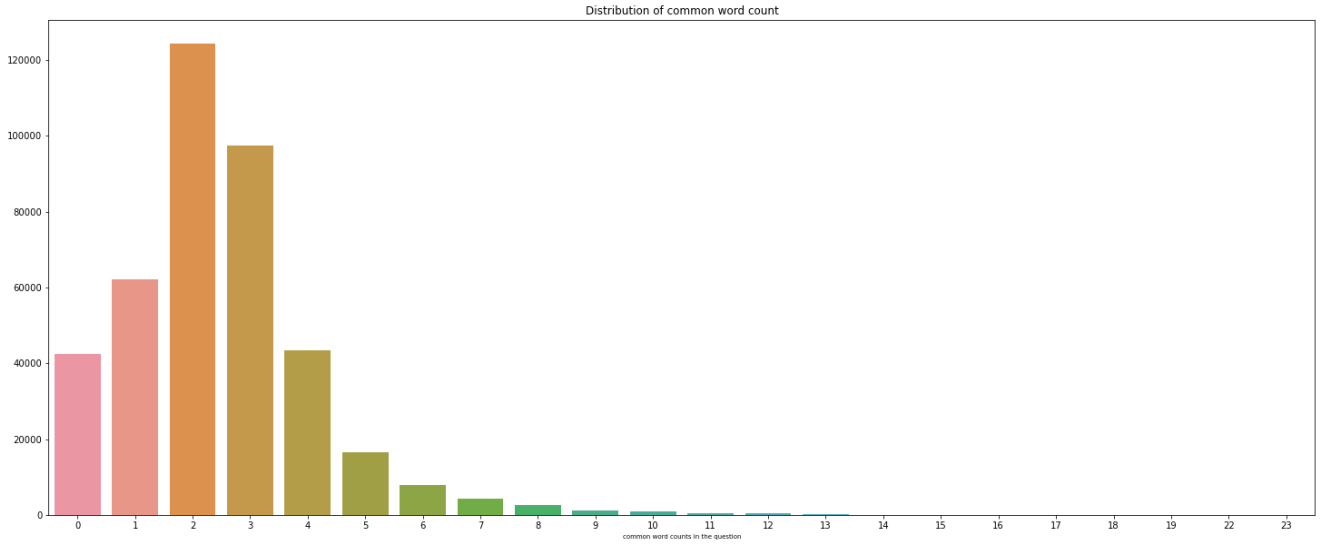


Figure 5-3 : Distribution of common words in each question

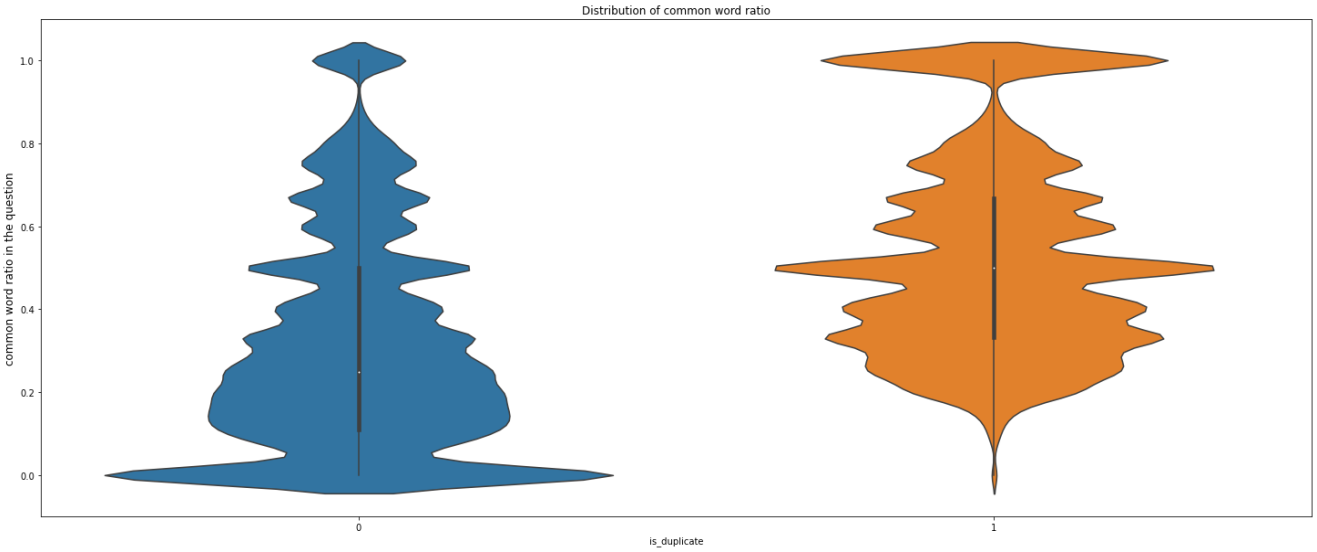


Figure 5-4 : Common words ratio for duplicate and non-duplicate question.

6. TRAIN AND TEST DATA SET CONVERSION

Since the dataset is skewed which means we have less than 50% of positive label, while splitting dataset to create train and test dataset, balance of label is maintained by specifying 'Stratify' argument. Other than that dataset is distributed randomly and blindly. Which means, since some questions are repeated frequently, it is possible that they will both appear in test and train dataset. Current analysis, ignore this fact and split data randomly by stratifying labels.

7. FEATURE SELECTION

7.1 WORD EMBEDDING

Word embedding is a well-known natural language processing technique to map words and phrases to vectors of real numbers. In other words, it is a mathematical embedding from one dimension per word to a continuous vector space. From this work, it is much easier for people to find synonym of one word using vectors.

Word embedding has been proved to boost the performance in NLP problem such as syntactic parsing and sentiment analysis. During research, it is found that GloVe (Global Vectors for Word Representation) performs excellent in this area [5]. It is an unsupervised deep learning algorithm for obtaining vector representations for words. There are some pre-trained GloVe word vectors available online. For the purpose of this analysis "*glove.6B.300d*" is used.

7.2 WORD SHARING

Apart from word embedding, word share feature is implemented. First I import a set of stop words from nltk.corpus. The stop words refer to the most common words in language such as 'and', 'also' and 'to'. This kind of words should be filtered out before processing. After removing these common words, each question pair is compared and the common words exist in both questions was found.

Figure 2 shows that, in the training set, which the label is already given, it seems powerful enough to label question pairs as non-duplicate if the percentage of shared word is low. But it is not good enough at identifying questions which are duplicate.

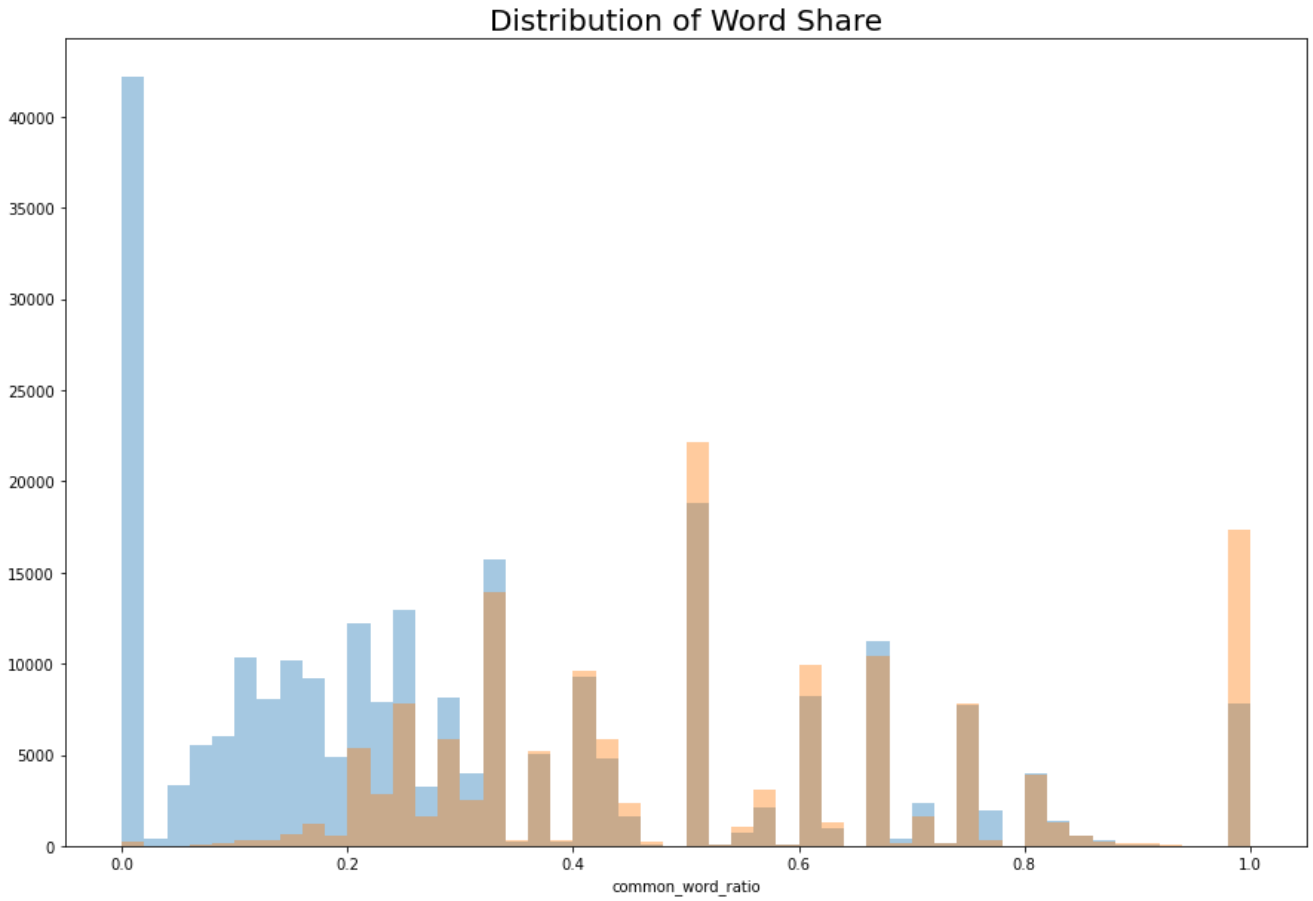


Figure 7-1 : Common word ratio for duplicate and non-duplicate questions

7.3 FUZZY SIMILARITY SCORE

There is a library called fuzzywuzzy , it is for string matching and depends only on the 'difflib' python library. It uses Levenshtein distance to calculate the differences between two sequences. Several ratios generated using this library and used as features.

- Fuzz Ratio

- Fuzz Partial Ratio
- Fuzz Sort ratio
- Fuzz token ratio

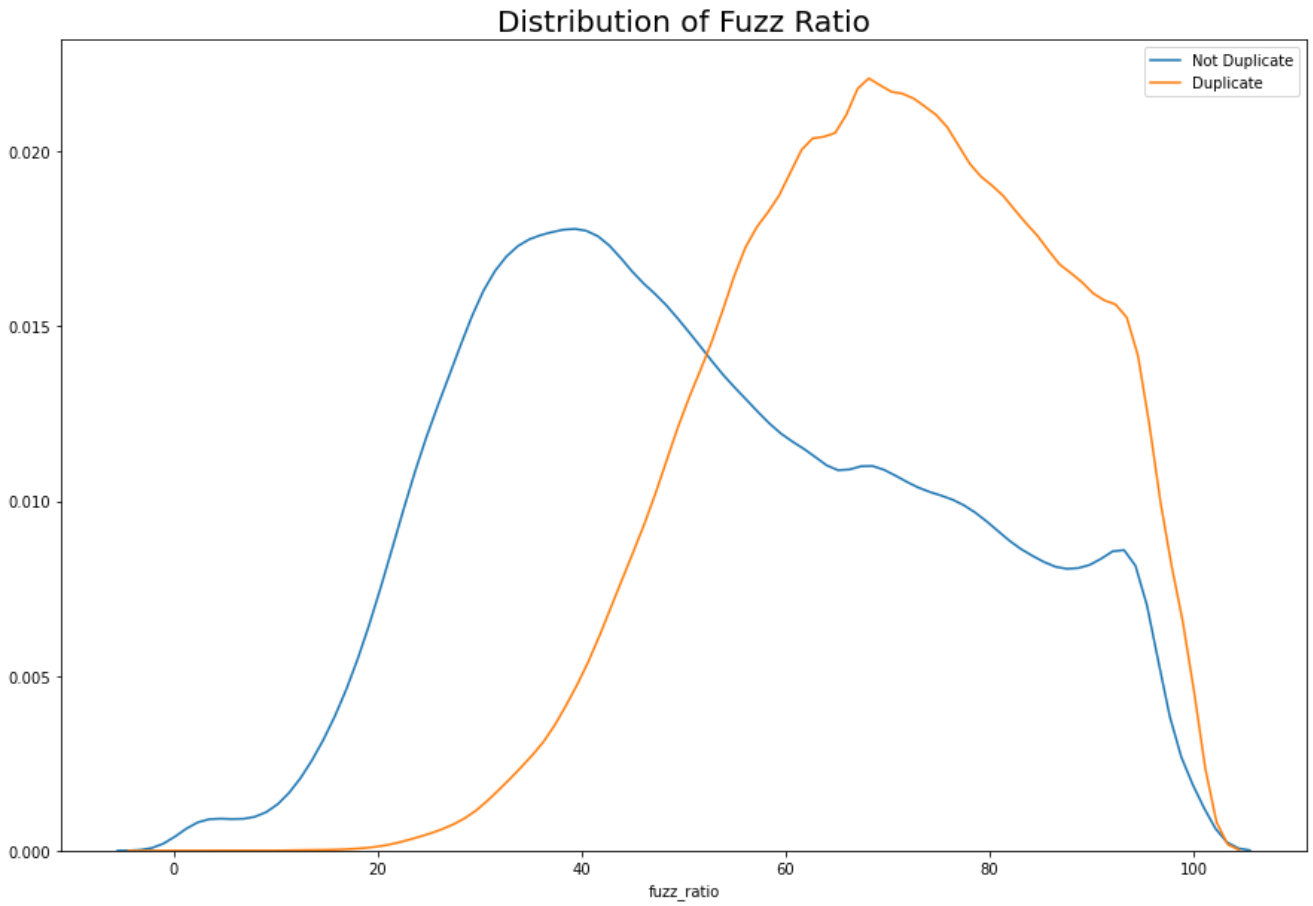


Figure 7-2 – Distribution of fuzz ratio (Similarity) for Duplicate and Non-Duplicate questions

8. MACHINE LEARNING PROCESS

There have been many studies on classification models predicting similarity between strings. Our classification goal is to predict which class the question pairs belongs to either **Duplicate** or **Not Duplicate**. In the following sections, we will share and discuss our experiments using Logistic Regression, Gradient Boosting and Random Forest for classification problem.

We also measure precision, recall, f1-score (the harmonic mean of precision and recall) and weighted average as defined below.

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

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - Score = \frac{2TP}{2TP + FP + FN}$$

Support = the number of true instances for each label Weighted-avg metric = metric weighted by support

Using features mentioned in Section 7, a scaled model split into test and train set. Ratio of 70%-30% selected and since, data labels are not balanced, stratified split is performed on data set so ratio of train and test data set is maintained.

8.1 LOGISTIC REGRESSION

In the field of machine learning, the goal of statistical classification is to use an object's characteristics to identify which class (or group) it belongs to. A linear classifier achieves this by making a classification decision based on the value of a linear combination of the characteristics. An object's characteristics are also known as feature values and are typically presented to the machine in a vector called a feature vector. Such classifiers work well for practical problems such as document classification, and more generally for problems with many variables (features), reaching accuracy levels comparable to non-linear classifiers while taking less time to train and use.

Logistic Regression takes in a list of features as input and outputs the Sigmoid of a linear combination of features weighted by learned parameters θ,

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta T} x}$$

To derive optimal parameters, the model iteratively updates weights by minimizing the negative log likelihood with L2 regularization

$$-\sum_{i=1}^{m}(y^i \log h_\theta(x^i) + (1 - y^i)\log(1 - h_\theta(x^i))) + \lambda\|2\|_2^2$$

After running Logistic Regression with the above setting for a maximum of 1000 iterations, we arrived at the following results:

Table 8-1 TRAINING SET RESULT

	Predicted Not Duplicate	**Predicted Duplicate**
Actual Not Duplicate	57614	13794
Actual Duplicate	19824	21875

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Not Duplicate	0.74	0.81	0.77	71408
Duplicate	0.61	0.52	0.57	41699
Weighted Avg	0.7	0.7	0.7	113107

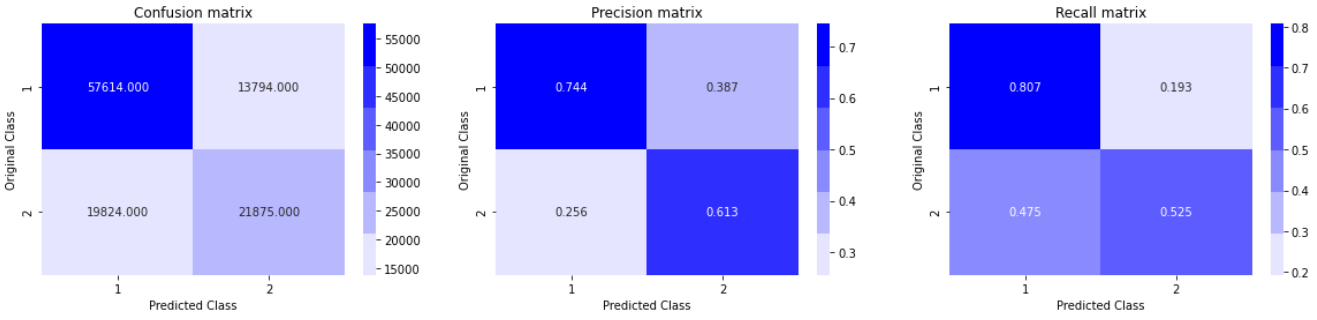
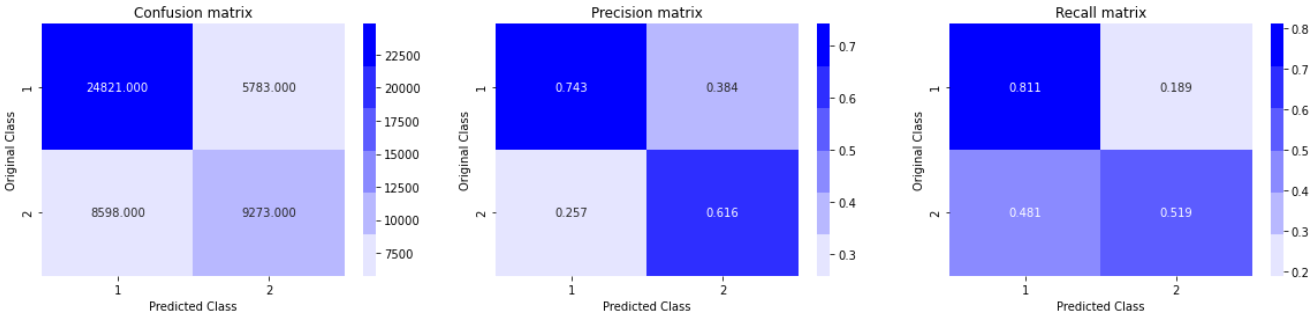


Table 8-2 TESTING SET RESULT

	Predicted Not Duplicate	Predicted Duplicate
Actual Not Duplicate	24,821	5,788
Actual Duplicate	8,598	9,273

Class	Precision	Recall	F1-Score	Support
Not Duplicate	0.74	0.81	0.78	30604
Duplicate	0.62	0.52	0.56	17871
Weighted Avg	0.7	0.7	0.7	48475



8.2 GRADIENT BOOSTING

Gradient boosting classifiers are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model. Decision trees are usually used when doing gradient boosting. Gradient boosting models are becoming popular because of their effectiveness at classifying complex datasets and have recently been used to win many Kaggle data science competitions.

The Python machine learning library Scikit-Learn, supports different implementations of gradient boosting classifiers, including XGBoost.

Classification algorithms frequently use logarithmic loss, while regression algorithms can use squared errors. Gradient boosting systems don't have to derive a new loss function every time the boosting algorithm is added, rather any differentiable loss function can be applied to the system.

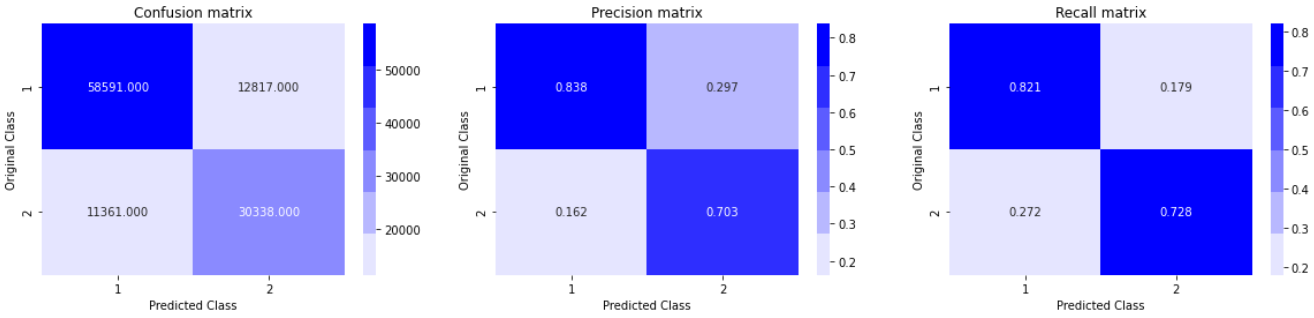


Table 8-3 TRAINING SET RESULT

	Predicted Not Duplicate	Predicted Duplicate
Actual Not Duplicate	58591	12817
Actual Duplicate	11361	30338

Class	Precision	Recall	F1-Score	Support
Not Duplicate	0.84	0.82	0.83	71408
Duplicate	0.7	0.73	0.72	41699
Weighted Avg	0.79	0.79	0.79	113107

Because the predictions of each tree are summed together, the contributions of the trees can be inhibited or slowed down using a technique called shrinkage. A "learning rate" is adjusted, and when the learning rate is reduced more trees must be added to the model. This makes it so that the model needs longer to train.

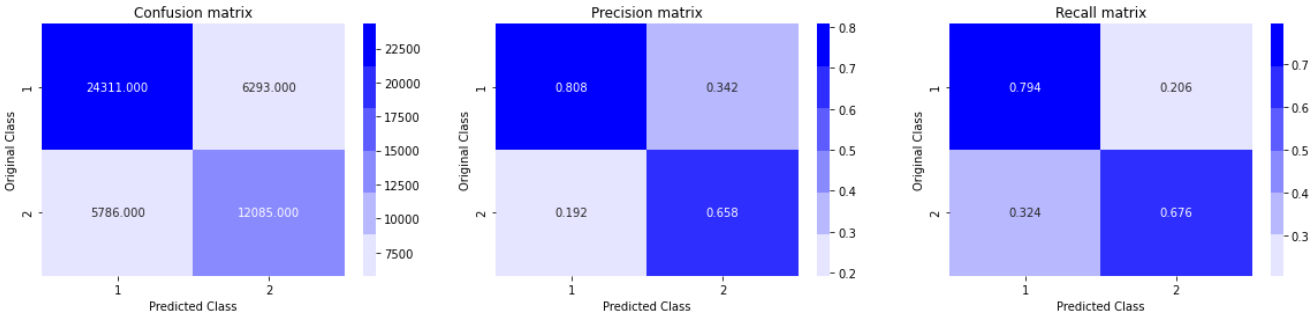


Table 8-4 TESTING SET RESULT

	Predicted Not Duplicate	Predicted Duplicate
Actual Not Duplicate	24,311	6,293
Actual Duplicate	5,786	12,085

Class	Precision	Recall	F1-Score	Support
Not Duplicate	0.81	0.79	0.8	30604
Duplicate	0.66	0.68	0.67	17871
Weighted Avg	0.75	0.75	0.75	48475

Compared to Logistic regression mode, Gradient Boosting model achieves higher accuracy as well as higher precision.

8.3 RANDOM FOREST TREE CLASSIFIER

Random Forest classifier is one of the tree ensemble methods that make decision splits using a random subset of features and combine the output of multiple weak classifiers to derive a strong classifier of lower variance at the cost of higher bias.

Best model uses 50 number of estimators and maximum 25 features.

$$1 - \sum_{j=0}^{1} p_j^2.$$

Decision splits are based on at most 50 features to reduce variance. After training, we reached the following result:

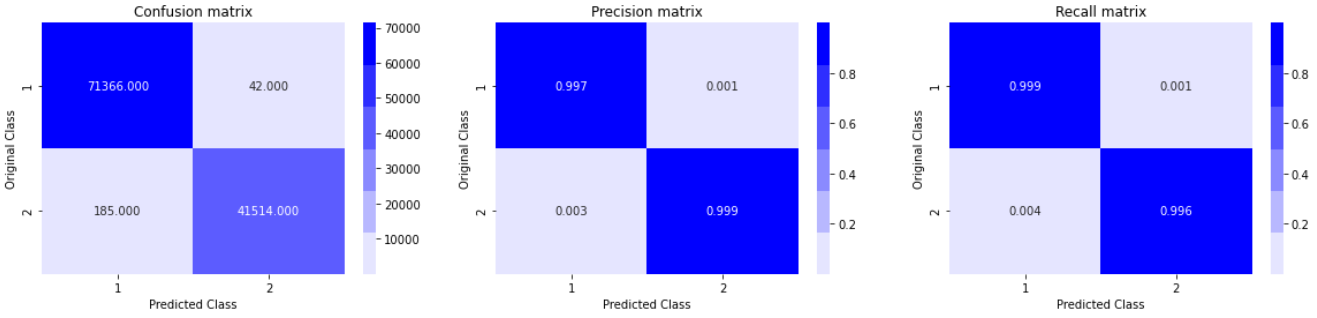


Table 8-5 TRAINING SET RESULT

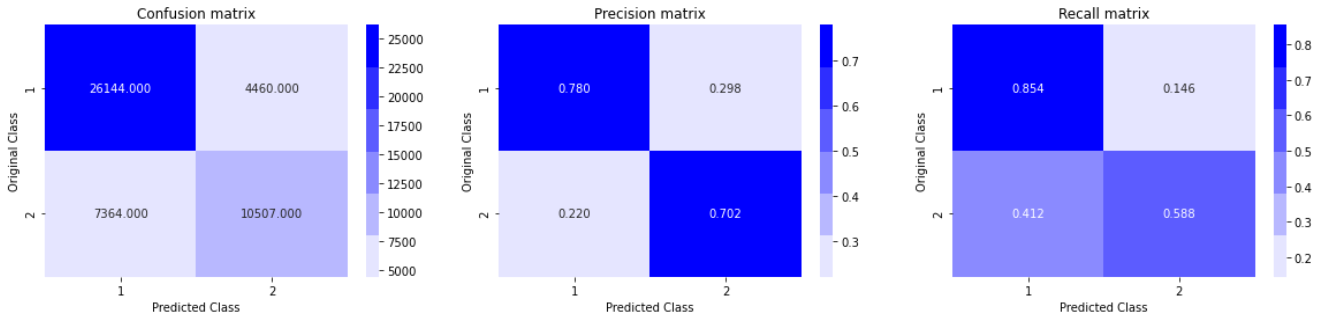
	Predicted Not Duplicate	Predicted Duplicate
Actual Not Duplicate	71366	42
Actual Duplicate	185	41514

Class	Precision	Recall	F1-Score	Support
Not Duplicate	1	1	1	71408
Duplicate	1	1	1	41699
Weighted Avg	1	1	1	113107

Table 8-6 TESTING SET RESULT

	Predicted Not Duplicate	Predicted Duplicate
Actual Not Duplicate	26227	4377
Actual Duplicate	6759	11112

Class	Precision	Recall	F1-Score	Support
Not Duplicate	0.8	0.86	0.82	30604
Duplicate	0.72	0.62	0.67	17871
Weighted Avg	0.77	0.77	0.77	48475



Although the performance is on par with Neural Network and Logistic Regression, Random Forest's overfitting problem is much more prominent than any other models even after restricting the maximum number of features considered for decision splits to 50.

9. CONCLUSION

Pairs of Quora question pair is analyzed for 3 different models. Logistic regression model gives lowest accuracy of 0.7 whereas Random Forest Classifier achieves accuracy of 0.78. There several features where created, and all the created features were used for all three models.

It is possible that different models achieve better accuracy with different combination of features. However, due to limiting time constraint that scenario is not analyzed.

Also, it has been seen that using Full data set (500,000) questions can achieve higher slightly higher accuracy (0.82) However, it is not performed in detail due to resource limitations.

Hence, there is future scope to improve this model by using

a) Full dataset
b) Combination of features.