QUOTA QUESTION PAIRS
Final Data Analysis Report

Yikun Zhang* SID:14336275

ABSTRACT

This report aims at elucidating the detailed data analyses on the classification of duplicate Quora questions pairs. The feature variables are created for modeling from two aspects, separating the raw texts into single words and comparing each pair of questions as whole strings. In the modeling part, Lasso, XGBoost, and Random Forest model are used separately to tackle the problem. Among these three models, Random Forest embraces the less Log Loss and thus has the most honorable performance. To further improve the model, an attempt to stack the preceding three models is implemented, though it cannot outperform its components due to sensitivity of the data set to be over-fitting.

Keywords: **Quora Question Pairs** **Lasso** **XGBoost** **Random Forest** **Over-fitting**

* *Department of Mathematics, Sun Yat-sen University, Guangzhou, China*

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Where else but Quora can a physicist help a chef with a math problem and get cooking tips in return? Quora is a place to gain and share knowledge about anything. It is 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 is 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. [1]

Therefore, to increase the efficiency of answering questions on Quora, it designs a natural language processing problem on Kaggle, where competitors are challenged to apply advanced techniques to classify whether question pairs are duplicate or not.

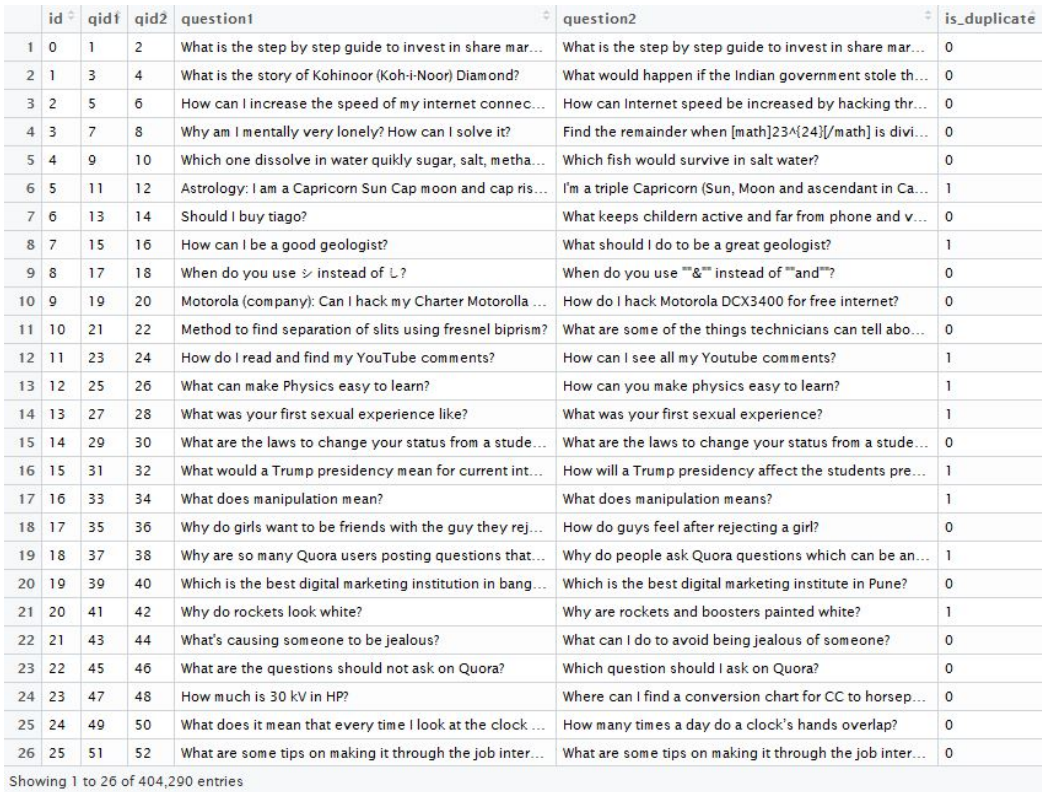
1.1 Data Description

Basically, the data of the competition is consisted of two files, i.e., train.csv and test.csv. They are all available on the competition website.[2] All of the questions in the training set are genuine examples from Quora. Also, in the training set, there exists a target variable (column) called **is_duplicated**, set to 1 if question1 and question2 have essentially the same meaning, and 0 otherwise. (See **Figure** 1 for details)

Data Fields

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

Caveat: Due to the time complexity of modeling, the following analyses will be based on a subset of the training set. The actual training set used for modeling contains **10000** rows that were randomly sampled from the original training set, and the actual test set used for validation contains **60000** rows that were also randomly sampled from the original training set. (For instance, in the following Feature Engineering part, the minimal elapsing time on the whole training set is approximately 300 hours. More importantly, this estimation is based on linear calculations, while the augmentation of running time could be nonlinear in practice.)



	id	qid1	qid2	question1	question2	is_duplicate
1	0	1	2	What is the step by step guide to invest in share mar...	What is the step by step guide to invest in share mar...	0
2	1	3	4	What is the story of Kohinoor (Koh-i-Noor) Diamond?	What would happen if the Indian government stole th...	0
3	2	5	6	How can I increase the speed of my internet connec...	How can Internet speed be increased by hacking thr...	0
4	3	7	8	Why am I mentally very lonely? How can I solve it?	Find the remainder when [math]23^{24}[/math] is divi...	0
5	4	9	10	Which one dissolve in water quikly sugar, salt, metha...	Which fish would survive in salt water?	0
6	5	11	12	Astrology: I am a Capricorn Sun Cap moon and cap ris...	I'm a triple Capricorn (Sun, Moon and ascendant in Ca...	1
7	6	13	14	Should I buy tiago?	What keeps childern active and far from phone and v...	0
8	7	15	16	How can I be a good geologist?	What should I do to be a great geologist?	1
9	8	17	18	When do you use シ instead of し?	When do you use ""&"" instead of ""and""?	0
10	9	19	20	Motorola (company): Can I hack my Charter Motorolla ...	How do I hack Motorola DCX3400 for free internet?	0
11	10	21	22	Method to find separation of slits using fresnel biprism?	What are some of the things technicians can tell abo...	0
12	11	23	24	How do I read and find my YouTube comments?	How can I see all my Youtube comments?	1
13	12	25	26	What can make Physics easy to learn?	How can you make physics easy to learn?	1
14	13	27	28	What was your first sexual experience like?	What was your first sexual experience?	1
15	14	29	30	What are the laws to change your status from a stude...	What are the laws to change your status from a stude...	0
16	15	31	32	What would a Trump presidency mean for current int...	How will a Trump presidency affect the students pre...	1
17	16	33	34	What does manipulation mean?	What does manipulation means?	1
18	17	35	36	Why do girls want to be friends with the guy they rej...	How do guys feel after rejecting a girl?	0
19	18	37	38	Why are so many Quora users posting questions that...	Why do people ask Quora questions which can be an...	1
20	19	39	40	Which is the best digital marketing institution in bang...	Which is the best digital marketing institute in Pune?	0
21	20	41	42	Why do rockets look white?	Why are rockets and boosters painted white?	1
22	21	43	44	What's causing someone to be jealous?	What can I do to avoid being jealous of someone?	0
23	22	45	46	What are the questions should not ask on Quora?	Which question should I ask on Quora?	0
24	23	47	48	How much is 30 kV in HP?	Where can I find a conversion chart for CC to horsep...	0
25	24	49	50	What does it mean that every time I look at the clock ...	How many times a day do a clock's hands overlap?	0
26	25	51	52	What are some tips on making it through the job inter...	What are some tips on making it through the job inter...	0

Showing 1 to 26 of 404,290 entries

Figure 1: A Snapshot of the Training Set

No items in the sampled training set and test set are overlapped. In addition, the proportion of items between the sampled training set and test set is closed to the actual context, which means that the following analysis and modeling is valid.

1.2 Evaluation

In this report, the results are mainly evaluated on the **Log Loss** between the predicted values and the ground truth. Given a predicted probability for each class of the target variables, the Log Loss metric is negative the log likelihood of the model that says each test observation is chosen independently from a distribution that places the predicted probability mass on the corresponding class, for each observation. In the binary classification case, the Log Loss becomes

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{2} y_{i,j}\log(p_{i,j})$$

where **N** is the number of observations, *log* is the natural logarithm, $y_{i,j}$ is 1 if observation *i* is in class *j* and 0 otherwise, and $p_{i,j}$ is the predicted probability that observation *i* is in class *j*.

2 EXPLORATORY DATA ANALYSIS

Unlike other usual problems that involve only numerical or categorical variables, natural language processing problems require programmers to extract useful information in the raw text and create their own feature variables for modeling. Thus, in this part, some exploratory data analyses (EDA) will be carried out in order to detect latent relationships among pairs of questions and the target variables. Before EDA, some basic data cleaning is needed.

2.1 Data Cleaning

One may cast doubt on the necessity of data cleaning at this stage, since it is more common to conduct data cleaning according to the EDA results. For natural language processing problems, however, it is of greater significance to carry out basic data cleaning before EDA so as to reduce noises in the following analyses and obtain more meaningful information.

Basically, there are four primary steps for data cleaning,

1. Remove some extra white spaces , "return" characters, and "new line" characters
2. Drop those html tags
3. Drop the text between "[]" referring to tags, e.g. [math]
4. Remove punctuation except for those dashes inside words

After the preceding four steps, all the character in questions will be converted into lower cases.

2.2 Target Variable vs Pairs of Questions

In our commonsense, the more identical words between two questions, the higher probabilities for two questions to be duplicate. Therefore, it becomes natural to explore the relationship between the number of words that occur both in Question 1 and Question 2 and the target variable. A histogram will be helpful to clarify this relation. (See Figure 2 for details). To create this plot, the "ggplot2"[4] package is used.

From the histogram, one can notice that pairs of questions are more likely to be independent when the identical words counting is low (less than 4). When the number of words in common is moderately high ($4 \leqslant comword < 7$), however, chances are high that those pairs of questions are duplicate. Nevertheless, this relation does not seem quite conspicuous when the identical word counting is extremely high. Based on this interesting relationship, it is convincing to include

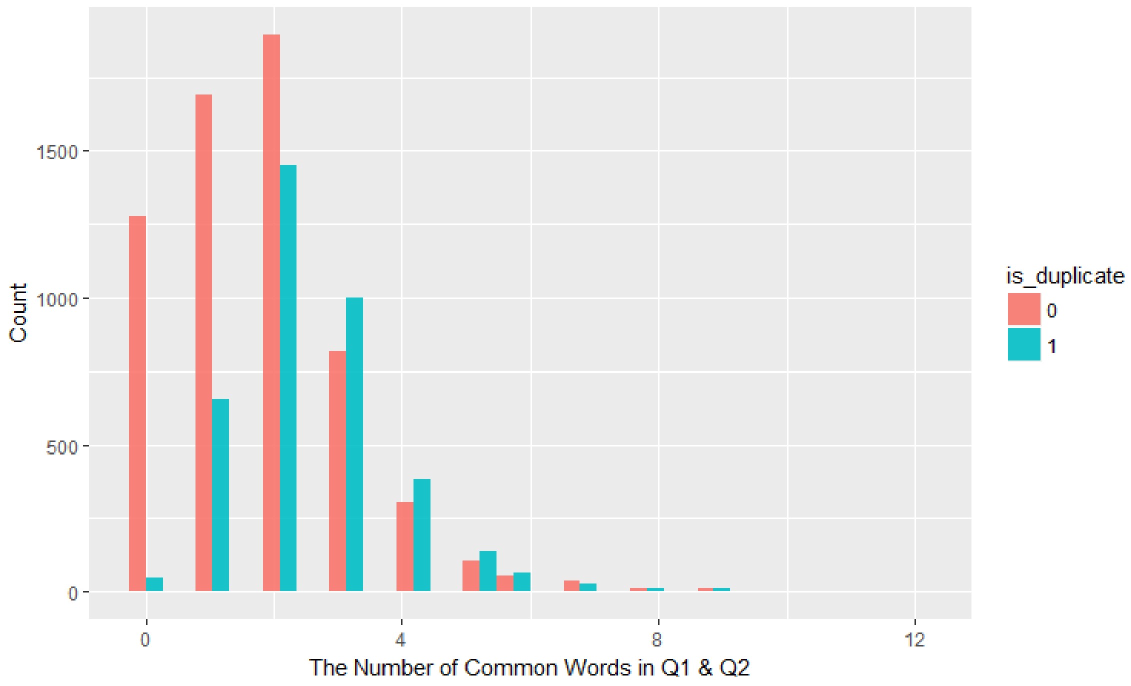


Figure 2: Histogram: The Number of Words that Occurs BOTH in Q1 and Q2

the identical words counting as well as the ratio of words in common in the creation of feature variables. Likewise, one can also explore whether the identical character counting may influence the target variable. For the simplicity of the report, this exploration is omitted.

2.3 Latent Features of Pairs of Questions

2.3.1 *Tf-idf Exploration*

A central question in text mining and natural language processing is how to quantify what a document is about. One measure of how important a word may be is its *term frequency* (tf), how frequently a word occurs in a document. There are words in a document, however, that occur many times but may not be important; in English, these are probably words like "the", "is", "of", and so forth. One might take the approach of adding words like these to a list of stop words and removing them before analysis, but it is possible that some of these words might be more important in some documents than others. A list of stop words is not a very sophisticated approach to adjusting term frequency for commonly used words. Another approach is to look at a term's *inverse document frequency* (idf), which decreases the weight for commonly used words and increases the weight for words that are not used very much in a collection of documents. The inverse document frequency for any given term is defined as

$$idf(\text{term}) = \log\left(\frac{n_{\text{documents}}}{n_{\text{documents containing term}}}\right)$$

This can be combined with term frequency to calculate a term's tf-idf (the two quantities multiplied together), the frequency of a term adjusted for how rarely it is used.[3]

Thanks to the advantage of the tf-idf concept, the following exploration will focus on mining those words with high tf-idf in Question 1 and Question 2, respectively. (See Figure 3 for details). To create this plot, the "ggplot2"[4] and "tidytext"[5] packages are used.

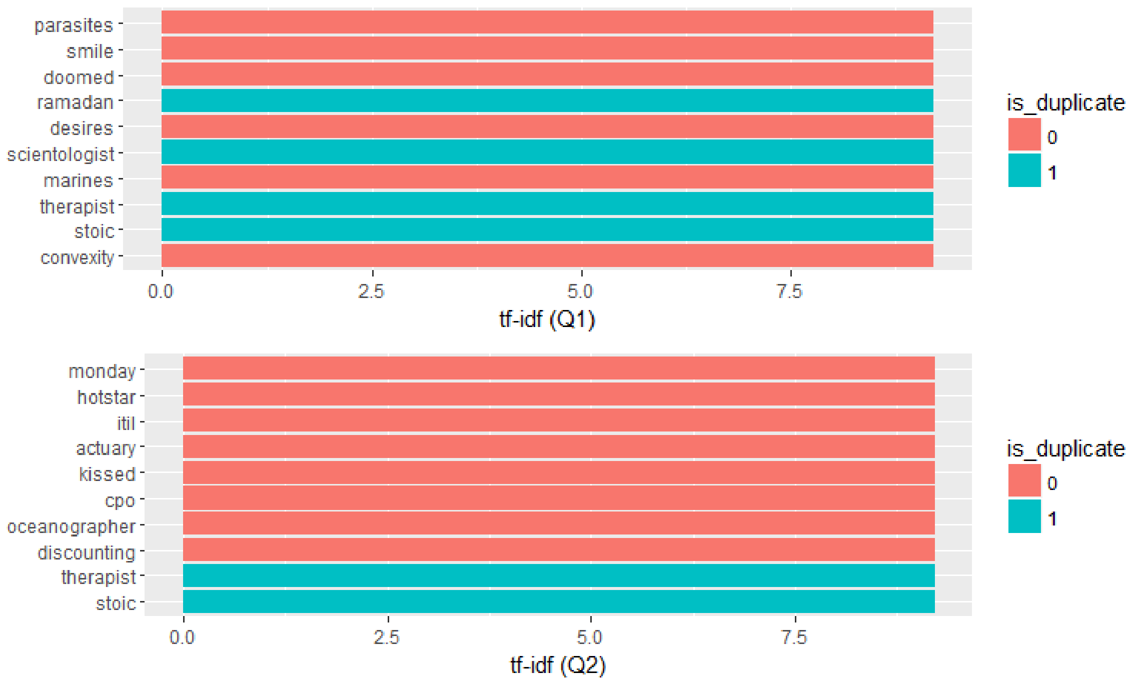


Figure 3: A visualization for high tf-idf words in Q1 and Q2 (the upper plot refers to Question 1 and the lower one corresponds to Question 2) [1] [6]

From Figure 3, 10 words with highest tf-idf were selected (with ties) from Question 1 and Question 2, respectively. A crucial phenomenon in this figure is that none of those words appearing in the plot can be regarded as meaningless words that may not contribute to the judgement of duplicate questions, which, in some sense, verifies the appropriateness of creating some features referring to

[1] *During this plotting, a helper function called "multiplot" is used. The helper function is created by a teammate, Haoran Li, when the author worked as an exchange student at University of California, Berkeley. The author wants to express his gratitude towards this teammate for his contribution in that final project. [6]*

tf-idf. More importantly, some careful readers may note that the word "therapist" appears in both of these two plots and belongs to the duplicate class in both cases. Thus, chances are high that the pairs of questions are duplicate if they all contain the word "therapist". These nontrivial phenomena can be generalized to other words in the questions, which is conducive to the classification of duplicate questions.

2.3.2 *Counting Pairs of Words in Questions*

Tf-idf method is useful when it comes to detecting some important words that are not commonly used in documents. However, there might be some words that tend to follow others immediately or co-occur in the same document. Hence it becomes crucial to count the number of times that two words appear within a pair of questions and to see how correlated they are. The "widyr"[7] package makes operations such as computing counts and correlations easy, by simplifying the pattern of "widen data, perform an operation, then re-tidy data".[3] Before analyzing the counting pairs of words in questions, Question 1 and Question 2 were literally concatenated and each row with a unique id in the data frame would be viewed as a single document. With the help of "ggragh"[8] and "igraph"[9] packages, the counting of pairs of words in questions and their clusters can be visualized clearly. (See Figure 4 for details)

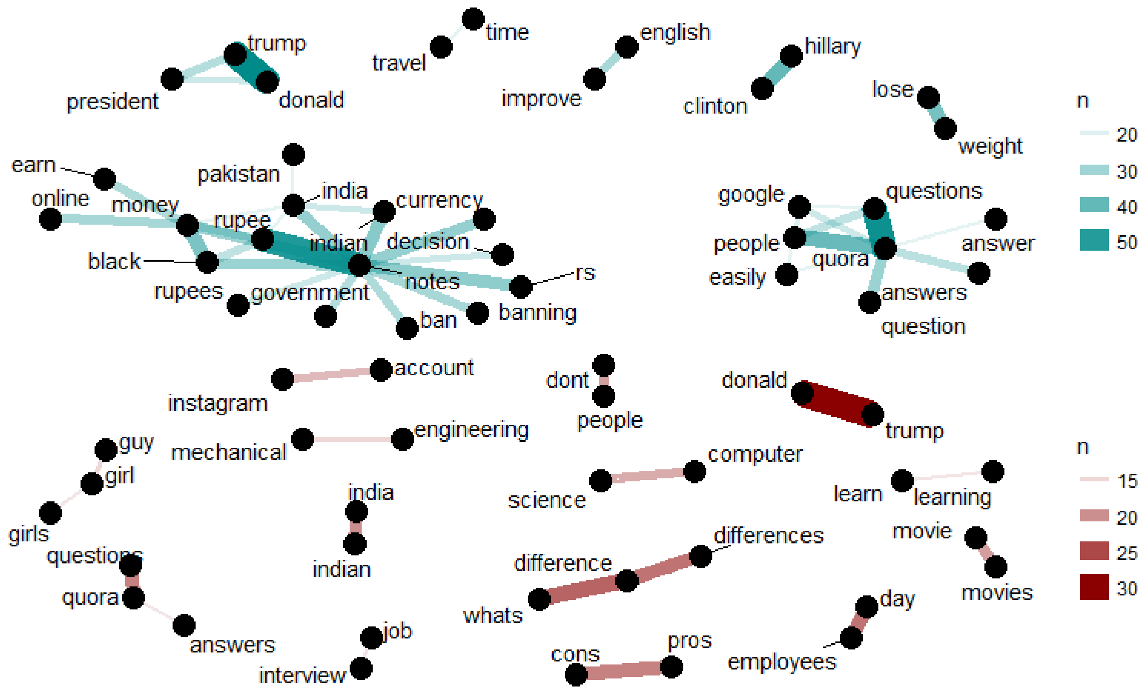


Figure 4: Networks of co-occurring words (The upper cyan plot refers to the clusters of words in duplicate questions, while the lower dark red one visualizes the network of words in independent questions)[2] [6]

[2] *The helper function "multiplot" is also used here. [6]*

From Figure 4, there are a bunch of words that tends to appear simultaneously in a single question. For example, it makes no sense to separate our discussions towards the pair of words, "donald" and "trump" or "hillary" and "clinton", since they are all referring to the presidential candidates' names. Moreover, those words ,like "currency", "ban", and "indian", which have strong correlations with government, have also been clustered into a huge connected network. In addition, some verbs embrace strong correlations with their derivative forms, like "ban" and "banning". Upon these findings, it is necessary to take the counting of pairs of words in questions into consideration.

3 FEATURE ENGINEERING

In the preceding part, some significant exploratory data analyses have been carried out in order to discover some interesting features and facilitate the modeling part. In this part, more emphasis will be placed on the construction of feature variables. As the paramount step before modeling, feature engineering works as the foundation of modeling and essentially determine the overall accuracy of a model. Therefore, it is worthwhile to illustrate how all those feature variables are created in this report.

3.1 Feature Variables Based on Words

As mentioned in the EDA part, the identical words counting and character counting come into data scientists' mind firstly when it comes to the comparison of some documents. Hence after removing stop words, it is natural to incorporate **identical words counting, the ratio of words in common**

$$1 - \frac{identical\,words\,counting}{maximum\,words\,counting\,of\,a\,question\,pair}$$

, and **the difference ratio of character counting**.

$$1 - \frac{|character\,count\,in\,Q1\, - \,character\,count\,in\,Q2|}{maximum\,character\,count\,of\,a\,pair\,question}$$

Likewise, **the difference ratio of word counting** is also included in the feature variables, which can be computed by the previous formula and substitute "character" with "word".

Apart from words in common, the exploration about tf-idf of words in questions unearth some important words in pairs of questions that might determine whether the pairs of questions are duplicate. As a result, the feature variables, **the difference ratio of top 3 words' tf-idf**

$$1 - \frac{|average\,top\,3\,tfidf\,in\,Q1\ -\ average\,top\,3\,tf - idf\,in\,Q2|}{maximum\,words'\,tfidf\,of\,a\,pair\,question}$$

, **the sum of identical words' tf-idf in Q1**, and **the sum of identical words' tf-idf in Q2** are made.

Besides tf-idf, the counting of pairs of words in questions also plays a crucial role in the determination of duplicity of a pair of questions. Since it is cumbersome and useless to take all the pairs of words into account, our focus can be narrowed to **the average counting for those identical words among pairs of questions**, which is substantial to represent the correlations between a pair of questions.

3.2 Feature Variables Based on Whole Strings

Up to now, all the feature variables are made after splitting the questions into single words or characters. On one hand, the splitting can reduce noises between words and concentrate on the comparison of small items. On the other hand, some structures of sentences or phrases would be destroyed after splitting. To compensate this shortage, a feature variable, **string distance**, is included. The string distance uses **Levenshtein distance** as its metric, which counts the number of deletions, insertions and substitutions necessary to turn string b into string a. [10]

In most cases of natural language processing, data scientists feel like converting the raw texts into a document-term matrix. This method successfully changes the untractable texts into numerical data and enables further applications of numerical modeling and analysis. In R software, many packages are available to tackle this problem. In this report, the "text2vec" package[11] is used to implement the transition to a document-term matrix and computes **the Jaccard similarity between a pair of questions**. The reason why Jaccard similarity was chosen here instead of Cosine Similarity is that the results of Jaccard similarities and Cosine similarities are essentially the same with respect to the order and scale.

Last but not least, **the positive sentiment difference between Question 1 and Question 2** are also incorporated into the feature variables, since the duplicate questions tend to effuse similar emotions.

To give a brief conclusion of the Feature Engineering part, nothing goes better than presenting a summary table that includes all the feature variables for further reference. (See Table 1 for details)

All the feature variables are numerical and most of them are within 0 to 1.

Table 1: Summary of Feature Variables

Feature Variables' names	Meaning
com_word	Identical words counting of Question 1 and Question 2
com_word_ratio	Defined ratio of words in common in Question 1 and Question 2
char_count_ratio	Defined difference ratio of character counting
word_count_ratio	Defined difference ratio of word counting
tfidfdiff	Defined difference ratio of top 3 words' tf-idf
q1_com_tfidf	Sum of tf-idf within Question 1 for those words in common
q2_com_tfidf	Sum of tf-idf within Question 2 for those words in common
com_pair_count	Average counting for identical words among pairs of questions
qu_dist	Levenshtein distance between Question 1 and Question 2
q1_q2_sim	Jaccard Similarity between Question 1 and Question 2
pos_diff	Positive sentiment difference between Question 1 and Question 2

4 MODELING

The modeling part aims at fitting a binary classifier based on the preceding feature variables to predict the value of the target variables in the test set and minimize the error under the prescribed metric (in this case, Log Loss). As indicated in the last part, all the feature variables are numerical. Hence it becomes natural to apply regression models to fitting a binary classifier. For a predicted probability of each class of the target variable, logistic regression is one of the most well-performed models to tackle this problem. Since the importance of each feature variables is unsure, one must be careful when applying a logistic regression. To avoid misusing some insignificant variables, Lasso Regression (Least Absolute Shrinkage and Selection Operator)[12] can be applied. Furthermore, xgboost (Extreme Gradient Boosting)[13] also provides an option to fit a logistic regression model using boosting methods. Finally, whereas Random Forest[14] embraces some honorable performances on Kaggle competition and its intrinsic property for variables' selection, it is worthwhile to apply the Random Forest method in this binary classification problem.

4.1 Lasso

T.Hastie et.al(2010) implemented the Lasso Algorithm in R via the "glmnet" package[15]. This outstanding package facilitates the implementation of Lasso and Ridge Regression and enables cross-validation when fitting a Lasso model.

During the programming, 5-fold cross-validation was used so as to minimize the test errors. The Log Loss of the lasso model on the test set is **0.6506921**.

4.2 XGBoost

In R, there is a package called "xgboost"[16], which makes it plausible to apply XGBoost on dense or sparse matrices. Instead of simply tuning the parameters manually, 5-folds cross-validation was used to grid-search the optimal values for "eta" (step size shrinkage used in update to prevents overfitting) and "gamma" (minimum loss reduction required to make a further partition on a leaf node of the tree). However, according to practical results, the Log Loss are sensitive to the number of rounds for boosting, especially the Log Loss for the test set. As a result, it is wise to grid-search the optimal values for "eta" and "nrounds" (the max number of iterations) simultaneously. Basically, the other tuning parameters of *xgboost()* function are set as follows. (See Table 2 for details)

Table 2: Parameters for XGBoost

Parameter	Meaning	Value
objective	Specify the learning task and the corresponding learning objective	"binary:logistic"
eval.metric	Evaluation metrics for validation data	"logloss"
booster	Which booster to use, gbtree and dart use tree based model	"gbtree"
gamma	Minimum loss reduction required to make a further partition on a leaf node of the tree	1
subsample	Subsample ratio of the training instance	0.7
colsample.bytree	Subsample ratio of columns when constructing each tree	1
min.child.weight	Minimum sum of instance weight (hessian) needed in a child	0.5
max.depth	Maximum depth of a tree, increase this value will make the model more complex / likely to be overfitting	4

Meanwhile, as mentioned earlier, 5-folds cross-validation was used to grid-search the optimal values for "eta" and "nrounds" simultaneously. To inspect the changes of Log Loss with respect to "eta" and "nrounds", a 3-D scatter-plot in the "plot3D" package[17] was drawn to visualize the nuances between each round of the double loops. (See Figure 5 for details)

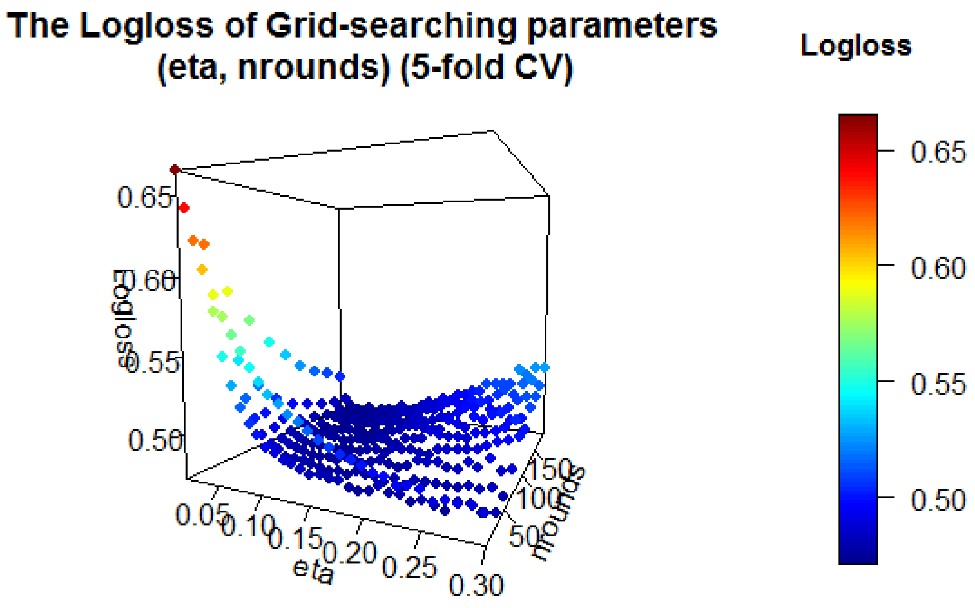


Figure 5: Scatter plot of Log Loss of Grid-searching parameters ("eta" and "nrounds") (5-folds cross-validation was used.)

From the above scatter plot, one can note that the Log Loss drops dramatically when increasing the "eta" and "nrounds". Careful readers might notice, however, that the Log Loss has a slight increase when the "nrounds" continues its increasing. Practical results shows that this peculiar phenomenon becomes more conspicuous when "nrounds" reaches 5000 or more and, in some sense, illustrates the sensitivity of Log Loss towards "nrounds" and the indication of over-fitting. Therefore, it is necessary to conduct a grid-search algorithm towards "eta" and "nrounds", though it goes against our commonsense.

After the preceding grid-search algorithm, the optimal values for "eta" is **0.07** and **190** for "nrounds". Finally, the Log Loss for XGBoost is **0.5070453**, which is noticeably better than Lasso model.

4.3 Random Forest

Last but not least, the "randomForest" package [18] was applied to build the random forest model. Since not all the feature variables are useful in the modeling, random forest algorithm provides a convenient way to select feature variables based on their importance. A visualization of the importance based on mean decrease Gini index was drawn as follows. (See Figure 6 for details)

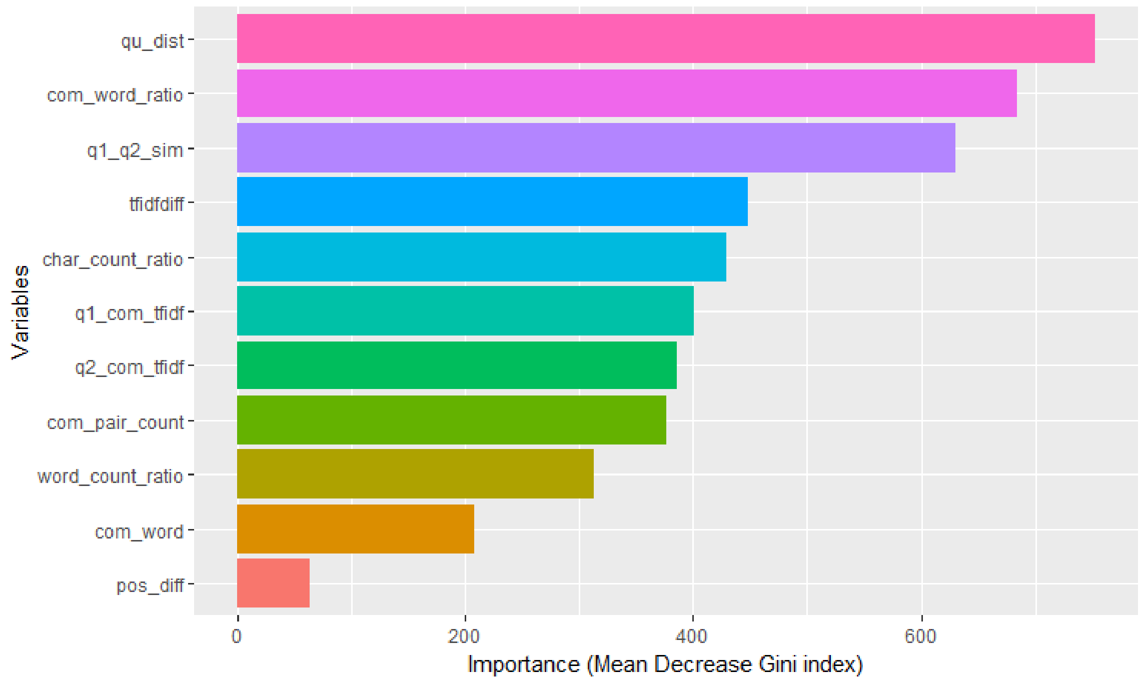


Figure 6: Importance of Feature Variables Based on Mean Decrease Gini

Figure 6 verifies that the Levenshtein distance between Question 1 and Question 2, defined ratio of words in common in Question 1 and Question 2, and Jaccard similarity between Question 1 and Question 2 are the top 3 feature variables that make greatest contribution to the modeling, whose importance indices are greater than 600. These significant variables are meaningful and should be reserved for future use. Moreover, the Log Loss for the random forest model is **0.5006323**, which is slightly better than the XGBoost model.

5 REFLECTION AND FURTHER IMPROVEMENT

Based on the previous results, the random forest model embraces the best performance among all three models with the Log Loss **0.5006323**. Since only a single model was used to build the preceding model, it is natural to reflect whether the performance would be improved when multiple models were combined or stacked. In addition, previous inspections on the importance of features variables reveal that not all the feature variables make significant contributions to the accuracy of the prediction. Hence it becomes necessary to drop those variables whose mean decrease Gini index is less than 400 before model combining.

5.1 Model Stacking

A quick reflection on the previous results reveals that Lasso model is the worst model among all three models while random forest model is the best. Does it always come true that the combination of three models embrace better performances than its components? Probably not. In model stacking theory, the predictions of one model can be considered as a feature variable of the later one. Therefore, the output of Lasso model played a role of a feature variable in the building part of XGBoost model. Later, the prediction of XGBoost model also worked as a feature variable of random forest model. The stacking process maximized the advantage of Lasso and XGBoost model and make use of the variable selection mechanism of random forest model to verify the validation of stacking algorithm. (See Figure 7 for details)

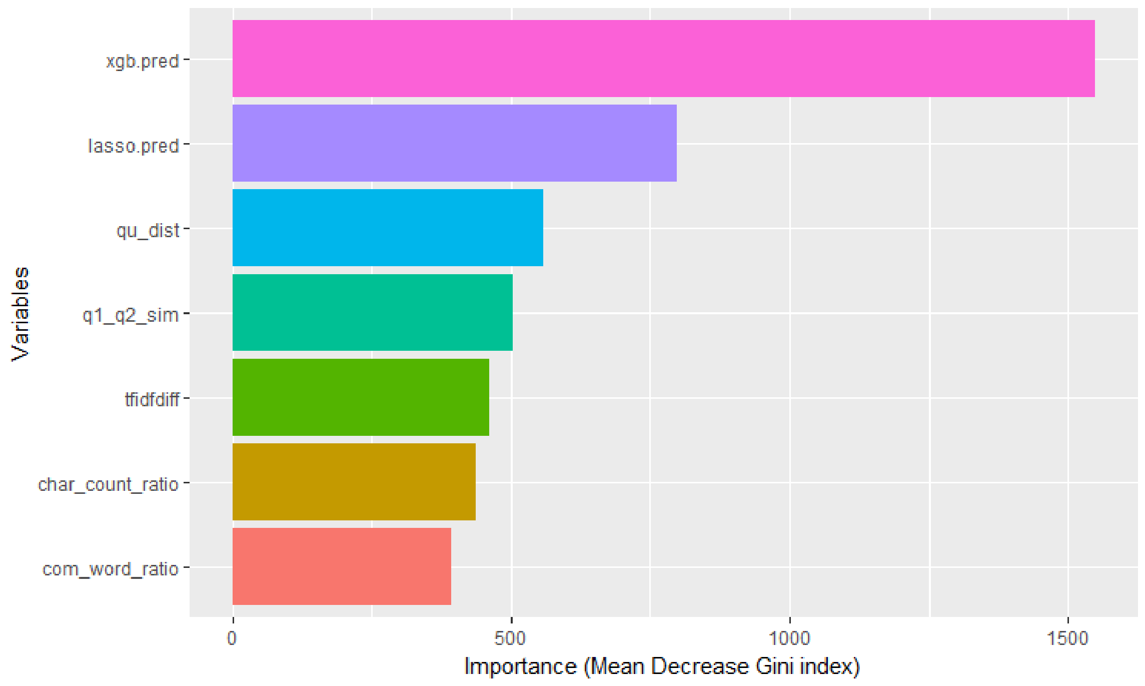


Figure 7: Importance of Feature Variables After Model Stacking Based on Mean Decrease Gini

Figure 7 could be a convincing evidence that model stacking made use of the prediction of the other two models, since the prediction of XGBoost and Lasso had the highest Gini indices among all the remained feature variables. The Log Loss of this model stacking, **0.5647179**, however, goes beyond our experience, for it is not only worse than its "component" XGBoost model but also its "architect" random forest model. Its reason basically lies in the independency of data items (rows). All the question pairs were carefully selected by organizers and had nearly no inner connection between each pair. The independency of sample items directly allows for its tendency for an algorithm to be over-fitting, which means that the better performance a model embrace on the training set, the more disappointing prediction it would make on the test set. Unfortunately, model stacking is an algorithm which relies heavily on the performance of some stacked models on the training set. Therefore, there is no doubt that worse Log Loss appeared on model stacking in this test set.

5.2 Reflection

Although this data set is extremely independent between rows and sensitive to over-fitting phenomenon, the creation of feature variables are inevitably responsible for some parts of the magnitude of this independency. If, on the other hand, some connections between variables could be carefully scrutinized, chances might be lower for the models to be over-fitting.

Apart from being less considerate when creating feature variables, the time complexity plays as the second obstacle for the models to have better performances. At first, the functional programming of feature engineering required more than **5 days** to clean the data set and carry out all the pre-specified feature variables. What's worse, even though a compromise had been made because all the analyses were based on a subset of original training set, it still took **237639.4s** to run a grid-searching algorithm for XGBoost. With such low time efficiency, it became a tough task to tune all the parameters in the models.

Last but not least, some other ensemble learning algorithm had been ignored. Actually, some superficial ensemble learning algorithm had been tested on the data set and also led to disappointing outputs. For instance, except for model stacking, it turns out that a weighted combination of different models still could not do better than a single one. Nonetheless, some other ensemble methods, like Bayesian modeling averaging, might substantially make a balance on bias-variance tradeoff and prevent overfitting. It is unconvincing to negate the ensemble learning before exhausting all the combinations of ensemble learning methods.

6 CONCLUSION

This report summarized all the main works of the author to tackling the classification problem of Quora Question Pairs. Some traditional text mining methods were used, like tf-idf methods, in order to create feature variables. Meanwhile, more attentions were paid on comparing each pair of question as a whole string instead of separating them into some bags of words. Such an innovative method is conducive to detecting the duplicity of each pair of raw text and can be further used. To further buttress the investigation of duplicate questions, Topics Model, especially Latent Dirichlet Allocation[19], could be a nice attempt to roughly categorize data items (rows) into some latent categories (topics). This preliminary investigation might be useful for modeling and work as an auxiliary classifier for the final model. The applications of detecting duplicate question pairs lie in various fields of our life. For instance, experts no longer have to answer the duplicate questions one by one and focus on dealing with new problems. At the same time, puzzlers need not to wait for a whole day to get a satisfying answer for their questions. More importantly, the model for detecting duplicate question pairs can also discover plagiarism of academic papers and prelude violation of academic integrity, though some details of the model should be carefully modified.

7 BIBLIOGRAPHY

REFERENCES

[1] Kaggle ***Quota Question Pairs (Overview)*** Retrieved from https://www.kaggle.com/c/quora-question-pairs.

[2] Kaggle ***Quota Question Pairs (Data)*** Retrieved from https://www.kaggle.com/c/quora-question-pairs/data.

[3] Julia Silge, David Robinson ***Text Mining with R: A Tidy Approach*** 2017-05-07 URL: http://tidytextmining.com/index.html

[4] Hadley Wickham ***ggplot2: Elegant Graphics for Data Analysis*** Springer-Verlag New York, 2009 URL: http://ggplot2.org

[5] Julia Silge and David Robinson ***tidytext: Text Mining and Analysis Using Tidy Data Principles in R*** The Open Journal, 1(3), 2016 URL: http://dx.doi.org/10.21105/joss.00037

[6] Chang Su, Haoran Li, Jingyi Liu, Wideet Shende, Yikun Zhang ***Election Prediction Report*** Fall 2016, Final Project for STAT 133: Concepts in Computing with Data, Instructor: Deborah Nolan, University of California, Berkeley

[7] David Robinson ***widyr: Widen, process, and re-tidy a dataset*** Retrieved from https://github.com/dgrtwo/widyr

[8] Thomas Lin Pedersen ***ggraph: An Implementation of Grammar of Graphics for Graphs and Networks*** R package version 1.0.0, 2017 URL: https://CRAN.R-project.org/package=ggraph

[9] Gabor Csardi and Tamas Nepusz ***The igraph software package for complex network research*** InterJournal, Complex Systems(1695), 2006 URL: http://igraph.org

[10] M.P.J. van der Loo ***The stringdist package for approximate string matching*** The R Journal, 6(1), 2014 URL: `https://CRAN.R-project.org/package=stringdist`

[11] Dmitriy Selivanov ***text2vec: Modern Text Mining Framework for R*** R package version 0.4.0, 2016 URL: `https://CRAN.R-project.org/package=text2vec`

[12] Robert Tibshirani ***Regression Shrinkage and Selection via the Lasso*** Journal of the Royal Statistical Society. Series B (Methodological) Vol. 58, No. 1 (1996), pp. 267-288 URL: `http://www.jstor.org/stable/2346178`

[13] Tianqi Chen and Carlos Guestrin ***XGBoost: A Scalable Tree Boosting System*** ACM ISBN

[14] Leo Breiman ***RANDOM FORESTS–RANDOM FEATURES*** Technical Report 567, 1999 URL: `https://www.stat.berkeley.edu/~breiman/random-forests.pdf`

[15] Jerome Friedman, Trevor Hastie, Robert Tibshirani ***Regularization Paths for Generalized Linear Models via Coordinate Descent*** Journal of Statistical Software, 33(1), 1-22, 2010 URL: `http://www.jstatsoft.org/v33/i01/`

[16] Tianqi Chen and Tong He and Michael Benesty and Vadim Khotilovich and Yuan Tang ***xgboost: Extreme Gradient Boosting*** R package version 0.6-4, 2017 URL: `https://CRAN.R-project.org/package=xgboost`

[17] Karline Soetaert ***plot3D: Plotting Multi-Dimensional Data*** R package version 1.1, 2016 URL: `https://CRAN.R-project.org/package=plot3D`

[18] Andy Liaw and Matthew Wiener ***Classification and Regression by randomForest*** R News, 2(3), 18-22, 2002 URL: `http://CRAN.R-project.org/doc/Rnews/`

[19] David M.Blei and Andrew Y.Ng and Michael I.Jordan ***Latent Dirichlet Allocation*** Journal of Machine Learning Research, 3(2003), 993-1022