Machine Learning Engineer
Nanodegree Program

Gaurav Baid
June 27th, 2018

Definition

Project Overview

With advancements in technology and a rate at which the data is consumed by individuals and companies today, the importance of data quality is getting more traction day by day. As the amount of data increases, it becomes very important for companies to make sure the data consumed by users is not duplicated or redundant which will in turn create a bad user experience. Companies nowadays are putting a lot time and resources to make sure the data quality is meeting the industry standards with respect to data deduplication, data quality and management

Problem Statement

As per the problem description stated in Kaggle :

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.

The goal of this competition is to predict which of the provided pairs of questions contain two questions with the same meaning. one of the solution used is Human labeling which is also a 'noisy' process.

Currently, Quora uses a Random Forest model to identify duplicate questions. The goal of the project is to devise and implement model by applying NLP techniques to classify whether question pairs are duplicates or not.

As stated in the proposal, The following steps will be taken while implementing the classification problem

1) Data cleaning and preprocessing

   This step is essential to clean the data set and making sure the data preprocessed to not contain any stop words, punctuations etc.

2) Creating new features

   Often times the data provided in training data set itself is not useful for the model building, hence new features needs to be created which can provide more valuable information during model building and implementation

3) Model building and Implementation

   As stated in the proposal we will use algorithms such as Random Forest, Logistics Regression as well as XGBoost to generate predictions

Metric

This problem statement will be evaluated based on the log loss function between predicted values and the ground truth. Ground truth are the set of predictions provided by human experts. Log loss is a suitable evaluation metric for this problem as it measures the accuracy of the classifier by considering the probabilities of the model and comparing it to the true labels. To increase the accuracy of the model, we need to minimize the log loss score on predictions. Rather than simply picking the most likely output as in accuracy metric and f1 score, log loss is used when evaluating a model which assigns a probability to outputs.

According to Sci-kit Learn, Log-Loss[6] is defined as :

This is the loss function used in (multinomial) logistic regression and extensions of it such as neural networks, defined as the negative log-likelihood of the true labels given a probabilistic classifier's predictions. The log loss is only defined for two or more labels. For a single sample with true label yt in {0,1} and estimated probability yp that yt = 1, the log loss is

$$-\log P(yt|yp) = -(yt \log(yp) + (1 - yt) \log(1 - yp))$$

Analysis

Data Exploration

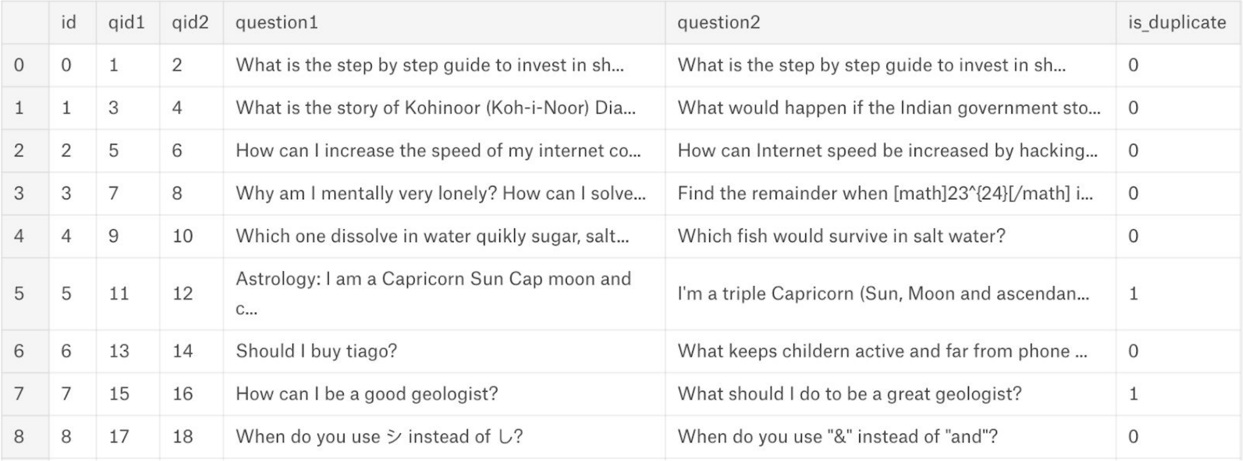
Kaggle provides us with training and testing data provided by Quora.

- 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 the same meaning, and 0 otherwise

on doing exploratory analysis, training data set contains 404290 with 6 features and testing data contains 2345796 with 3 features. Training data has 63% (255027) of non duplicates and 37% duplicate records. Also training data has 3 null values and testing data has 6 null values. The model has to be created using the free form text or free text available in question1 and question2 column of both training and testing data set.

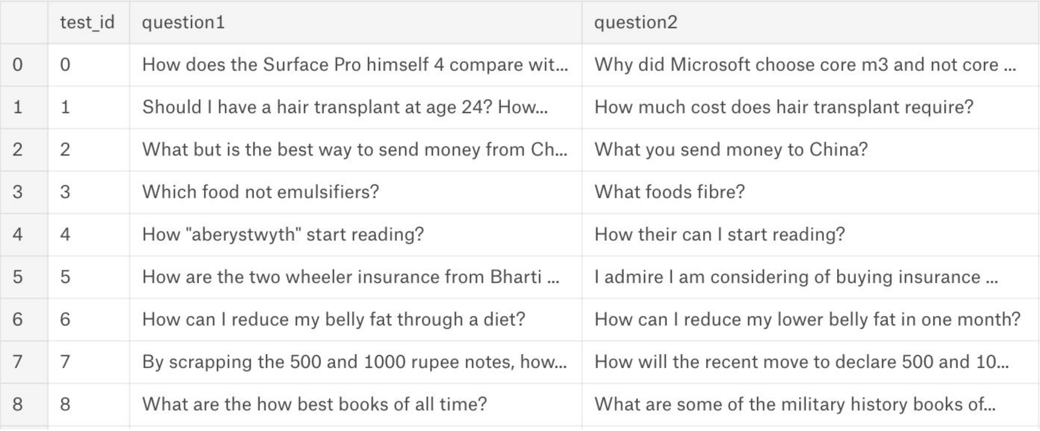
Below are the few records from the training and testing data set

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
5	5	11	12	Astrology: I am a Capricorn Sun Cap moon and c...	I'm a triple Capricorn (Sun, Moon and ascendan...	1
6	6	13	14	Should I buy tiago?	What keeps childern active and far from phone ...	0
7	7	15	16	How can I be a good geologist?	What should I do to be a great geologist?	1
8	8	17	18	When do you use シ instead of し?	When do you use "&" instead of "and"?	0

Testing Data:



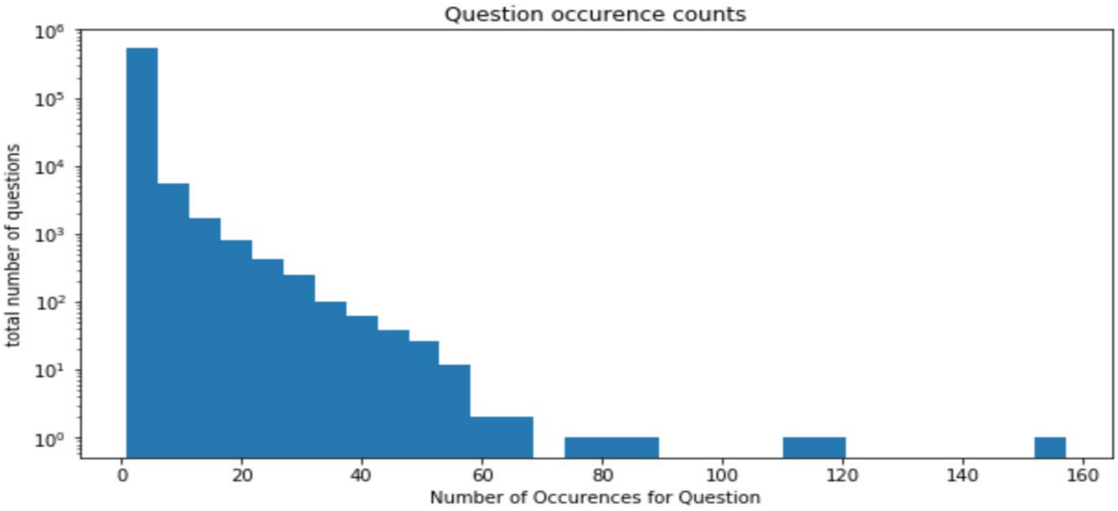
	test_id	question1	question2
0	0	How does the Surface Pro himself 4 compare wit...	Why did Microsoft choose core m3 and not core ...
1	1	Should I have a hair transplant at age 24? How...	How much cost does hair transplant require?
2	2	What but is the best way to send money from Ch...	What you send money to China?
3	3	Which food not emulsifiers?	What foods fibre?
4	4	How "aberystwyth" start reading?	How their can I start reading?
5	5	How are the two wheeler insurance from Bharti ...	I admire I am considering of buying insurance ...
6	6	How can I reduce my belly fat through a diet?	How can I reduce my lower belly fat in one month?
7	7	By scrapping the 500 and 1000 rupee notes, how...	How will the recent move to declare 500 and 10...
8	8	What are the how best books of all time?	What are some of the military history books of...

Exploratory Visualization

As the analysis has to be done on the full form text, we are limited to making visualizations directly on the text available in question1 and question2. New features like length of each question, difference in number of words, common words etc. can be generated using the text and and exploratory visualizations can be created on them

After doing initial exploration on the data, below observations were made

1. 111,780 questions appeared multiple times in training data out of 537933 unique questions
2. As seen in the below histogram, majority of the questions appeared more than once
3. There is a vast majority of questions that appeared less than 60 times
4. A very small portion of the questions appeared more than 100 times



Algorithms and Techniques

The following algorithms will be used for solving the problem:

Logistic Regression

As mentioned in the solution statement in the project proposal, I tested the data with logistic regression which is defined as :

Logistic regression or Logit is a modelling technique used for classification. Logistic regression is similar to linear regression in all aspects except, Logistic regression outputs predicted probabilities associated with the class, which is a better indicator of confidence in the given class.

As per Scikit learn [9] -
Logistic regression, despite its name, is a linear model for classification rather than regression. Logistic regression is also known in the literature as logit regression, maximum-entropy classification (MaxEnt) or the log-linear classifier. In this model, the probabilities describing the possible outcomes of a single trial are modeled using a logistic function.

There are different solvers for logistic regression like “liblinear”, “newton-cg”, “lbfgs”, “sag” and “saga”. I have decided to use ‘sag’ as it uses stochastic average gradient descent and works faster on larger data sets

XGBoost

Another technique used for classification is XGBoost. As explained in machinelearningmastery.com, XGBoost is an implementation of gradient boosted decision trees designed to achieve speed and performance. XGBoost uses boosting which is a ensemble technique where new models are created that predict the errors made by the existing models and then added together to output the final prediction.

Benchmark

Quora question pairs competition on Kaggle has leaderboard that has 3307 submission as of 06/20/2018 which displays the best log loss scores obtained by different teams or individuals. Currently Quora is using random forest method measured using log loss metric to identify duplicate questions. The current median score on the Kaggle leaderboard for the competition is ~0.36.

As log loss score is used to evaluate results of the problem, it makes more sense to use the same for benchmarking

Methodology

Data Preprocessing

After doing the exploratory analysis on the training and testing data set, the first step was to find the null values in the data set. The training data set had 3 and 6 null values in the training and testing data set respectively. The null values were treated by replacing them with 'NA' text.

The data preprocessing involved the following steps

- Convert data to lowercase.

  As the case of text does not provide any valuable information in itself, both question1 and question2 were converted to lower case for consistency. Also there was good probability that the algorithm might not match similar words because of different case for e.g Gradient BOOSTING and gradient boosting and hence the conversion. `lower()` function was used to convert the questions to lowercase.

- Removing stop words and punctuations [1]

  After converting the text to lowercase, stop words and punctuations were removed to ensure the algorithm can easily identify common words. Stop words are words like 'a', 'the', 'is' , 'are' etc which appear frequently in the data but may not provide any valuable of contextual information. Standard NLP involves removing the stops words and punctuations as they are considered irrelevant for the analysis. This step was achieved by using remove_stop_words( ) and remove_punctuation( ) functions.

- Perform stemming [2]

  Later stemming was performed on the questions using porter stemmer. As per nltk.org, stemmer Stemmers remove morphological affixes from words, leaving only the word stem. Stemming is done to identify words with same intended meaning by converting the words to its roots.

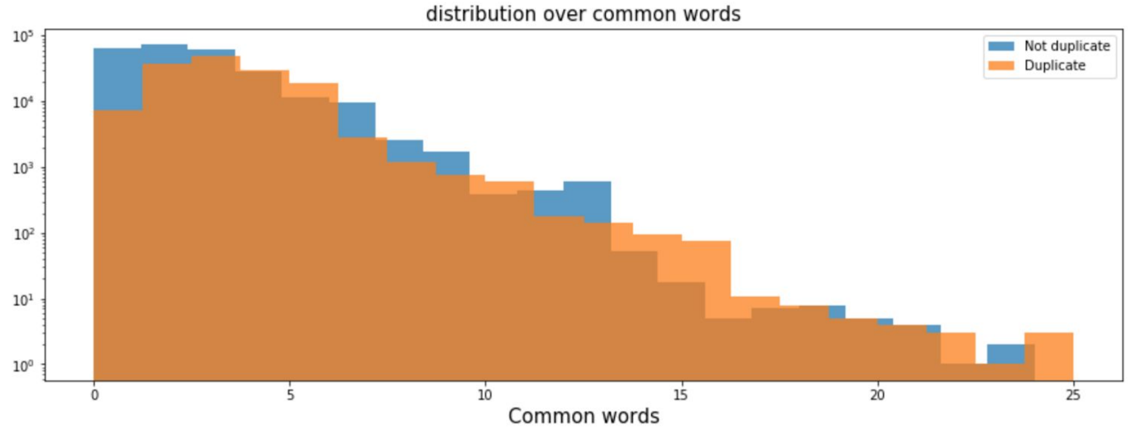
Feature Engineering

As defined by machinelearningmastery.com [3], Feature engineering is defined as a process to create new features from raw data, that better represents the underlying problem to the predictive models, increasing the accuracy on testing data.

The following features were created and tested in different iterations on model implementation to minimize the log loss score.

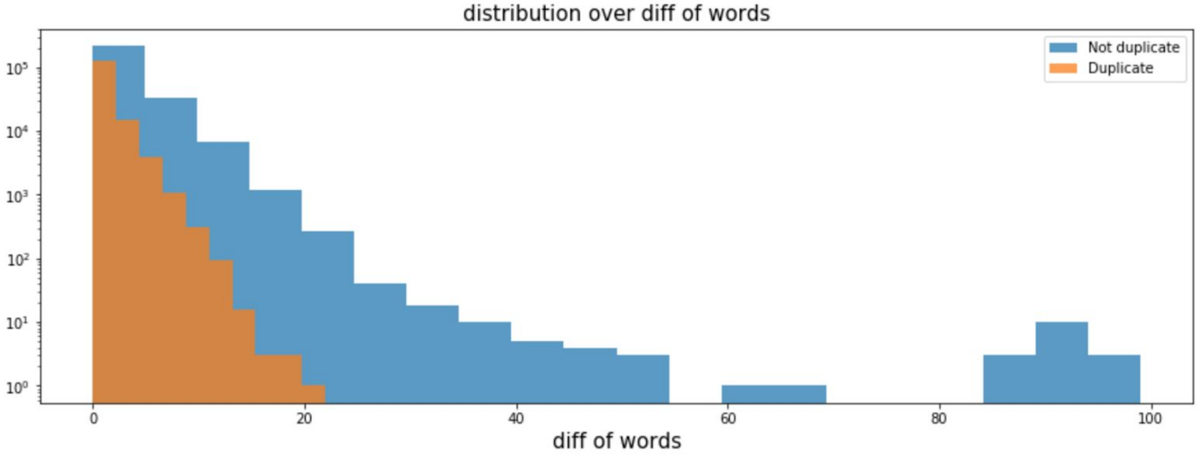
Number of Common words

One of the most common features is comparing the number of common words between two sentences to predict the similarity of the sentences. This was achieved by the common_words ( ) function in the implementation. If the two sentences have a lot of common words it is more likely that they are similar. However the problem with the above assumption is that not all the sentences are of similar length, Thus it is very difficult to identify duplicate questions for non duplicate as shown in the below diagram. Another feature that will provide more information as compared to number of common words is the ratio of common words.



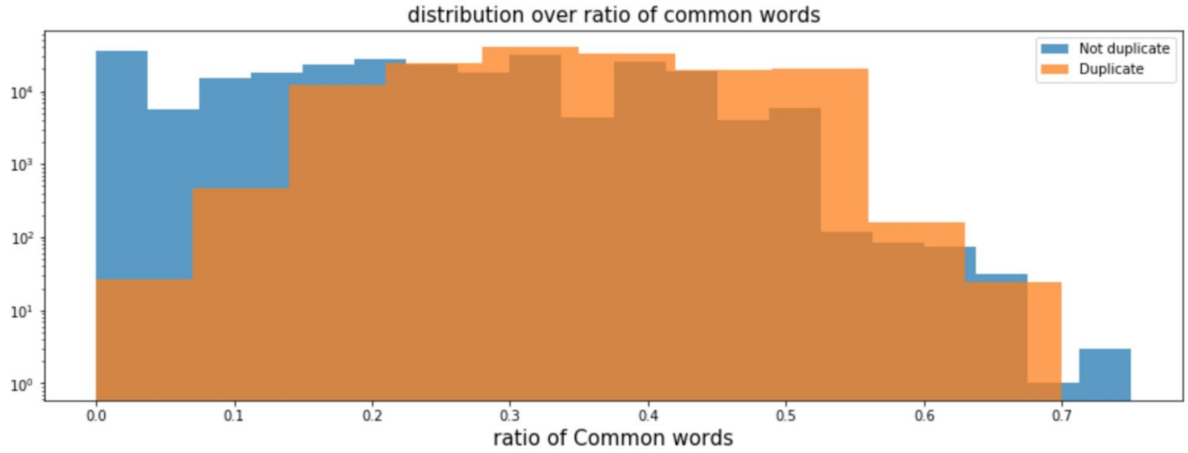
Difference of words

As during the iterations, the number of common words between two sentences to predict the similarity of the sentences was not predictable enough, we calculated absolute difference of length of two questions. This was achieved by the diff_words ( ) function in the implementation. As seen from the diagram below, this feature does a better job of identifying the duplicates from non duplicates as compared to number of common words



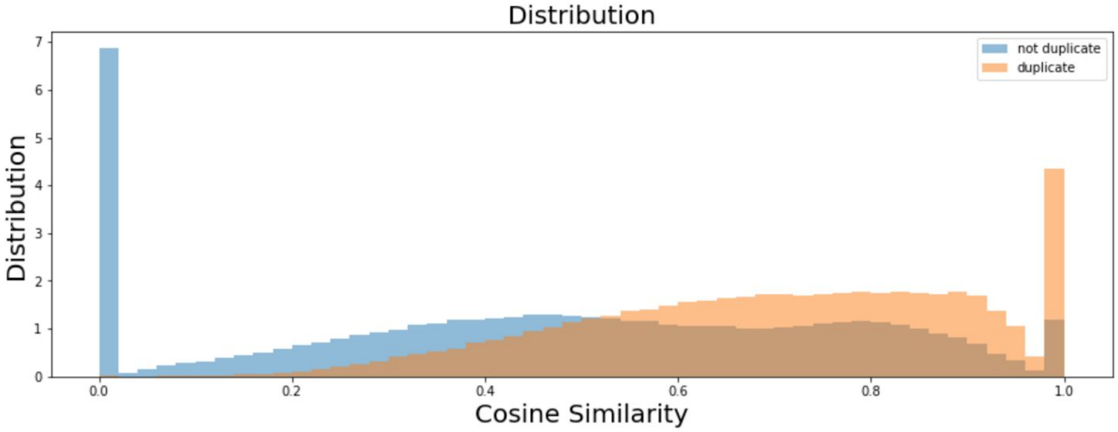
Ratio of common words

Ratio of common words feature takes into account the length of each sentence which was missing from the number of common words feature by dividing the number of common words by the length of both the questions into consideration. Ratio of common words feature was calculated by ratio_common_words function. The importance of the ratio of common words feature has improved as compared to the number of common words as show in the diagram below.



Tf-Idf Vectorizer and Cosine SImilarity

One of the widely used method in natural language processing is TF-IDF, its a numerical statistics that is intended to reflect how important is a word to the document in a collection or corpus. TF idf vectorizer is used to extract vector scores on words in the questions and then cosine similarity is calculated to for each pair of questions.



In addition to existing features, Additional features like length of question1, length of question 2, number of words in question 1 and number of words question 2 were also tried and tested.

Implementation

After data preprocessing which included

- removing the ‘id', ‘qid1' and ‘qid2' from training set as they do not provide valuable information
- Using empty string to fill in with ‘NA’ text values
- Cleaning the text by converting the questions to lowercase
- Removing the stop words and punctuations
- Using stemming to converts the words to its roots
- Creating additional features like
    - Difference in length of words
    - Number of common words
    - Ratio of Common Words
    - Cosine SImilarity

After the features were developed, The data was scaled to make sure the regression algorithm works seamlessly on the data before building the model. Min_max_scaler was used to scale the data function from the SciKit-learn.

The first step in building the model was to split the training data set into training and validation set. The data was split into training and testing to make sure there is no bias during the selection of data. For this exercise, training data was split into 80% training data and 20% validation data set by using train_test_split function from the SciKit-learn. Random forest, Logistic regression and XGBoost algorithms were used for the classification.

The prediction rate of training data set was identified to be approximately 37% during the exploratory analysis of the training data.

Random forest classifier was able to output a log loss score of ~0.533 without any parameter tuning and with n_estimators set to 10 and using difference in common words and ratio of common words feature created using existing data.

Logistic regression with GridSearch was used with both training and testing data set for predictions. Logistic regression did improve the log loss score as compared to Random Forest with new features like difference in words, cosine similarity and ratio of common words implemented and after removing the stop words, punctuations and using stemming. ‘SAG’ solver was used in the implementation which guarantees fast conversion on the features with same scale

XGBoost was also implemented in addition to to random forest and logistic regression algorithm. More information on XGboost was provided under algorithms and techniques.After tuning of different parameters and adding different features for random forest classifier, logistic regression and XGBoost, XGBoost gave the best results with the least log loss score of ~0.49, which is less than the median score for the leaderboard on the Kaggle but still good to identify duplicates and non duplicates. The final predictions on the testing data set will be uploaded to kaggle board using csv file to check for the log loss score.

Refinement

A lot of iterations on the data set using combination of data preprocessing and feature addition were performed during the refinement phase. Different models were trained by making adjustments to data preprocessing steps by adding and removing different features created during feature engineering phase.

Random forest classifier was able to output a log loss score of 0.533 without any parameter tuning and with n_estimators set to 10 and using difference in common words and ratio of common words feature created using existing data. In the later stages, different parameters were tuned and a combination of features and parameters were used for of all the three algorithms (Random forest, Logistic Regression and XGboost).

After finalizing the features and processing steps, logistic regression gave a log loss score of ~0.56 and XGboost was able to improve the log loss score to ~0.49. To achieve this result following new features were added in addition to existing ones

- Length of question 1
- Length of question 2
- Number of words in question 1
- Number of words in question 2

Parameters for XGboost

- max_depth = 8
- eval_metric = 'logloss'
- eta = 0.4
- objective = 'binary:logistic'

Model Evaluation and Validation

XGboost model was able to achieve the best results for the given problem i.e the model was able to make better predictions of whether the questions pairs are duplicate or not as compared to benchmark and logistic regression model. All the three algorithms were tested with different random data split for training and testing data sets to make sure it generalizes on different buckets.

Xgboost was able to achieve log loss score of ~0.49 on the training data set .The numerous iterations on the building model are satisfactory as the log loss score dropped significantly through those iterations. As stated in the benchmark section of the report the median log loss score form the leaderboard on the kaggle was ~0.36 and the score obtained from the implementation though not less is still closer. Overall, this model can definitely be used to identify duplicates and non duplicates with a good accuracy but a lot of improvements are still required.

Justification

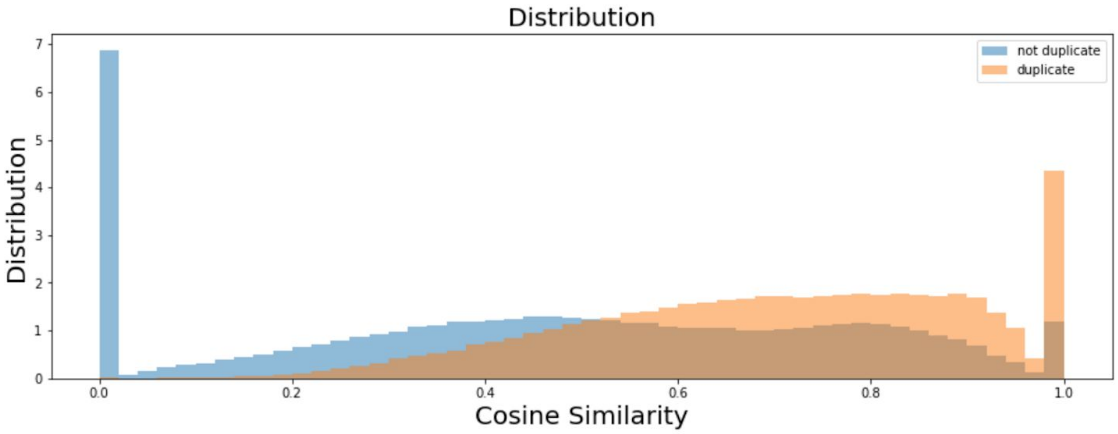
The benchmark used for the project was based on the random forest classifier. As stated earlier the expectation was to obtain a log loss score of ~0.36 as it was the median of all the scores submitted by 3007 participants in the competition. The random forest classifier used for the benchmark gave a log loss score of 0.56 without any data preprocessing and parameter tuning. However, XGBoost gave the best log loss score of ~0.49 which is a good improvement over logistic regression. Though the log loss score is not close enough to expectation, Future improvements to the model and features can be made to improve it.

The model is doing a good job of predicting whether the question pairs are duplicate or not and thus serves the purpose. Although the model can be improved to do a better job as compared to the existing model in the future.

Conclusion

Free Form Visualization

One of the important aspects of the project was the use of natural language processing techniques to identify the similarity between pairs of questions. Natural language processing techniques are focused on working with free form text data and allows user to create new features based of existing text and derive patterns in text data. One of the technique that drew my attention was tf-idf or **term frequency–inverse document frequency.** It reflects how important is a word to a document. Tf-idf was used during feature generation to identify the weighted score of each word in the document which increase proportionally with occurences of the word in the document. Based on these weighted scores cosine similarity is calculated to identify the similarity of two questions. Below diagram depicts the distribution of question pairs and their cosine similarity.



Reflection

The whole journey in Machine Learning Nanodegree has helped me improve my technical as well as has helped me improve on my research skills on different machine learning topics, algorithms and techniques. It also helped me developed an end to end implementation of real world problem.

Quora's question pairs competition on Kaggle was an interesting and challenging project with Natural language processing techniques. The competition presented the user with unique opportunity to work on different feature creation methods and algorithms used in natural language processing.

The project involved a lot steps like

- Data exploration
- Data cleaning and preprocessing
    - Removing stop words
    - Removing punctuations
    - Using stemming to reduce the words to its roots
- feature creation using the existing data
- Implementation
- validation of the results

Out of all the steps involved in the project, the most challenging and rewarding step was feature creation using existing data. This gave opportunity to research different methods involved in Natural language processing techniques and I would like to explore this area into more depth in future. The project also helped understand the XGBoost algorithms in depth in addition to very commonly used logistic regression. Overall the solution to this project was satisfactory although there are opportunities to improve upon the existing work that was done. In future I would like to explore the use of Neural networks , Support Vector machines on larger datasets thats require more processing power and better Natural Language processing techniques

Improvement

There are many improvements that can be made to XGBoost algorithm implementation, I feel there is a lot of potential to be uncovered in the XGboost algorithm. I would like to explore XGboost in more detail in future and try to work on different projects that can make effective use of the algorithm and techniques.

One of the improvements that was tried in the implementation was to use rebalancing of data as defined in the blog [12] . Instead of scaling the data using min_max_scaler function for scikit learn, using the rebalancing of data technique, The training data has 37% of the positive class while testing data has only 17%of positive class. rebalancing the data will make sure the training data also will have 17% of positive class in the data. Adding rebalancing data section helped improve the log loss score on XGboost significantly to ~0.39. I haven't used the technique in the submission because of low confidence on the theory.

In the future I would also like to try different algorithms like support vector machines, neural networks to solve these kind of problems. One thing I would like to improve is the expertise in Natural Language processing specifically learning about the process of creating new features to improve the algorithms.

References

1. Stop words : https://www.geeksforgeeks.org/removing-stop-words-nltk-python/
2. stemmer : http://www.nltk.org/howto/stem.html
3. Feature-Engineering: https://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/
4. TF-idf : https://en.wikipedia.org/wiki/Tf%E2%80%93idf
5. Cosine Similarity: https://janav.wordpress.com/2013/10/27/tf-idf-and-cosine-similarity/ https://stackoverflow.com/questions/6255835/cosine-similarity-and-tf-idf
6. Log Loss : https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234
7. Log Loss https://datawookie.netlify.com/blog/2015/12/making-sense-of-logarithmic-loss/
8. LogisticRegression https://towardsdatascience.com/logistic-regression-using-python-sklearn-numpy-mnist-handwriting-recognition-matplotlib-a6b31e2b166a
9. Logistic Regression http://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
10. XGBoost https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/ https://machinelearningmastery.com/develop-first-xgboost-model-python-scikit-learn/
11. https://www.kaggle.com/davidthaler/how-many-1-s-are-in-the-public-lb
12. Rebalancing Data: https://www.kaggle.com/anokas/data-analysis-xgboost-starter-0-35460-lb