Machine Learning Engineer Capstone Project Report

I. Definition:

Project Overview:

Quora is a place to gain and share knowledge—about anything. It's a platform to ask questions and connect with people who contribute unique insights and quality 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 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:

The goal of this project is to predict whether question pairs are duplicate or not i.e predict if the question pairs are of the same meaning. I am going to use NLP techniques for this problem. Doing so will make it easier to find high quality answers to questions resulting in an improved experience for Quora writers, seekers, and readers.

Metrics:

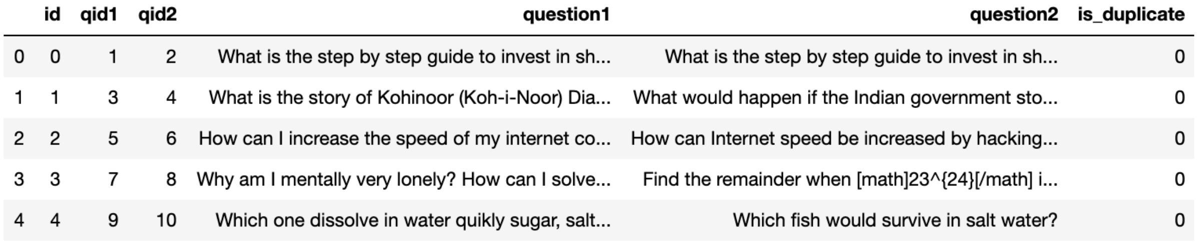
Predicted results are evaluated based on the log loss between the predicted values and ground truth. The ground truth is the set of labels that have been supplied by human experts. The ground truth labels are inherently subjective, as the true meaning of sentences can never be known with certainty. Human labeling is also a 'noisy' process, and reasonable people will disagree. As a result, the ground truth labels on this dataset should be taken to be 'informed' but not 100% accurate, and may include incorrect labeling. We believe the labels, on the whole, to represent a reasonable consensus, but this may often not be true on a case by case basis for individual items in the dataset. Log loss takes into account of the uncertainty of your prediction based on how much it varies from the actual label. This gives us a more nuanced view into the performance of our model. Lower the log loss better the model performs.

II. Analysis:

The datasets are provided by Quora on Kaggle.
Dataset Link: https://www.kaggle.com/c/quora-question-pairs/data
Input Data fields:

- id - the id of a training set question pair
- qid1, qid2 - unique ids of each question (only available in train.csv)
- question1, question2 - the full text of each question
- is_duplicate - the target variable, set to 1 if question1 and question2 have essentially t he same meaning, and 0 otherwise.

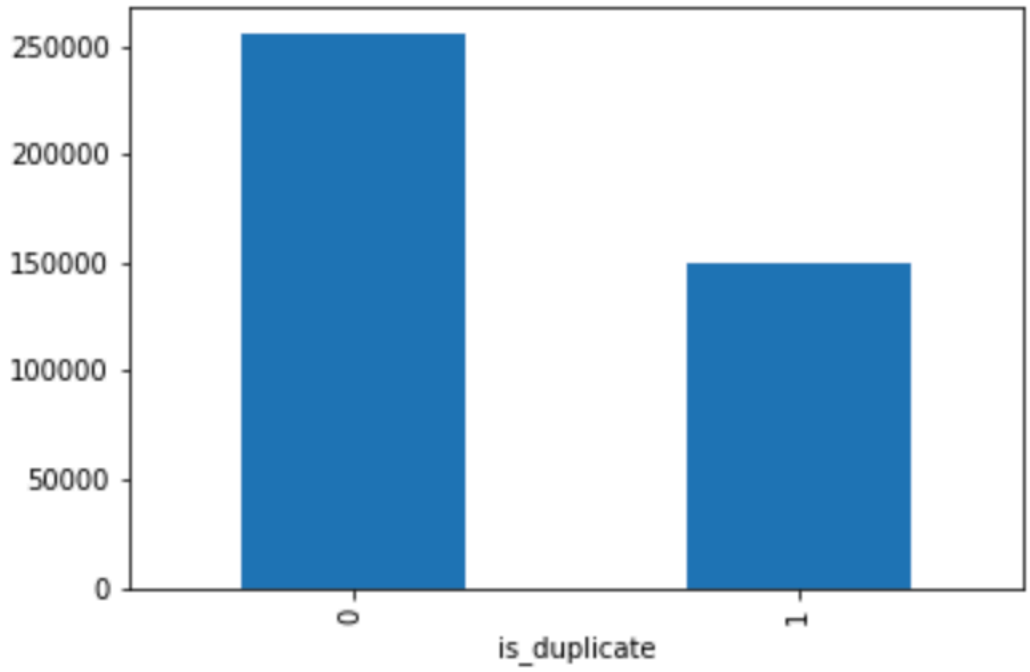
First 5 lines of the training data:



	id	qid1	qid2	question1	question2	is_duplicate
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	3	4	What is the story of Kohinoor (Koh-i-Noor) Dia...	What would happen if the Indian government sto...	0
2	2	5	6	How can I increase the speed of my internet co...	How can Internet speed be increased by hacking...	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
4	4	9	10	Which one dissolve in water quikly sugar, salt...	Which fish would survive in salt water?	0

Data Distribution in the training data:

Size: 63.4 MB
No of records: 2425740
Total number of questions: 537933
Number of questions appearing multiple times: 111780
Features: id, qid1, qid2, question1, question2
Label: is_duplicate(0 or 1)
Label distribution:
0 255027 (63.07%)
1 149263 (36.91%)

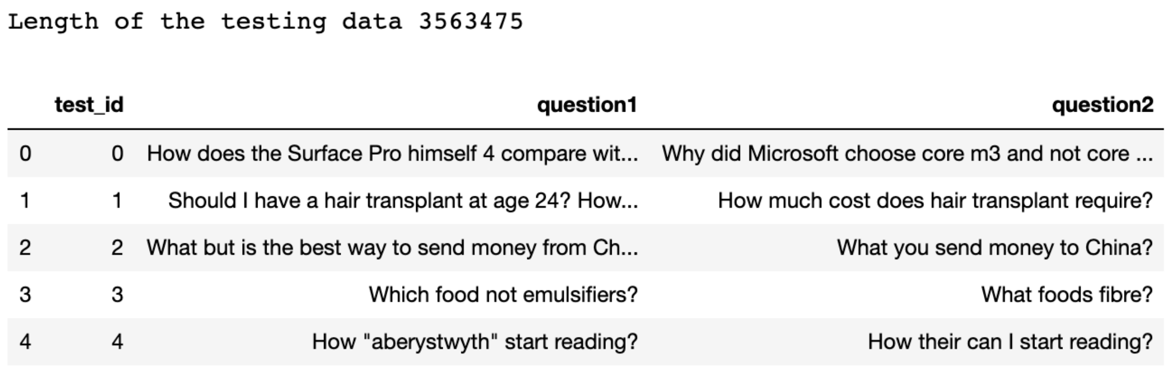


Test Data:

Size: 477.6 MB

No of records: 3563475

Features: test_id, question1, question2

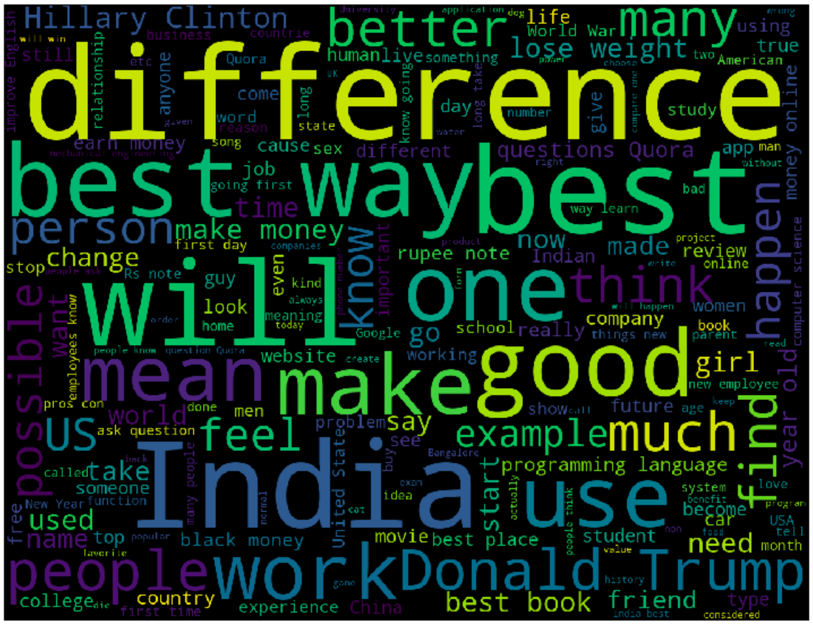


```
Length of the testing data 3563475
```

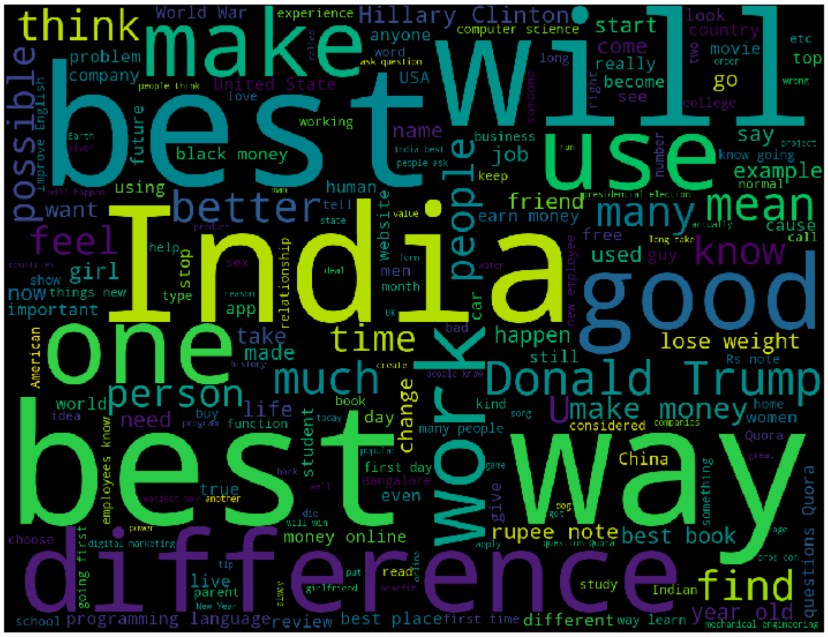
	test_id	question1	question2
0	0	How does the Surface Pro himself 4 compare wit...	Why did Microsoft choose core m3 and not core ...
1	1	Should I have a hair transplant at age 24? How...	How much cost does hair transplant require?
2	2	What but is the best way to send money from Ch...	What you send money to China?
3	3	Which food not emulsifiers?	What foods fibre?
4	4	How "aberystwyth" start reading?	How their can I start reading?

Another way to get the sense of most common words is using the wordcloud library.

Quesiton1:



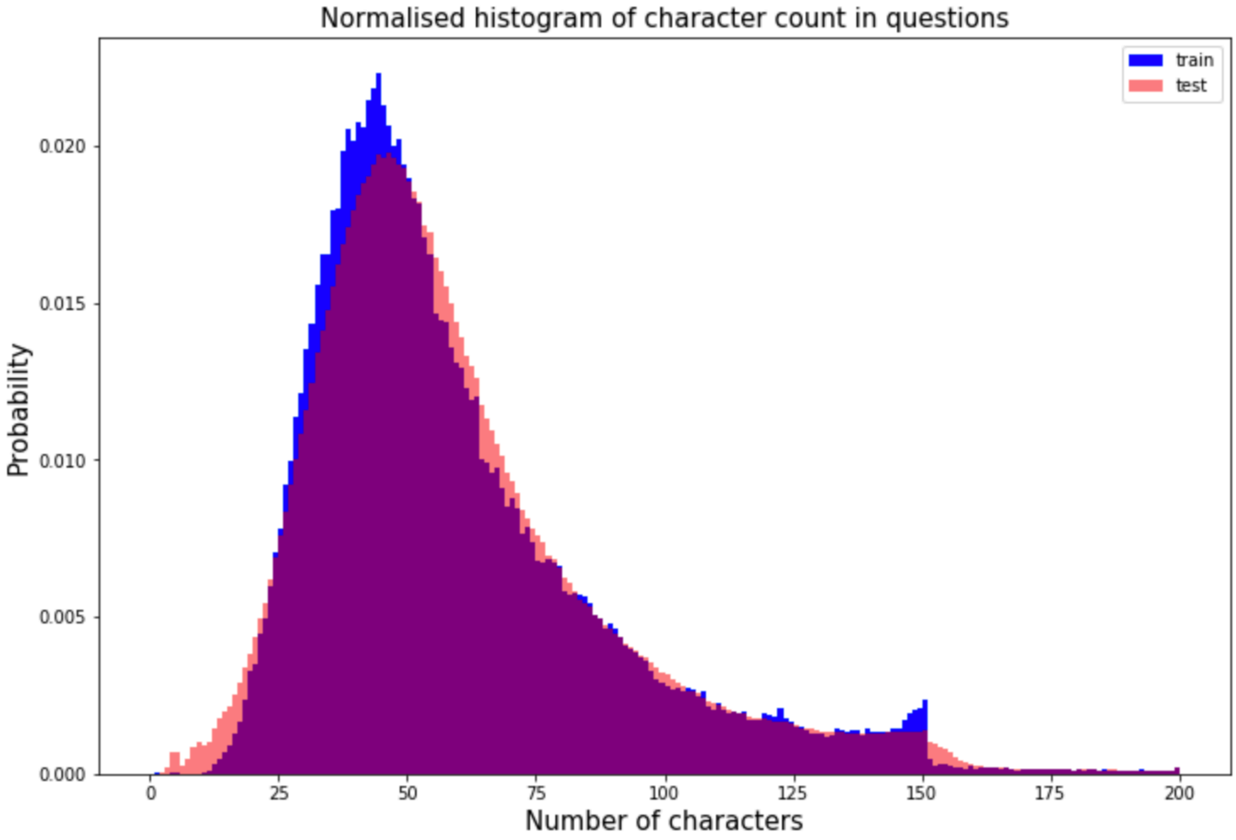
Question2:



After looking at the above two word cloud images we can see some words common in both question1 and question2 like best, difference, India, Trump, make etc…. From these words we can understand common topics in questions.

Exploratory Visualization:

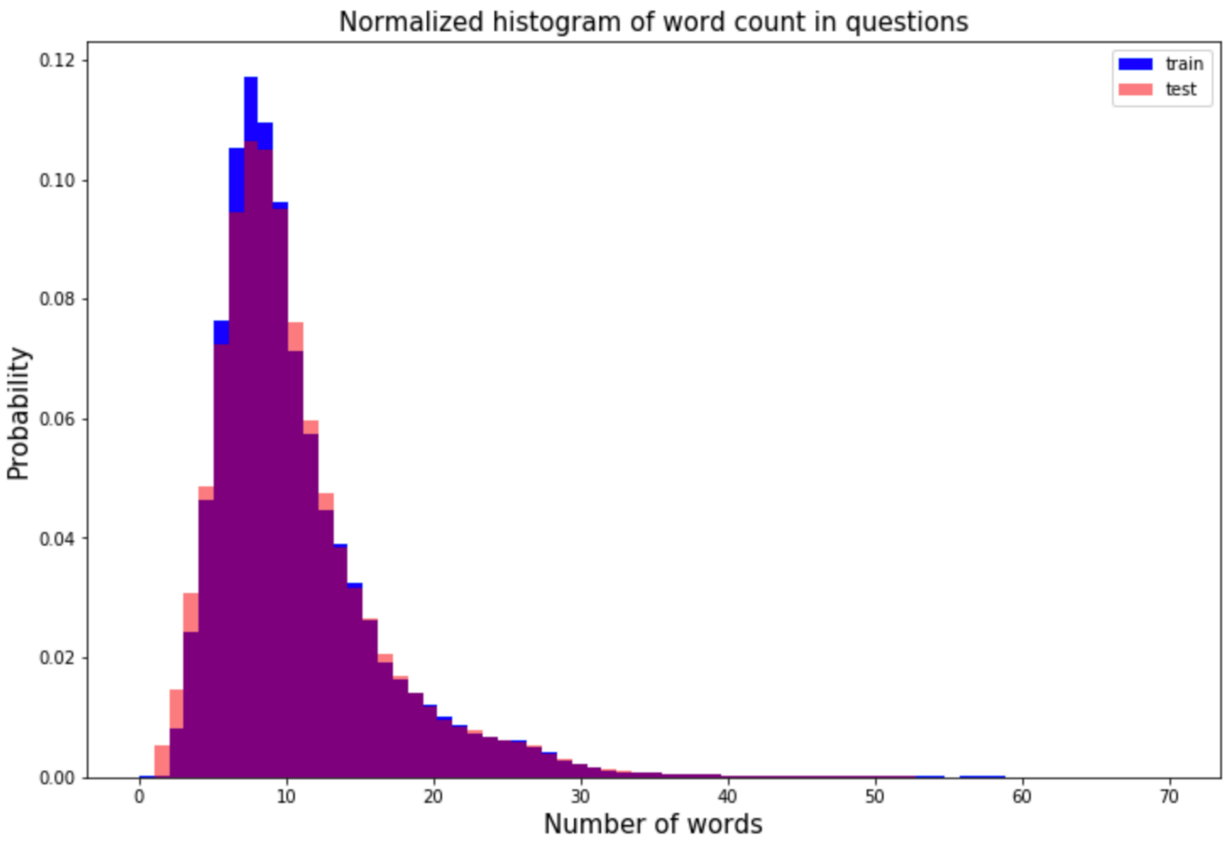
Normalized Histogram of character counts:



From the above plot, we can see most questions have anywhere from 15 to 150 characters. It looks like test data distribution is little different from the train one, but not too much different.

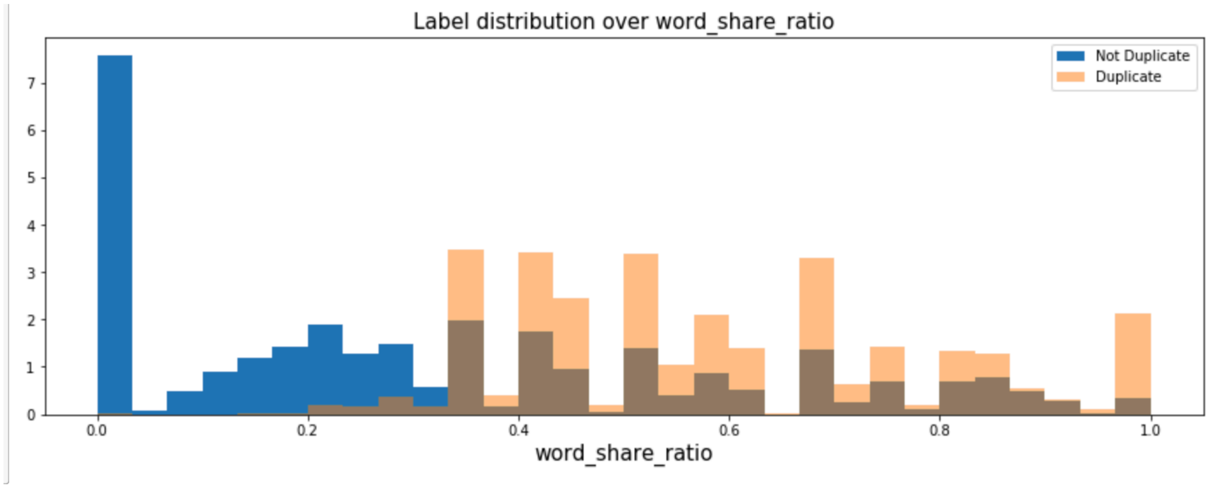
Another important feature in this plot is you can see characters getting count falls steep after 150. This could be due to some sort of quota question limit. This may be due to quora website limiting the amount of words we can type in.

Normalized Histogram of word count in questions:



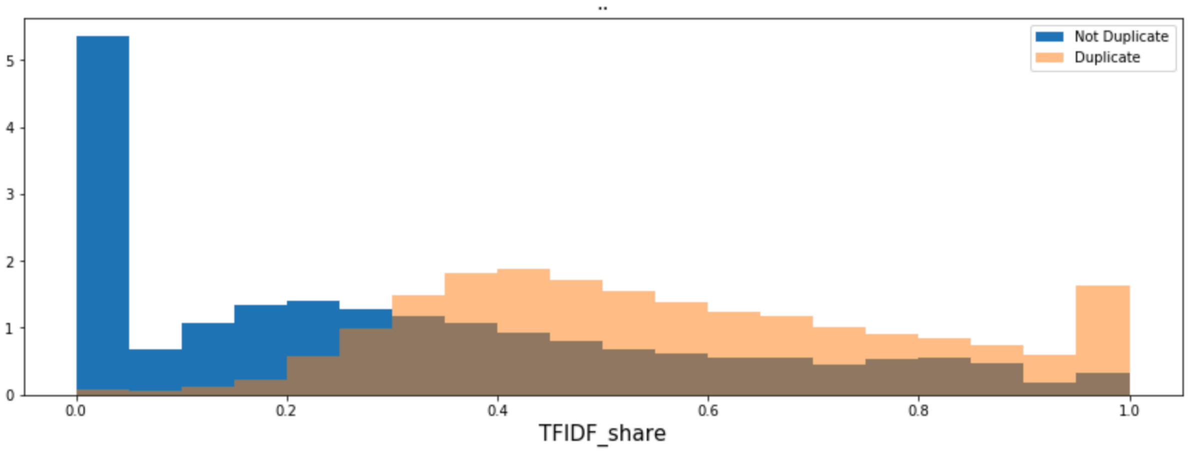
We can see a similar distribution for word count, with most questions being 10 word long. It also looks like the distribution of the training set seems more ‘pointy’, while the test set is wider. Nevertheless both train and test looks similar.

Label distribution over word_share_ratio:



Word share ratio is an extracted feature from the training data. The larger the value, more words are shared among the questions pair. In this bar plot, we can see the word share ratio for duplicate and non duplicate pairs. From the plot we can see that the duplicate questions generally have more questions between them.

Label distribution over TF-IDF share ratio:



TF-IDF share ratio is a feature extracted from the training data and will be discussed in the next section. TF-IDF share ratio plot is showing similar characteristics to the word share ratio.

Algorithms and Techniques:

This is a binary classification problem, and also supervised learning problem, I will be using supervised learning algorithms. First I will clean up the questions by removing leading and trailing spaces and also converting them to lowercase. The features extracted from the training data are: character count, word count, word_share_ratio, TF_IDF share ratio.

The algorithms to be used are:

1. Random Forest
2. Logistic Regression
3. Decision Trees
4. k nearest neighbors
5. Naive bayes
6. Support Vector Machine
7. Gradient Boosting

I will use the validation dataset to find out the best technique. Then I will do fine tuning on the best algorithm to get optimal accuracy.

Logistic Regression is a regression model where the outcome is binary and model predicts the probability of the outcome.

Decision tree algorithm tries to solve the problem using tree representation, each internal node correspond to the feature and the leaf corresponds to the label.

Random forest is the ensemble of decision trees and final prediction is made by the taking the mean or mode of the prediction of all trees.

K nearest neighbors algorithm uses the average of k nearest datapoint to predict the testing data value. The distance here is usually euclidean distance.

Naive babes is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. It assumes the presence of particular feature in the class is unrelated to the presence of any other feature.

Support Vector Machine is an algorithm which outputs an optimal hyperplane to separate the labeled training data.

Gradient boosting model is similar to random forest with the difference being how we train them. For Gradient boosting, we assign prediction score in each leaf instead of a binary value and during training we add one tree to ensemble at a time and do the optimizing.

Benchmark Model:

Benchmark model for this problem is the random forest classifier. There is no published result for this problem, but this is gaggle competition, I can compare the score on the leaderboard and understand my model performance.

III. Methodology:

Data Preprocessing:

First I stripped the leading and training spaces and then convert into lowercase letters. There are 6 features. Question1 number of words, Question2 number of words, Question1 character count, Question 2 character count, word share ratio, TF-IDF word share ratio.

Before measuring word share ration and TF-IDF word share ratio I removed the stop words from the questions. I dowloaded the stop words from the nltk library.

Word share ratio is calculated as the number of words appearing in both question1 and question2 divided by the total number of words in question1 and question2. This result in the normalized value if how much overlap between question1 and question2.

For calculating TF-IDF word share ratio, First I combine all questions into one big corpus, and then calculate weight for each word. The weight is calculated by the inverse of word count plus epsilon (=5000) such that the less common words get higher weights. I ignored words that only appear once in the whole corpus for the reason that it may be typos. Then I calculate the tfidf_word_share_norm using share_weight divided by total_weight. Share_weight is the total weight of the words appearing in both question1 and question2. Total_weight is the total weight of every words in either question1 or question2. The result is a normalized TFIDF_weight_share. Since TFIDF discounts on the common words that appear universally in our language, this measure represents more closely to how similar two questions are.

I tried using stemming of words to improve the performance but it was of no use, so I ended up not using that.

Implementation:

1. Logistic regression: I used GridSearch cross validation to find best parameters (‘C’, ‘penalty’), which are (‘1000’, ‘l2’).
2. Decision tree: For this one I just simple fit and predict functions of sklearn library.
3. Support Vector Machine: I did GridSearch cross validation to find best parameters (‘C’, ‘kernel’), which are (‘1000’, ‘linear’). I set max iteration to 500 to avoid “out of memory” issue.
4. K-Nearest Neighbors: I also did GridSearch cross validation to find best parameters (’n_neighbors’, ‘weights’), which are (‘8’, ‘uniform’).
5. Naive Bayes: Naive Bayes features work better if their values are integers. Since Naive Bayes is based on probability, there is no need for feature value normalization.
6. Gradient Boosting (XGBoost): My setting for parameters are {‘objective’: ‘binary:logistic’, ‘eval_metric’: ‘logloss’, ‘eta’: 0.02, ‘max_depth’: 4} for the beginning. I did 500 rounds of boosting and if there’s no improvement after 50 rounds the training will stop.

Refinement:

Based on the Kaggle leaderboard score for the 6 models, XGboost gives the best result. Therefore, I started playing around with ‘eta’ (learning rate) and ‘max_depth’ (model complexity).

Eta	0.01	0.02	0.1	0.2
Max_depth	3	4	5	6

The numbers I have tried are : I found that ‘eta’=0.2 and ‘max_depth’=6 gives me best result without overfitting. When my validation data has higher logloss than my training data, I know there is overfitting. After finding the optimal parameters, I did 1000 rounds of boosting to get my final logloss number, which is 0.39064.

IV: Results

Model Evaluation and Validation:

The final model I used is XGBoost(Gradient Boosting), which is similar to random forest but gives better performance for this problem. I used validation set so that model won't overfit for the training data. The model is robust compared to the random forest because it uses result from the previous training to improve on it, meanwhile random forest uses the best result among many trees. Kaggle competition score is based on the unseen data that are not accessible to anyone. I am able to get similar logloss in both by training data and test data. So, I am confident with the model.

Justification:

Final result is slightly better than the benchmark model. Although the idea behind the random forest and gradient boosting, the training process is different. In my project the benchmark model has the logloss of 0.55 and the XGBoost model has the logloss of 0.39. I believe the final result is good enough for this problem and 80% accuracy to enough for prevent the user from the wasting time on duplicate questions on quora platform.