International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-4, February 2020

A Machine Learning Model to Identify Duplicate Questions in Social Media Forums

Sandeep Kumar Panda ,Vivek Bhalerao, Sathya A.R

Abstract*: *In recent years, digital platform forums where question and answers are being discussed are attracting more number of users. Many discussions on these forums would be repetitive nature. Such duplicate questions were provided by Quora as a competition on Kaggle. It is observed that the dataset provided by Quora, requires many modifications before training machine learning models to obtain a good accuracy. These modifications include feature extraction, vectorization and tokenization after which the data is ready for training desired models. While analyzing each model after prediction, it gives plenty of information about its efficiency and many other factors. Later, these information of different models are compared and helps to choose the best model. These models later can be combined and used as a single model with best accuracy. In this paper, a Machine Learning model which will predict duplicate questions is proposed.

Keywords* : *Machine learning, Feature extraction, Vectorization, Efficiency.

I. INTRODUCTION

In present days, the proliferation of online competition is playing a vital role for academia as well as industries. Likewise, Kaggle is one of the platforms which enable anyone to learn and mentor each other on personal, academic, and professional data science journey. This platform conducts competitions, discussions, courses etc. One such open competition is posted by Quora.com [1]. Quora, itself faces challenges like, the presence of questions with same intent called as 'duplicate questions'. These questions make writers to answer in multiple versions. However, Quora uses random forest model to identify these duplicate questions. But, there is need of better model for this recognition. Therefore, this paper presents a hybrid novel approach and delivers a better solution to the problem faced by them.

Revised Manuscript Received on February 28, 2020.
***** Correspondence Author
Sandeep Kumar Panda, Department of Computer Science and Engineering, Faculty of Science and Technology, IcfaiTech, ICFAI Foundation for Higher Education, Hyderabad, India..Email: skpanda00007@gmail.com
Vivek *Bhalerao*, Department of Computer Science and Engineering, Faculty of Science and Technology, IcfaiTech, ICFAI Foundation for Higher Education, Hyderabad, India. Email: vivekbhalerao67@gmail.com
Sathya AR*, Department of Computer Science and Engineering, Faculty of Science and Technology, IcfaiTech, ICFAI Foundation for Higher Education, Hyderabad, India. Email: sathya.renu@gmail.com

Various machine learning models intended to identify these duplicate questions are presented in this paper. These identifications have specific accuracy value, using the way of obtaining the results; the model can be improvised as per Quora requirements. Hence for this recognition, three machine learning models were used, and their accuracy is analyzed using various statistical methods such as log-loss, confusion matrix. Such analysis helps to provide a better understanding and a decision to choose the best model.

II. LITERATURE SURVEY

As it is known that, reading each question provided to Quora , searching for its duplicate question and answer the same solution as the previous one is time taking and requires a lot of man power. To solve this problem Quora is currently using random forest algorithm for find duplicate question with certain accuracy. To improve its accuracy it has posted a challenge to identify duplicate questions with better accuracy. This challenge consists of a dataset and information regarding the dataset. The dataset provided is a label dataset consisting of perfect labels provided by professionals.

In the past, many research works were done in finding duplicate texts consisting of Greedy String Tiling Wise, 1996; Mihalcea et al., 2006. A paper by Torsten Zesch et al., 2012 tells that the database contain texts with different words but have same intent if it is looked as a whole text. Whereas, answering a sentence is a task of selecting a text containing the information satisfying that sentence (Lei Yu et al., 2014). Using measures for string similarities by (Lei Yu et al., 2014) proposed that, text from same field domain may contain similarities. Other techniques were also used for recognition of duplicate text which also required modification for better results.

III. DATASET

The dataset provided by Kaggle consist of six columns. These are labeled as id, qid1, qid2 , question1, question2 , Is_duplicate. Here, the dataset consist of 404351 pairs of question and each question has a unique id mentioned. However, the questions provided are repeated and can be discarded. These questions also contain many special characters and need to be analyzed before training the model. The machine learning model cannot understand words or questions in a way present in the dataset. Therefore, they need to be converted in a way such that the proposed model can take these sentences as input. The dataset also provide the information like the given questions pairs are duplicated or not.

This shows that, the model training is of supervised type of machine learning.

IV. PROPOSED METHOD

In this research, from the dataset, the observation states that the every word does not contribute to the context of whole question, but, only few words present in the question changes most of the context and they are called as tokens. As per the research requirement, tokens from online source named as "spacy-en_core_web_sm" are collected. This will act as a better input for training our models. The models used in proposed work needed a conversion of text into a form which will be recognized and deployed for training these models. Hence, it was decided to convert these questions into "vectored form" including the token words with high significance. The proposed model follows various steps as shown in Fig. 1.

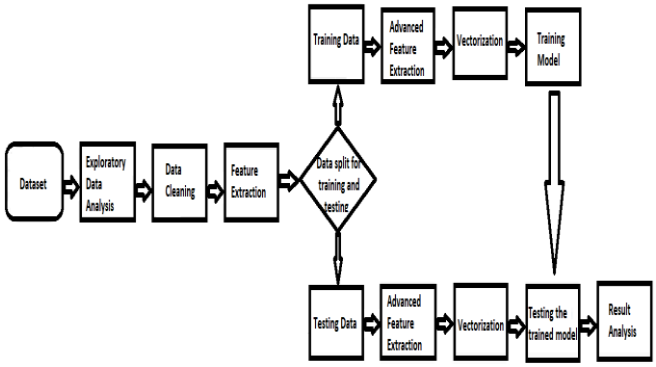

Fig. 1. Steps involved in training the model.

A. Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a method of understanding a data in every possible way and makes use of it in the way it is required to process. In this work, an EDA was made to the dataset. It gave information about the number of questions repeated i.e. complete same sentence being repeated and the number of times they have been repeated. The histogram given in Fig.2., shows repetition of a question for almost for 50 times. In few cases, few rows are completely repeated i.e. same questions and question ids. The dataset used in this work contains 149306 questions pairs which are duplicate and 255045 questions pairs which are not duplicated.

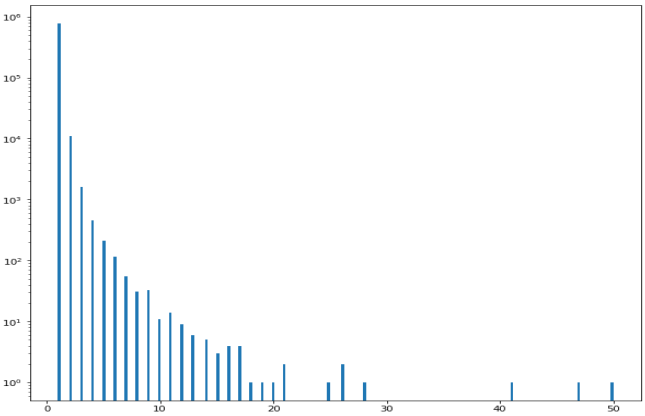

Fig. 2. Number of Repeated Questions & Repetition of Questions having same words

B. Data Cleaning or Data Filtration

The analysis obtained from the basic EDA provides the data that are not needed i.e. repeated rows. Thus, those data are to be removed from the dataset which will reduce the data size to an extent and thereby making the model faster to train. This extra data required extra memory and increases time complexity. However, the data which is distinct is maintained in the dataset. While, the repeated data being deleted.

C. Feature Extraction

Extraction phase allows the data to be observed to extract the basic features from the data. These features give a basic idea about the similarities and dissimilarities present in the question pairs. The extracted features are like frequencies of question id 1 and 2, length of question in question pairs, number of words present in question 1 and 2, number of common words, total number of words, ratio of word share. These features gave information about the available data and gave no additional information. Therefore training the model for better output is not possible with this.

D. Splitting of Data

Prior to the process of extracting "advanced features" of the available data, it is necessary to ensure that there is no "data leakage" during the training of models using the data. For this reason, the data is divided as 67% for training and 33% for testing.

E. Advanced feature extraction with EDA

The words which majorly contribute in changing the context of a question need to be the source during the training of our model. Hence, to accomplish that, usage of "stopwords" from "NLTK" called tokens were made. Later, this tokens were used in extraction of other advanced features such as number of common token words, their mean. However, modifications to abbreviations such as "can't" to "cannot" are also done. The implications of fuzzy words were done so as to match the words of same meaning. These changes were used to extract features to have more similarities if possible. These advanced features are not basically observed from existing data but extracted depending on an external sources or specific words (in this case). These features also have a great impact in the output produced by models. The following Fig.4 tells the similarities and differences between each feature with respective to other features. These features are common token count (ctc_min), common word count(cwc_min), common stopwords count(csc_min), and token sort ratio. The graph between two different features gives the distribution of duplicate and non duplicate recognized questions. Whereas, in case of graph of same feature, it is the area which tells the presence of duplicate and non duplicate question recognitions as per the feature used. Fig.3 shown below depicts the repetition of words, as the bigger the word the more number of the word is present.

Fig. 3. Repetition of same words

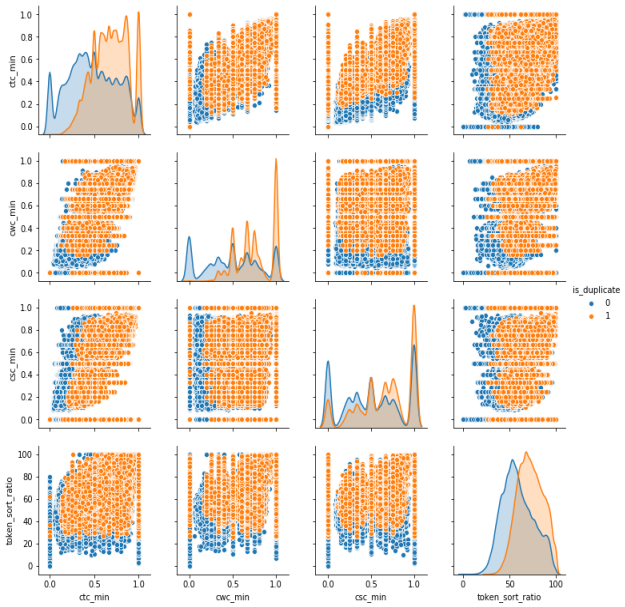

Fig.4.The plot of is_duplicate label according to the features extracted

F. Vectorization

As the system used for this work machine doesn't accept text for training, the text is converted into a form understandable by the machine. So, vectored form data is used. This vectorization is based on the "spacy-en_core_web_sm" which is an online dictionary that provides words which are used in the questions. It is implemented using "spacy" package in python. The Vectorization was done for every question present in columns "question1" and "question2" separately. Also, the questions in training and testing data (split) were vectorized separately .

G. Model selection

The most important part of this research work is to select a model which provides a prediction with better accuracy for the vectorized form of data input. Hence, it was decided to use "Naïve Bayes algorithm", "Karnough Nearest Neighbors (KNN)", "Decision tree" and "regression" as training models. These algorithms are known to produce a better output for text data. For each method, the "Grid search CV" is used to find the hyper-parameter for obtaining best result from a particular model. Therefore, three of the machine learning models is used to analyze the output. The predictions were not based on a single model but on multiple models, because each model had different error aspect.

H. Hyper-parameters

The machine learning models used here required hyper-parameter for output with better accuracy. Hence, "Grid Search CV" method is used. This method produces results as same as KNN model . i.e it had the n nearest neighbors to consider as "two" and the regression (logistic regression) value of alpha as "0.1".

V. RESULT ANALYSIS

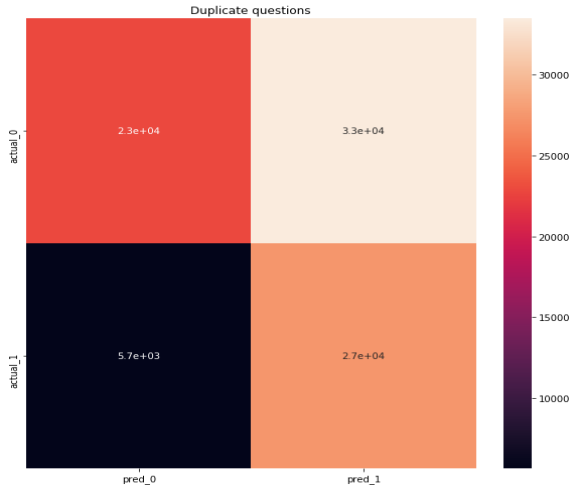

Fig. 5.a Naïve Bayes Algorithm

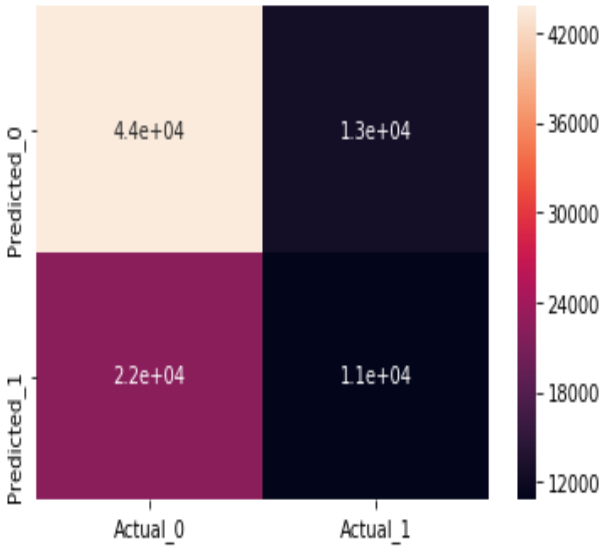

Fig.5.b Karnough Nearest Neighbor

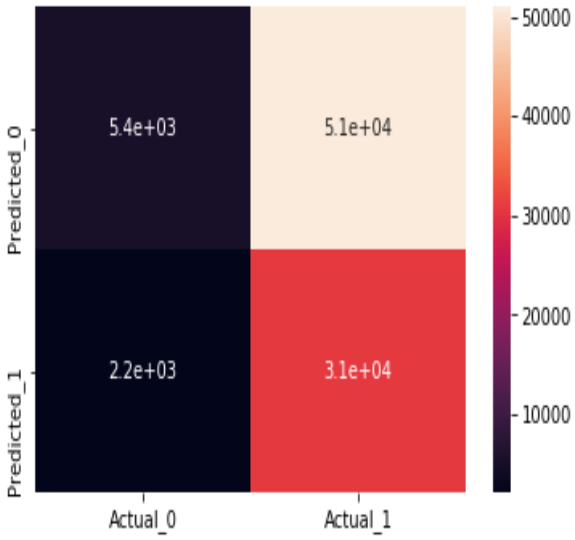

Fig.5.c Logistic Regression

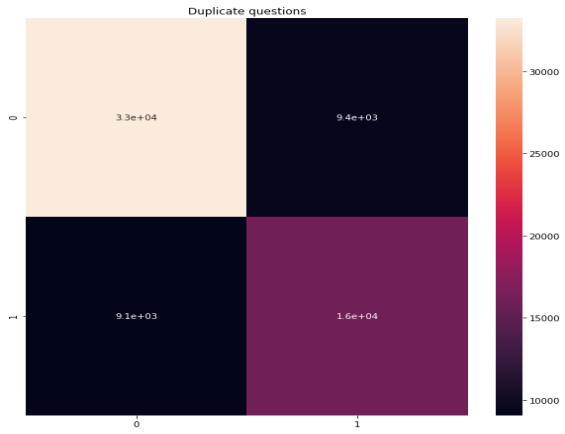

Fig.5.c Decision Tree

Table I: Result Analysis

Model	Accuracy (%)	Misclassification Rate (%)	True-Positive (%)	True-Negative (%)
Decision Tree	73	27	24	**49**
Naïve Bayes	56	44	31	**26**
KNN	61	39	12	**49**
Logistic Regression	41	59	35	**6**

Where each column heading represents the following (according an online source[3]):

Accuracy (%) = the percentage of ratio of correct predictions to the total predictions.

Misclassification Rate (%) = the percentage of ratio of incorrect predictions to the total predictions.

True Positive (%) = the percentage of ratio of number of observations is positive, and is predicted to be positive to the total number of predictions.

True Negative (%)= the percentage of ratio of number of observations is negative, and is predicted to be negative to the total number of predictions.

False Positive (%) = the percentage of ratio of number of observations is negative, and is predicted to be positive to the total number of predictions.

False negative (%)= the percentage of ratio of number of observations is positive, and is predicted to be negative to the total number of predictions.

Precision = true positive / (true positive + false positive)

Recall = true positive / (true positive + false negative)

F measure = (2*precision*recall) / (precision+recall)

Log loss: The analyses made using log loss gave us upsetting results. These were obtained as follows

Decision tree - 9.42

Naïve bayes classification -15.12

Karnough Nearest neighbor -13.14

Logistic Regression -20.14.

Therefore these values need to be reduced as much as possible.

VI. CONCLUSION AND FUTURE WORK

Hence, this research work provides good results and can be used in predicting duplicate questions for study purposes. However, few complications like, extraction of many features and vectors, heavy use of memory by .csv file or any other file has to be taken care in future work .Due to memory issues it is difficult to load and save any changes every single time. Therefore, it is better to use "pickle" form of a file for efficient use of data. In order to reduce the risk of "Data Leakage" the data can be split and be used before training the models. To obtain the best parameter rather an implementing a random parameter for the models it is suggested to use "Grid search CV" or "Random search CV". Furthermore, "XG Boost" can be utilized to provide most accurate output, in real time problem solving.

REFERENCES

1. https://www.kaggle.com/c/quora-question-pairs
2. https://dataaspirant.com/2017/05/22/random-forest-algorithm-machine-learing/
3. https://www.geeksforgeeks.org/confusion-matrix-machine-learning/
4. hanh Ngoc Dao, Troy Simpson. Measuring Similarity Between Sentences.
5. https://www.tensorflow.org/tutorials/word2ve
6. https://code.google.com/archive/p/word2vec/
7. http://machinelearningmastery.com/sequence-classification-lstmrecurrent-neural-networks-python-keras/
8. http://scikitlearn.org/stable/auto_examples/classification/plot_classifier_comparison .html
9. http://www.erogol.com/duplicate-question-detection-deep-learning/
10. https://www.linkedin.com/pulse/duplicate-quora-question-abhishekthakur
11. Torsten Zesch et al. 2012. UKP: Computing Semantic Textual Similarity by Combining Multiple Content Similarity Measures
12. Lei Yu et al. 2014. Deep Learning for Answer Sentence Selection
13. Mikhail Bilenko et al. 2003. Adaptive Duplicate Detection Using Learnable String Similarity Measures
14. Eneko Agirre et al. 2012. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity

AUTHORS PROFILE

Dr. Sandeep Kumar Panda is currently working as an Associate Professor in Department of Computer Science and Engineering, Faculty of Science and Technology at ICFAI Foundation for Higher Education (deemed to be University), Hyderabad, Telangana, INDIA. His research interests include Software Engineering, Web Engineering, Cryptography & Security, Blockchain Technology, Internet of Things and Cloud Computing. He has published many papers in international journals and in international conferences. He received "Research and Innovation of the Year Award" from WIEF and EduSkills on January 2020. He has Six Indian Patents in his credit. His professional affiliations are MIEEE, MACM and LMIAENG.

Vivek Bhalerao, pursuing his Bachler's in Computer Science and Engineering at Faculty of Science and Technology, Icfai Tech, Icfai Foundation for Higher Education(IFHE), Deemed to be University, Hyderabad.

Sathya AR , completed her Master's degree in Computer Science and Engineering at Anna University, Tamilnadu, India 2007. Her research interests include Software Engineering, Blockchain Technology and Machine Learning. She is currently working as an Assistant Professor at ICFAI Foundation for Higher Education (IFHE), Deemed to be University, Hyderabad.