Quora Question Pair Similarity

Haneesha Aldena  
 *Computer Science*  
*University of Central Missouri*  
700741127

Manohar Swarna  
 *Computer Science*  
*University of Central Missouri*  
700742333

Priyanka Nimmagadda  
 *Computer Science*  
*University of Central Missouri*  
700734089

***Abstract*— A user-generated database of questions and answers can be found on the expanding platform known as Quora. Users design, edit,and arrange the questions and responses. The issue of duplicate questions is brought up by the enormous number of users on the Quora website. This makes it inevitable that there will be several inquiries from different people with similar intentions. Identifying same questions would make the Quora platform stable and provide effective user experience to the users. we used Quora question pair dataset from kaggle and trained different model and performed Empirical evaluation.**

**Keywords:- Quora, question-pair, similarity, machine learning**

# **Introduction**

On the question-and-answer website Quora, individuals can ask and answer questions on a range of subjects. Global users of Quora frequently discuss their doubts or provide solutions based on personal experience. There have been countless inquiries. permits many people to ask the same question at the same time if two or more questions nearly match one another or have slightly different orders. All of the solutions can then be located in one place after they have been connected as one. It undermines the user experience and also allows users to search for answers to questions that have already been posed by others. If the solution is known, they might receive it right away.

The repetition of questions is one of the many issues that Quora has. The experience is ruined when a question is asked again, by both the questioner and the answerer. We may just show the questioner the responses to the earlier question since it is a duplicate question. Additionally, for questions that are almost identical, the respondent need not repeat their response.

The goal of this study is to determine whether or not the semantic meanings of two TVCO sentences are similar. The primary task is to identify which questions on Quora have already been asked. Eke's Questions "How to Become a actor?" and "What Should I Do to Become a Good actor?" are nearly identical but slightly worded differently. For MOOC courses, there are several community-based forums. which are confronted with the same issue Because of the millions of members, it is possible for multiple users to ask the same question.

Therefore, the answers to both questions will be the same. Therefore, we just need to display the responses to the first query. In this method, the person asking the question will receive the responses right away, and those who have already responded to the first question won't have to say it again.

The Quora question pairs dataset, which was originally made available to the public in 2017, has been extensively utilized for testing duplicate question detection techniques and training models. It consists of 404351 question pairs that have been manually tagged to show whether or not they are logically duplicates.

If two questions are comparable, they are labeled 1; otherwise, label 0 is applied. However, different humans disagree on a number of the labels in this dataset. Because of this, the ground truth labels in this dataset may not be true to the letter or could contain inaccurate labeling.

For short text similarity, some approaches have been proposed. They include a lexical match, knowledge base, bag of words model, and neural network. Lexical match primarily compares character similarities between two short texts using edit distance , largest common subsequence distance, Jaccard similarity coefficient, or lexical overlap. However, because lexical matches do not take into account semantic information within and between words in a short text, their effectiveness is limited. For example, the literals of synonyms differ, but their meanings are the same. WordNet and Wikipedia1 are the most commonly used knowledge bases for comparing short text similarity.

In this paper we used the quora question pair similarity dataset from the kaggle, And posses into a ML problem and looked the data overview and performed the Exploratory data analysis and performed text preprocessing. Also did advanced feature extraction and performed the data visualization. We used advanced features, basic features and text features to train a model. Here used the w2v/tfidf word to vector to build the text features.

Where the single question is represented as 384 dimension vector.

# **II. Motivation**

Where else but on Quora can a physicist assist a chef with a math issue in exchange for food advice? A place to learn and share information about anything is Quora. It serves as a forum for queries and connections with experts who offer insightful observations and thorough responses. People are better able to grasp the world and learn from one another as a result. It's hardly surprising that many questions on Quora are similar in wording given that over 100 million people visit the site each month. Multiple inquiries with the same objective can make readers feel as though they must respond to various variations of the same inquiry, while also making seekers spend more time looking for the best solution to their problem. Canonical questions are highly valued on Quora because they provide active writers and seekers a better experience and more long-term value. Quora platform providing useful answers to the questions for the users and within their production they using Machine learning model to map the answer to the duplicate questions. It’s a nice where got motivated and we have handful of data to play and train our own model to learn new things in Machine learning and NLP.

# **III. Contribution and objectives**

* The cost of miss-classification is very high, if we miss-classified the two questions are similar, which are actually not, and providing irrelevant answer to the question

Leads to worst experience.

* Want probability of two pair of questions to be duplicated, so that we can choose the threshold of choice.
* No any strict latency.
* Interpretability is partially important not fully.
* And contribution, along with basic features we performed advanced feature extraction and used the NLP techniques to build the text features for both question1 and question 2 in the data.

**IV. RELATED WORK**

I Question pair dataset question similarity task

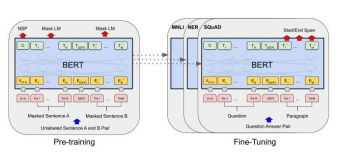
Bert is a potent paradigm for semantic representation. Sometimes a computer employing Bert can make a forecast that is more accurate than one made by a person. For instance, a BERT-based algorithm determines that the following question combination is "not duplicated" but is duplicated by humans. Is free healthcare a good idea? If not, why not? / Should Americans have access to universal health care? If not, why not?

These inquiries do not address the same issue. The outcome of the BERT-based algorithm is thus accurate in this instance.

To create a semantic representation of questions and to perform the classification task for each question pair, we therefore employ Bert. We suggest methods to eliminate out erroneously labeled question pairings in the Quora dataset by examining the discrepancy between Bert-based findings and manually annotated labels. We also suggest a way to change the labels on these pairs.

The remainder of the essay is structured as follows. Section 2 introduces our approach, which is used to determine whether two questions are comparable. The output of our Bert model is compared to the labels that were manually assigned in the Quora development dataset in Section 3. Our rule-based approach to identifying pairs of dissimilar questions is introduced in Section 4. In Section 5, a scenario for changing the labels for question pairings in the Quora dataset is laid out. Finally, Section 6 provides a summary of the paper.

The Bidirectional Encoder Representations from Transformers (Bert) algorithm is intended to jointly condition on both left and right context in all layers to pre-train deep bidirectional representations from an unlabeled text. A single sentence or two sentences in one token sequence are acceptable inputs for this system. Token embeddings, segment embeddings, and position embeddings are all used to represent each input token. A one-hot vector made from the 30,000-token WordPiece dictionary is used to illustrate token embedding. Segment embeddings show which of a pair of sentences the current token belongs to, either the first sentence or the second sentence. Position embeddings provide details about the locations of tokens within a given sentence.

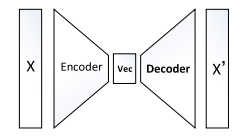
 II Text similarity with LSTM Encoder

A type of neural network called a recurrent neural network (RNN) uses sequence data (x1,..., xT) as its input, and the connections between its nodes create a directed graph along the sequence. Recently, it has become widely employed in picture description generation, speech recognition, and natural language processing (NLP). The output of the hidden layer from a prior time is included in the input of the hidden layer in addition to the output of the input layer. St = f(U\*Xt + W\*St1), where Xt is the input at time t, U denotes the weight matrix from input to current hidden state, W denotes the weight matrix from previous hidden state to current hidden state, and f denotes the activation function. RNN, though, exhibits gradient vanishing.

For short text similarity, some approaches have been proposed. They include a lexical match, knowledge base, bag of words model, and neural network. Lexical match primarily compares character similarities between two short texts using edit distance, largest common subsequence distance, Jaccard similarity coefficient, or lexical overlap. However, because lexical matches do not take into account semantic information within and between words in a short text, their effectiveness is limited. For example, the literals of synonyms differ, but their meanings are the same. WordNet and Wikipedia1 are the most commonly used knowledge bases for comparing short text similarity.

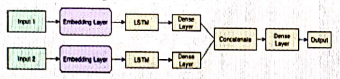
Gradient-based training learning techniques for long-distance learning can have problems where the gradients grow vanishingly small over lengthy sequences in backpropagated processes. Long short term memory (LSTM)  is created to address gradient vanishing. A RNN variation that can recognize long-term dependencies is the LSTM. Only one neural network layer exists in an RNN's hidden unit. The LSTM hidden unit, on the other hand, is made up of a memory cell, an input gate, an output gate, and a forget gate. Memory cells are used to store one or more values; input gates determine how many values enter the unit; output gates determine how many values exit the unit; and forget gates determine whether the value stays in the unit.

An unsupervised artificial learning neural network is the autoencoder. One input layer, one or more hidden layers, and one output layer are also included. As closely as possible, the output of the output layer will match the input of the input layer. Features are effectively compressed before being decompressed. It is mostly employed for feature extraction and dimensionality reduction of data.It currently includes an encoder and a decoder and is utilized in generative models. The output of an encoder is typically used as a data feature. The autoencoder schematic.



III Enhanced Deep learning model for duplicate question detection.

The Siamese network is an architecture composed of parallel neural networks, specifically LSTM units for parallel processing of two questions, each of which passes through an Embedding Layer. an LSTM unit, followed by a dense layer Following parallel processing, the outputs of two networks are combined and compared, yielding a similarity score indicating how similar two questions are.

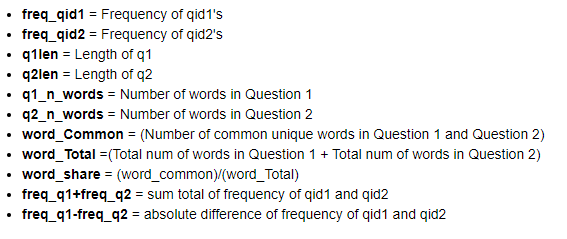


A Siamese LSTM neural network is a type of neural network. It is a neural network composed of two identical neural networks that are combined via a dense layer to produce final output. Here. For questions 1 and 2, two identical sub-networks are used. Subnetworks use the same parameters. The learned parameters of each sub network are identical, indicating that only one set of weights, not two, should be trained.

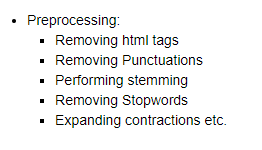
The loss function was set to Binary Cross Entropy loss, and the Adam optimizer was chosen because it was best suited to the architecture. Furthermore, Adam optimizer is the best optimize until now. The learning rate was held constant at 0.0001. In this case, sigmoid function was used to determine whether two questions were duplicates or not.

**V. Proposed Framed Work**

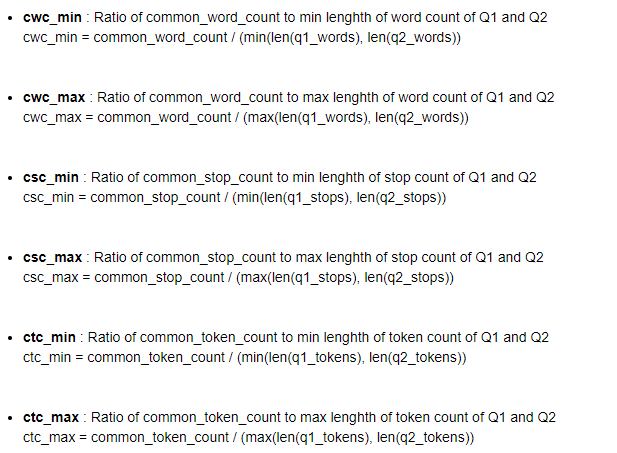
We first loaded the data and look the distribution of the class label and performed the basic analysis like percentage of the question pairs which are similar and not similar, number of unique questions. And checked for duplicates , occurrence of each question and also taken care of the null values. We performed the basic feature extraction before cleaning the data those are,

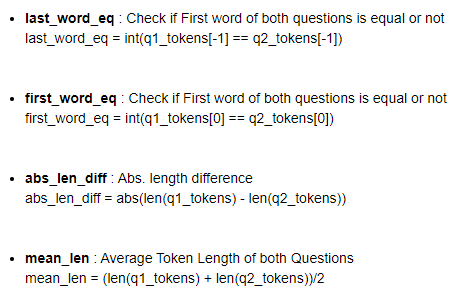


And performed some analysis on the above extracted features. And we also performed the Exploratory data analysis on the advanced features extractions, fistly we performed the preprocessing of the text data i,e



And then performed the some other feature extraction and extracted the some advanced features those are





And we other features are fuzz ratio,fuzz partial ratio, token sort ratio, token set ratio and longest substr ratio.

We performed the analysis on the advanced extracted features by plotting the word cloud on duplicate question pair and on non duplicated question pairs. And plotted the pair plot on the features ['ctc\_min', 'cwc\_min', 'csc\_min', 'token\_sort\_ratio'].

We performed the visualization on the extracted 15 features using TSNE.

Lastly we featurizing text data with tfidf weighted word to vectors, After we find TF-IDF scores, we convert each question to a weighted average of word2vec vectors by these scores. here we use a pre-trained GLOVE model which comes free with "Spacy". It is trained on Wikipedia and therefore, it is stronger in terms of word semantics. At the end total we got 384 features for each question in the dataset.

Loaded all the final data into final\_features csv file. And then coming to the implementation we loaded the data from the file into sql table for easy retrieval of data. after we read from sql table each entry was read it as a string we convert all the features into numeric before we apply any model. As because of lack of computational resources we taken the 100001 rows of data and performed the modeling on that. We randomly splitted the data into 70:30 i,e 70% train data and 30% test data.

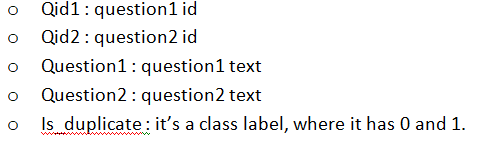
After all these we start building the random model on the data to get the worst case log loss value and then we trained the logistic regression with hyper-parameter tuning. And then trained the linear svm mode on the data with hyper-parameter tuning. At last we trained the advanced model XGBoost on the data. For all the models we plotted the confusion matrix, precession matrix and recall matrix.

**VI. Data Description**

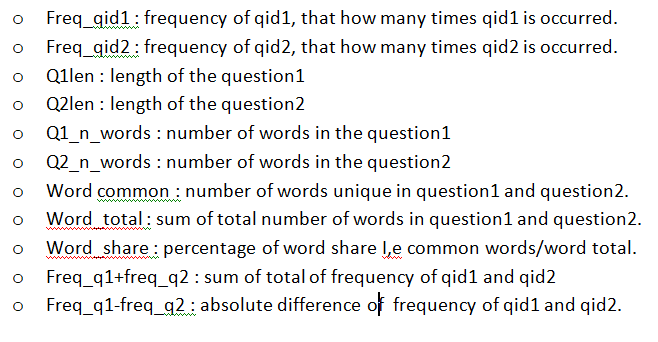
Predicting which of the offered pairs of questions has two questions with the same meaning is the objective of this competition. The labels that have been provided by human specialists are the actual facts. The ground truth labels are necessarily arbitrary because it is impossible to know with absolute certainty what a statement really means. Many sane individuals will disagree, but human labeling is also a "noisy" process. Because of this, it should be assumed that the ground truth labels on this dataset are "informed," but not entirely accurate, and may contain inaccurate labeling. Although we think the labels, as a whole, represent an acceptable consensus, this may not always be the case for specific dataset elements. Example data point i,e



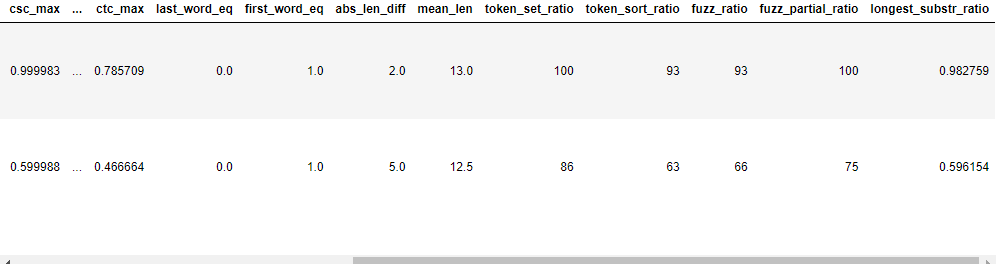
Within the considered dataset we have to total 5 features I,e



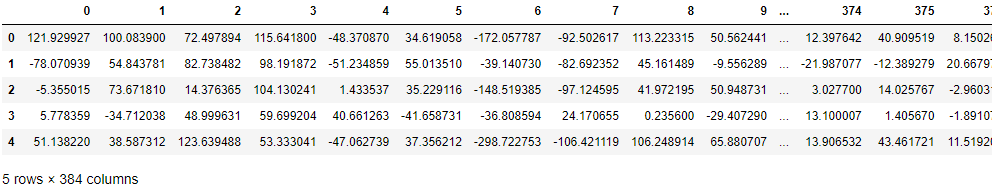
Along with this we designed few more features I,e



Advanced feature extraction are i,e

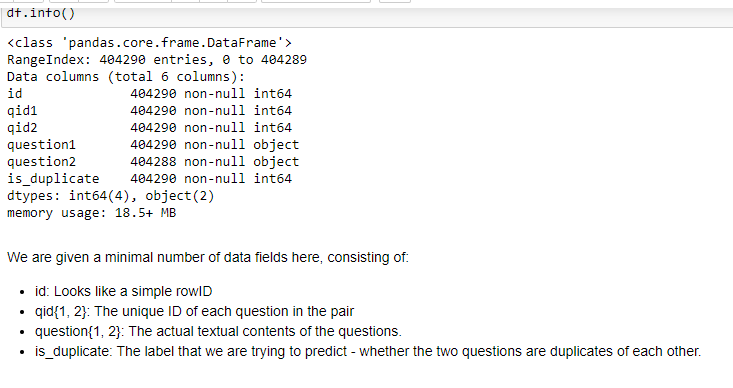


Featurizing the text data into tfidf weighted word vectors i,e

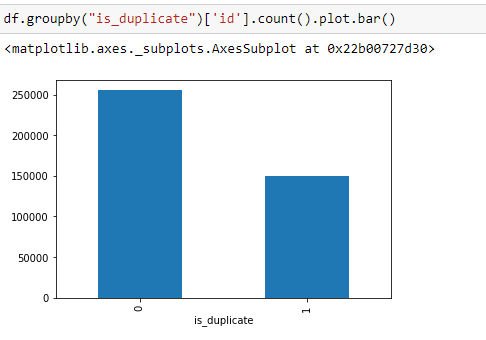


**VII. Results/Experimentation and Comparison Analysis**

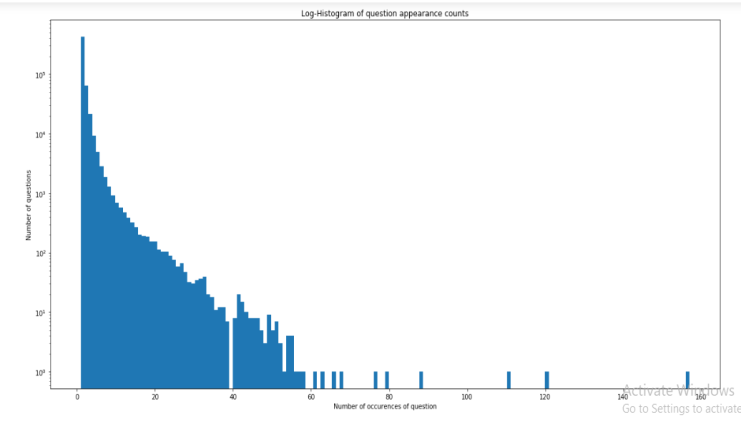
Reading the data and basic stats i,e



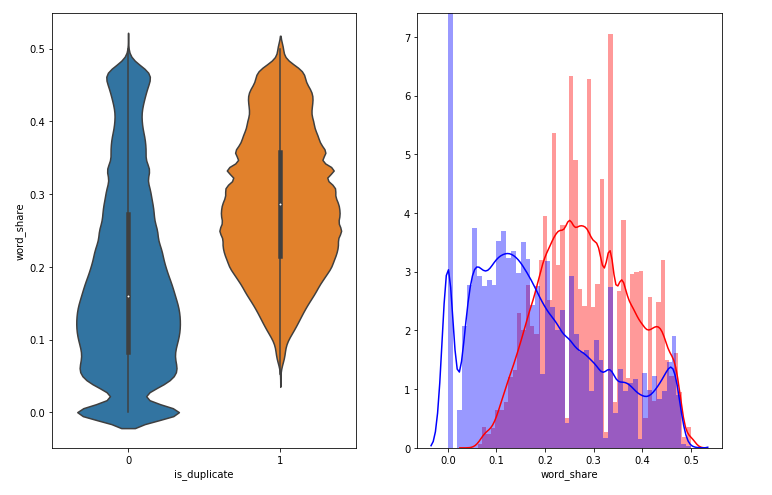
Distribution of the data points among class labels i,e

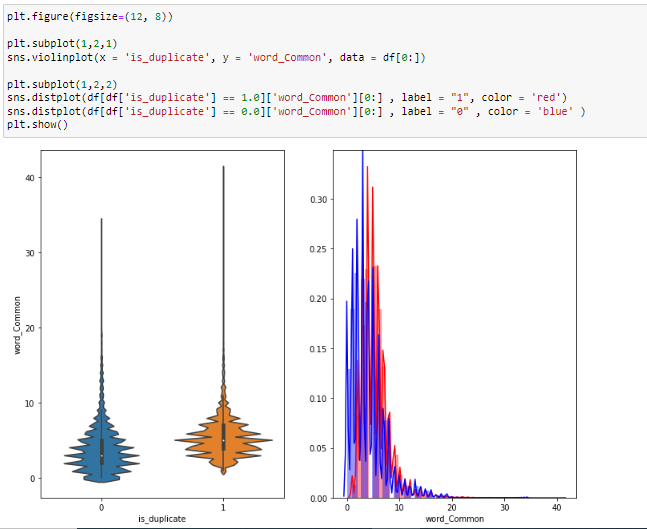


Number of occurrence of each question i,e



Analysis on the extracted features i,e



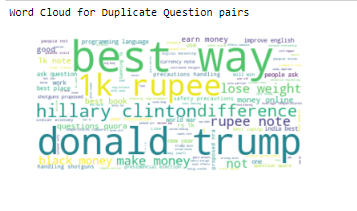


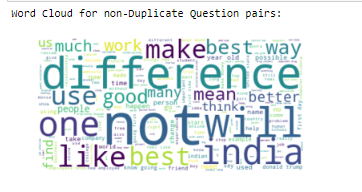
Advanced feature extraction i,e



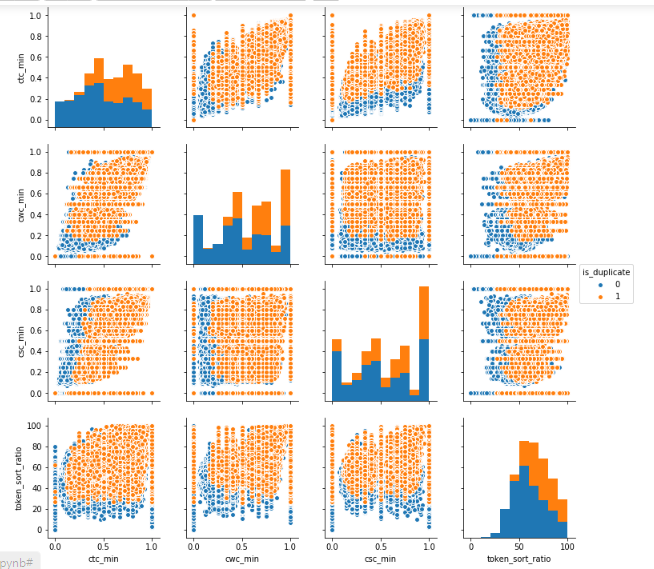
Analysis of the advanced features i,e

Word cloud for duplicate question pairs i,e

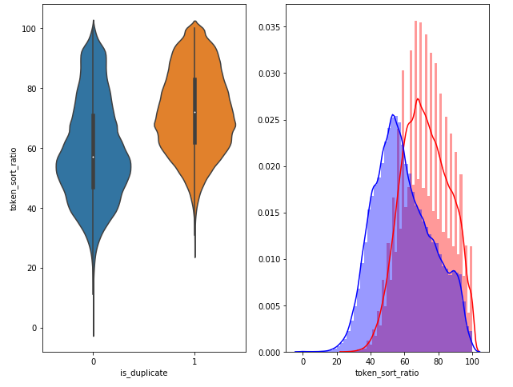




pair plot of features ['ctc\_min', 'cwc\_min', 'csc\_min', 'token\_sort\_ratio']



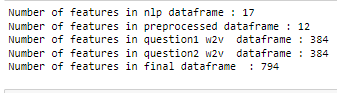
Distribution of the token sort ratio feature i,e



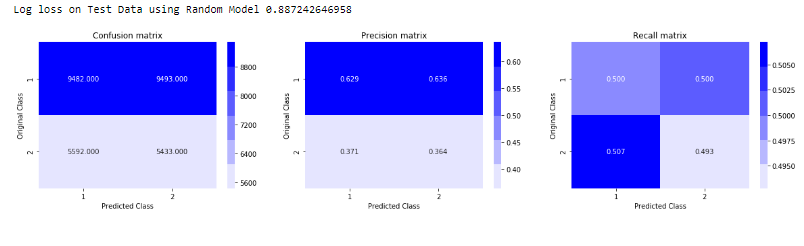
Visualization, Using TSNE for Dimensionality reduction for 15 Features(Generated after cleaning the data) to 2 dimension i,e



Featurizing text data with tfidf weighted word-vectors i,e

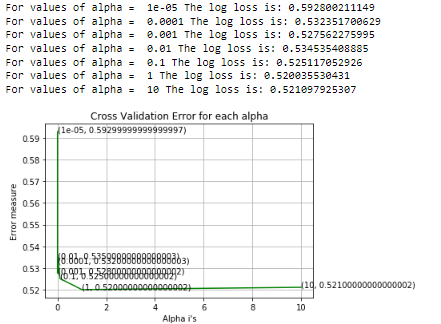


Building the random model and finding the worst case log loss i,e

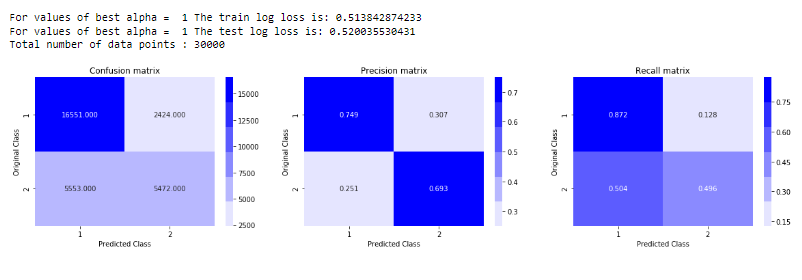


**Logistic Regression :**

Finding the best alpha value for logistic regression by doing hyper-parameter tuning we got best log loss for alpha =1 i,e

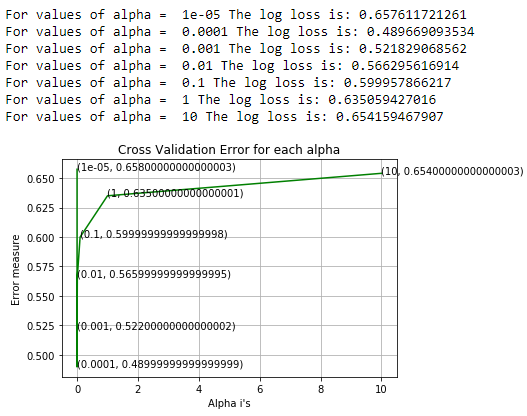


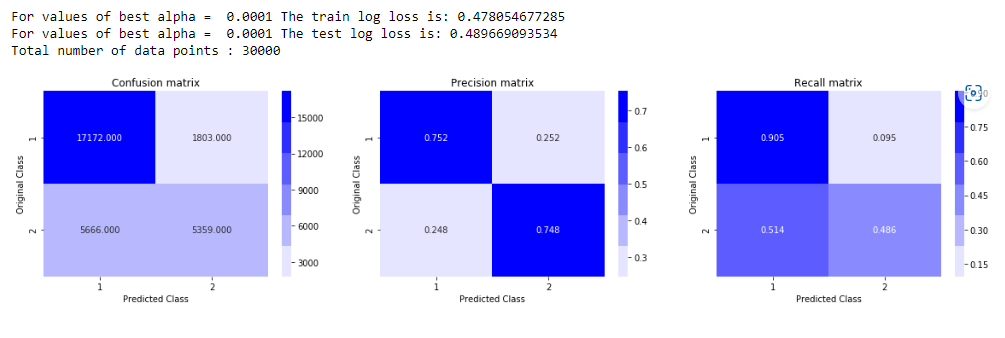
Trained the logistic regression model with alpha=1 and got the below results i,e



**Linear SVM :**

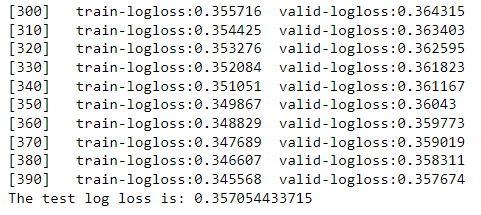
Finding the best alpha value for linear SVM by doing hyper-parameter tuning we got best log loss for alpha =0.0001 i,e

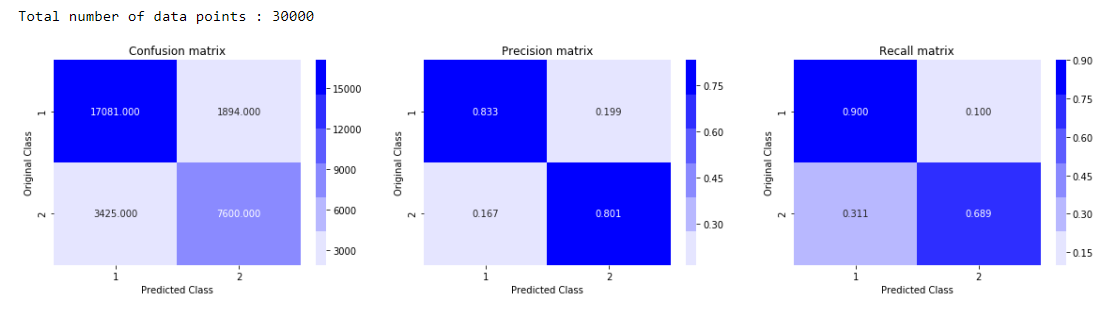




**XGBoost :**

Training the XGBoost model on the final data i,e





**Comparison/Analysis :**

With the random model we got the worst case log loss as 0.887242646958, and then we trained the logistic regression with best alpha value and got log loss as 0.520035530431. Which is certainly much better than the random model but in the precision matrix and in the recall matrix 50% datapoints which belongs to class 2 are miss-classified which is more concerning and need to take.

And in next evaluation, we trained the Linear SVM with best alpha value and we got log loss as 0.489669093534. Which is better than the logistic regression model. And if you look into the precision matrix compared to logistic regression the precision matrix score for class 2 is improved, but where as in recall matrix still 50% of the datapoints for class 2 label is miss-classified which is more concerning.

And in next evaluation, we trained the XGBoost model and got log loss as 0.357054433715. which is much better than the logistic regression and linear svm models, and precision matrix score and recall matrix score got improved more for the class label 2 which is good sign and we are looking for.

Within this experimentation the Logistic regression and linear SVM models was impacted by class bias, where the advanced model XGBoost performs well out of all models where we got less log-loss and train and test log-loss is similar only, which means model is not over fitting, and also got much good precision and recall score.

##### References

1. D. Liang- F. Zhang. W- Zhang. Q. Zhang. J. Fu. M. Peng- T. Gui and X- Huang. "Adaptive multi-attention network Incorporahng answer infonnabon for duplicate queshon detection." SIGIR 2019
2. M. Z. Ur Rehman. M. Shaheen. M. X Khan and M- Habib, "Deep leaning based semantic 5imilmity detection using text daä," Information Technology and Control, VOL 49, no. 4, pp. 495-510.2020.
3. X. Zhang, X. Sun and H. Wang, "Duplicate question identification by integrating FrameNet with neural networks;" 32nd A-AAI Conference on Artificial Intelligence. AAAI 201S- pp. 6061-6068, 201S.
4. A. Dhakal, X Poudel, S. Pandey; S. Gaire and H. P. Baral,

Deep Learmng Semantic Question Matching," Proceedings on 201S IEEE 3rd Intemational Conference on C omputing, Communication and Secmity. lcccs pp. S6-91. 201s.

1. L Sharma- L- Cnaezser, N. Nangia and U. Evci, 'Natural Languaze Understanding Tv-ith the Quora Question Pairs Dataset," 2017.
2. R: Seema„ A. Kumar and N. Kumar, "Detectug Duplicate Bi-Lingual Mash-up Queztion Pairs Using Siamese Mula-layer Pereeptroa
3. M. Mohler, R. Bunescu, and R. Mihalcea, ‘‘Learning to grade short answer questions using semantic similarity measures and dependency graph alignments,’’ in Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol., vol. 1, 2011, pp. 752–762.
4. M. Chen, X. Jin, and D. Shen, ‘‘Short text classification improved by learning multi-granularity topics,’’ in *Proc. IJCAI*, 2011, pp. 1776–1781.
5. K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, ‘‘BLEU: A method for automatic evaluation of machine translation,’’ in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*, 2002, pp. 311–318.
6. B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas, ‘‘Short text classification in Twitter to improve information filtering,’’ in *Proc. 33rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2010, pp. 841–842.
7. W. H. Gomaa and A. A. Fahmy, ‘‘A survey of text similarity approaches,’’ *Int. J. Comput. Appl.*, vol. 68, no. 13, Pp. 13–18, 2013.
8. V. Jijkoun and M. de Rijke, ‘‘Recognizing textual entailment using lexical similarity,’’ in *Proc. PASCAL Challenges Workshop Recognising Textual Entailment*, 2005, pp. 73–76.
9. F. Šarić, G. Glavaš, M. Karan, J. Šnajder, and B. D. Bašić, ‘‘TakeLab: Systems for measuring semantic text similarity,’’ in Proc. 1st Joint Conf. LexicalComput.Semantics,Conf.SharedTask,6thInt.WorkshopSemantic Eval., vol. 2, 2012, pp. 441–448.
10. R. Ferreira, R. D. Lins, F. Freitas, S. J. Simske, and M. Riss, ‘‘A new sentence similarity assessment measure based on a three-layer sentence representation,’’ in Proc. ACM Symp. Document Eng., 2014, pp. 25–34.
11. T. Mikolov, K. Chen, G. Corrado, and J. Dean. (2013). ‘‘Efficient estimation of word representations in vector space.’’ [Online]. Available: https://arxiv.org/abs/1301.3781
12. E. Gabrilovich and S. Markovitch, ‘‘Computing semantic relatedness using Wikipedia-based explicit semantic analysis,’’ in Proc. IJcAI, 2007, pp. 1606–1611.
13. Dasha Bogdanova, Cicero dos Santos, Luciano Barbosa, and Bianca Zadrozny. Detecting semantically equivalent questions in online user forums. Proceedings of the 19th Conference on Computational Natural Language Learning, 1:123–131, 2015.
14. G. Huang, C. Guo, M. J. Kusner, Y. Sun, F. Sha, and K. Q. Weinberger, ‘‘Supervised word mover’s distance,’’ in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 4862–4870.
15. Damar Adi Prabowo, Guntur Budi Herwanto. 2019. Duplicate

Question Detection in Question Answer Website using Convolutional

Neural Network. 2019 5th International Conference on Science and Technology (ICST)

1. M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, ‘‘From word embeddings to document distances,’’ in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 957–966.
2. I.Srba and M. Bielikova, “Why is Stack Overflow failing? Preserving sustainability in community question answering,” IEEE Software, vol. 33, no. 4, pp. 80–89, July 2016.
3. B. Dolan, C. Quirk, and C. Brockett, ‘‘Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources,’’ in *Proc. 20th Int. Conf. Comput. Linguistics*, 2004, p. 350.