**Identifying Question Intent Similarity**

|  |  |  |
| --- | --- | --- |
| N.BalaKrishna,  Assistant Professor, Department of CSE  Sree Vidyanikethan Engineering College,  Tirupati, Andhra Pradesh, India  [balu1203@gmail.com](mailto:balu1203@gmail.com) | Dr. M B Mukesh Krishnan,  Professor, Department of Networking and Communications,  SRMIST | V. Abhishek,  UG Scholar, Department of CSE  Sree Vidyanikethan Engineering College,  Tirupati, Andhra Pradesh, India  [abhishekshek442000@gmail.com](mailto:samanumanvithareddy@gmail.com) |
| V. Dheeraj,  UG Scholar, Department of CSE  Sree Vidyanikethan Engineering College,  Tirupati, Andhra Pradesh, India  [dheeraj.vem@gmail.com](mailto:samanumanvithareddy@gmail.com) | V. Kushal  UG Scholar, Department of CSE  Sree Vidyanikethan Engineering College,  Tirupati, Andhra Pradesh, India  [chantikushal@gmail.com](mailto:samanumanvithareddy@gmail.com) | V. Sai Rakesh Reddy  *UG Scholar, Department of CSE*  *Sree Vidyanikethan Engineering College,*  Tirupati, Andhra Pradesh, India  [sairakeshreddy.v@gmail.com](mailto:sairakeshreddy.v@gmail.com) |

1. **ABSTRACT**

To predict and Identify Question Intent Similarity within a sample dataset consisting question pairs. It is to improve upon and streamline the process of predicting duplicate question pairs in the given data. To perform the prediction, we use XGBoost and bag of words machine learning techniques, specifically bag of words, which is used in natural language processing. We evaluate our system through a dataset that consists of duplicate questions to emulate human interaction within the given data. To establish a base score, we perform Random Forest tree algorithm on the system then switch to bag of words individually for each question pair. Considerable growth in accuracy can be noted from the findings. We also take repeating questions into consideration to demonstrate the number of times a question pair has repeated. Words common, total number of words and words shared (A total estimate) are taken as features to further expound on the similar question pairs. This demonstration within our consensus should prove to be computationally efficient through the use of Bag of words and XGBoost.

**KEYWORDS**

Bag of words (BoW), XGBoost, question pairs, Random Forest tree, natural language processing, machine learning.

1. **INTRODUCTION**

Since man became self-aware of his existence, he has embarked on a journey, out of sheer curiosity in search of truth. This urge to quench one’s thirst for knowledge and imagination has brought humans together into complex formations; from the early man’s socializing around fire to complex structures built by cultures around the world to discuss and debate their ideologies or to simply have one of similar distinction closer to them. Contemporary technologies have ensured our search for truth can begin right out of our pockets with mobile devices connected to the Internet, using this technology as a basis we have developed sophisticated solutions to connect with our peers and the world. The marvel of the 21st century being ‘Social – media’ a solution that revolutionized communication, making it seamless and remotely accessible to everyone. Quora and stack overflow are two of many such social media solutions that specifically specializes as a question-answer forum for users to post questions and receive answers. Though the idea behind it is simple we must understand that as the volume of users increase so will the number of questions and answers on the website increase by lakhs. We found that when users post a question, if it is relevant, it is always the case that someone else has the same question but in a different format. For a more linear approach it is necessary to streamline the user-experience to find answers for similarly intended questions present on the website. It saves time and reduces clutter on the system.

Multiple questions with identical words can cause readers to spend more time searching for appropriate responses and force authors to respond to multiple replies to the same question. For example, questions like ``How can I improve my photography skills?'' and ``What should I do to improve my photography skills?'' are identical because they both imply the same thing and should only be replied once. Some questions, like `` What is your age?'' and ``How old are you?'' have different phrasing. However, the context still stays the same. Consequently, these queries are also considered duplicates and candidates for merging, it minimizes duplicated content, simplifies the process of finding answers, decreases the number of unanswered questions and cuts down the time and energy required for users to search for information. Thus, users can obtain answers to all inquiries, and writers can avoid duplicating responses across multiple locations for the same questions. Quora currently uses Random Forest and decision tree with many hand-crafted features to integrate similar questions into a single query. This model will process vast quantities of data. In 2022, inspired by advances in deep learning and machine learning models, Quora hosted a contest on Kaggle. The participants were tasked with applying advanced Machine Learning techniques to the dataset to improve the results' reliability and precision. By employing advanced Neural Network Architecture, this study seeks to achieve the same goal of attaining greater accuracy.

1. **LITERATURE SURVEY**

Identifying Question Intent Similarity, heretofore called as identifying similar question pairs, is an important task in natural language processing (NLP) with numerous applications, including duplicate detection in question-answering systems like stack overflow and community question-answering such as Quora. In early work multiple machine learning methods have been discussed for this problem but recently the emergence of complex technologies utilizing the deep learning approach has made it efficient to study and learn from large subsets of data, and this literature survey will deliver an overview of these methods.

One of the earliest approaches to identifying similar question pairs is to implement string similarity measures, such as Levenshtein distance, Jaccard similarity, and cosine similarity. These measures are simple to implement and can be effective for capturing the surface-level similarity between questions. However, these measures are limited in capturing the semantic similarity between questions. Later, supervised techniques of machine learning including logistic regression and support vector machines (SVM) and neural networks have been proposed for differentiating the similitude between question pairs. These approaches typically rely on feature engineering to represent the questions and use labeled data to train a model to predict the similarity between question pairs. Semantic analysis of sentences has traditionally focused on logical inference, Inference Corpus of Stanford Natural Language. Which used LSTMs to concentrate word-by-word recognition methods. Neural network methodologies provide for a larger selection of NLP tasks. For instance, Sankar et al. (2013) used features such as word overlap, IDF (inverse document frequency), and word2vec embeddings to identify similar question pairs. They compared a number of ensembles of machine learning algorithms, including random forest, support vector machine and decision tree and discovered that SVM performed better than the other algorithms. They also noted that the addition of word embeddings and IDF features improved the performance of the model.

Recently, deep learning-based methods have shown promising results in identifying similar question pairs. These approaches typically use neural network architectures to learn a similarity function directly from the raw text of questions, without feature extraction by humans. This reduces time and increases efficiency in the long-term. For instance, Wang et al. (2017) used Siamese Networks, however during the inception of this technique the Siamese architecture whilst being lightweight and easily trainable, there is clearly a lesser impact of correlation between the parameters due to which information is lost. To overcome the problem to be able to capture two sentences the Compare-Aggregate model was proposed, which involve using two identical neural networks with shared weights to encode each of the input questions. The outputs of the networks, then compared using a similarity function to determine the similitude between the question pairs. They reported high accuracy in detecting similar question pairs on various datasets. This ensured that the restrictions of the Siamese framework were triumphed. Similarly, Chen et al. (2019) used a combination of CNNs and RNNs to encode the questions and predict the similarity between question pairs. Taking notes from a previous work by Yin et al(2015), where they understood neural networks (RNNs) and convolutional neural networks (CNN or convnet) together to encode the questions and predict the similarity between question pairs. They used word embeddings that had already been trained and said that they were very good at finding similar question pairs in the Quora question pairs dataset.

Finally, transformer-based models have recently achieved state-of-the-art performance in identifying similar question pairs. These models, such as BERT, ALBERT, are pre-trained on large amounts of text data and can be fine-tuned on a specific question similarity task to achieve high accuracy. Devlin et al. (2018) proposed BERT, which is pre-trained on a large corpus of text data using a masked language modeling task and a next sentence prediction task.

In summary, Identifying Question Intent Similarity using machine learning and deep learning techniques is an active research field, with a multitude of different approaches being explored in parallel with existing methodologies. Although the choice of method has fared well when the specific task and resource availability has been taken into consideration, there is still a window for improvement in establishing a more accurate and efficient result.

1. **PROBLEM STATEMENT AND DATA**
2. **Problem statement**

The purpose of this paper is to observe prediction of question pair duplicity i.e., is to identify questions that are similar in a given sample and map them to equally coherent questions that are of the same nature. This matches as an extension for the answers that were received by a duplicate question or similar question, of the same nature, through this process we reduce the complexity while addressing large data.

1. **Objectives**

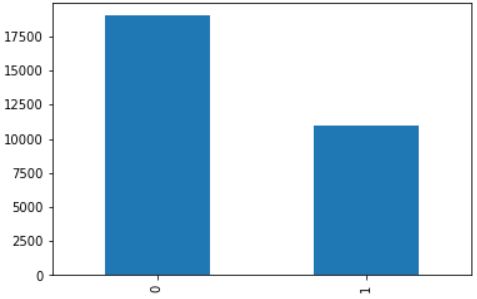
* Understand and implement various pre-processing techniques on textual data.
* Perform feature engineering & hyperparameter tuning to improve the model performance.
* Explore various tree-based algorithms like XGBoost and Bag of word.

1. **Data**

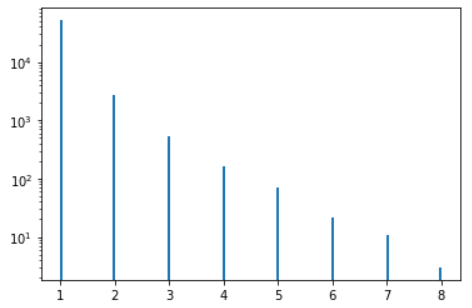
The sample data we have was collected and shared by Kaggle on behalf of Quora. The ground truth is the list of labels that experts have given to each question-answer pair. The ground truth labels are always subjective because it's impossible to know for sure what a sentence means. Labelling people is a "noisy" process, and reasonable people will have different opinions about it. Because of this, the ground truth labels on the dataset are thought to be "informed," but they are not always correct and may contain wrong labels. We think that the sample data is a reasonable consensus on how to solve the problem, but this may not always be true for each item in the dataset.

**Data fields**

* **id** – the id of a training set question pair.
* **qid1, qid2** – unique ids of each question. (Only available in train.csv).
* **question1, question2** – each question's full text.
* **is\_duplicate** – the target variable, which should be set to 1 if question1 and question2 have almost the same meaning and 0 if they don't.



**Figure 1.** The class disparity between non - duplicate (0) and duplicate (1) questions in the dataset.



**Figure 2.** The frequency of repeating questions observed in the dataset.

1. **XGBoost**

XGBoost is a distributed gradient boosting library optimized for scalable and efficient training of machine learning techniques. It utilizes ensemble learning that integrates the predictions of multiple weak models to generate a stronger prediction as per our requirements. XGBoost, which is an abbreviation of "Extreme Gradient Boosting," it has gained prominence in contemporary use with machine learning models due to its propensity to manage large datasets and achieve a level of performance in machine learning techniques including techniques such as classification and regression. It was initially conceived by Tianqi Chen as a research project later it took off to become a regularizing gradient boosting framework for several platforms such as C++, Scala, R, Python, Julia, Perl and Java. It runs on distributed processing frameworks at the same time on a single machine, this speaks volumes in terms of flexibility. Some of those frameworks are as follows: Apache, Hadoop, Apache Spark, Apache Flink, etc.

XGBoost is efficient in managing missing values without comprehensive preprocessing. It has built-in support for parallel processing, allowing for models to be trained over larger datasets within sensible timeframes. it is highly customizable and permits the fine-tuning of various model parameters to optimize performance. Gradient-boosted decision trees are implemented through XGBoost. It places a significant emphasis on weights, by doing that it makes feature analysis a more rewarding processing for improving the performance of the machine learning techniques employed in the project. All independent variables possess weights, which are then incorporated into the decision tree used to predict outcomes. The weight of variables incorrectly predicted by the tree is increased, and then these variables are transmitted to a second decision tree. These individual classifiers/predictors are then combined to create a model that is robust and more accurate. It is applicable to regression, classification, ranking, and user-defined prediction issues. We utilize it after applying BoW to streamline the dataset. It renders the best performance through which predicting the multiple question pair monogamy becomes quite easy which would be considered a tiresome and time-consuming task to even show any distinction from current parameters available to us.

1. **BAG OF WORDS**

In this paper, we will talk about the Bag of Words model, which is a way to model text using Natural Language Processing. In NLP, every algorithm we use works on numbers. We can't just put our own writing into that algorithm. So, the Bag of Words model is used to preprocess text by turning it into a bag of words, which keeps track of how many times the most common words are used.

This model can be seen in a table, which shows the number of times each word appears in the table.

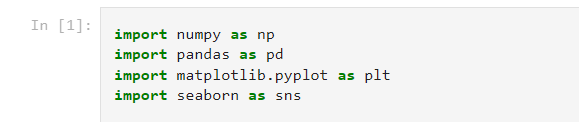
The Bag of Words (BoW) is a text representation model frequently employed in natural language processing (NLP) tasks such as query pair similarity. In BoW, each document is represented as a collection of words, and the frequency of each word is used to create a vector representation of the document.

To create a BoW representation of a question, we first tokenize the question by splitting it into individual words. Then, we remove stop terms, which are prevalent words with little meaning, such as "the" or "and". Finally, we create a vector representation of the question by counting the frequency of each word in the question and mapping it to a unique index in a vocabulary.

BoW does come with one limitation that it does not take the order or context/relevance of the word in the document. This means that two questions with the same words in a different order or with different synonyms may have a low similarity score. To address this limitation, other models such as word embeddings, which capture the meaning of words in a more nuanced way, can be used in combination with BoW.

1. **RESULT AND OBSERVATION**

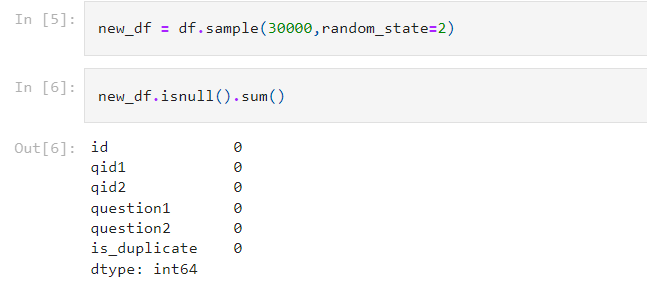
**IMPORTING LIBRARIES:**



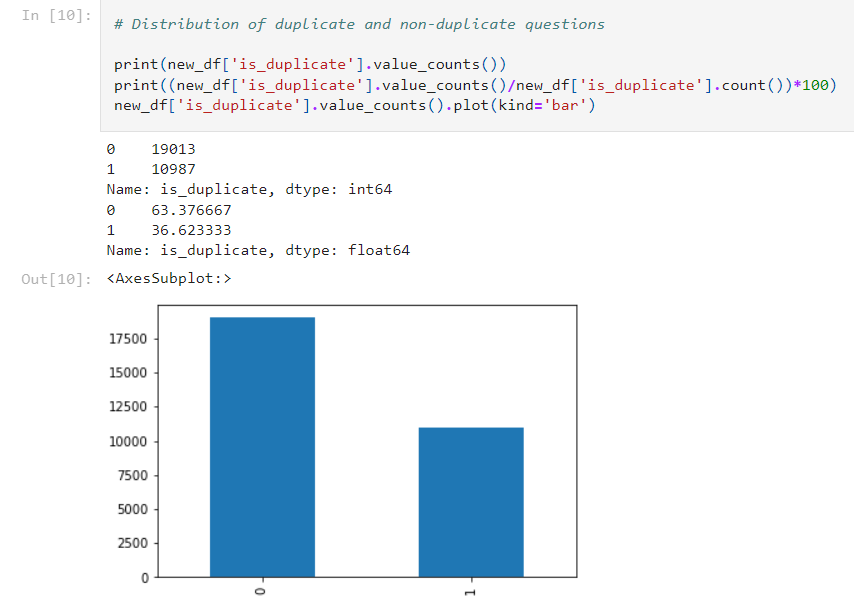
**IMPORTING THE DATASET:**



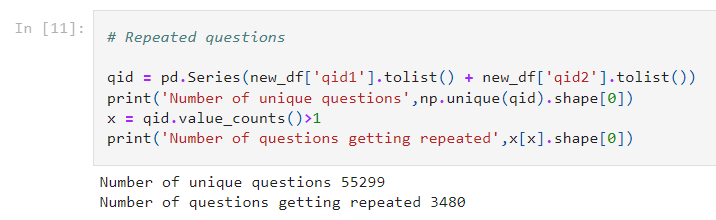
**TOOK A SAMPLE DATA AND CHECKING FOR NULL VALUES:**



**DUPLICATE AND NON-DUPLICATE QUESTIONS:**

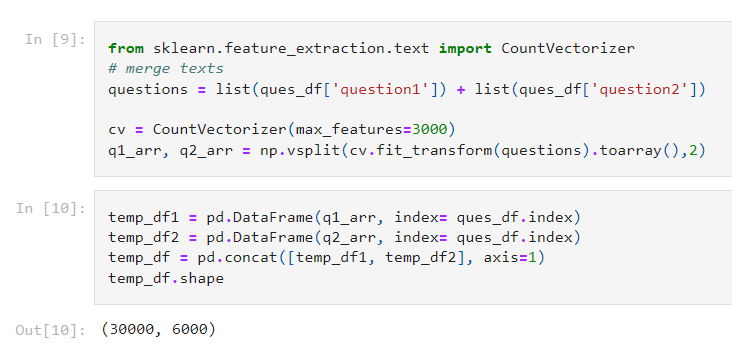


**NUMBER OF UNIQUE AND REPEATED QUESTIONS:**



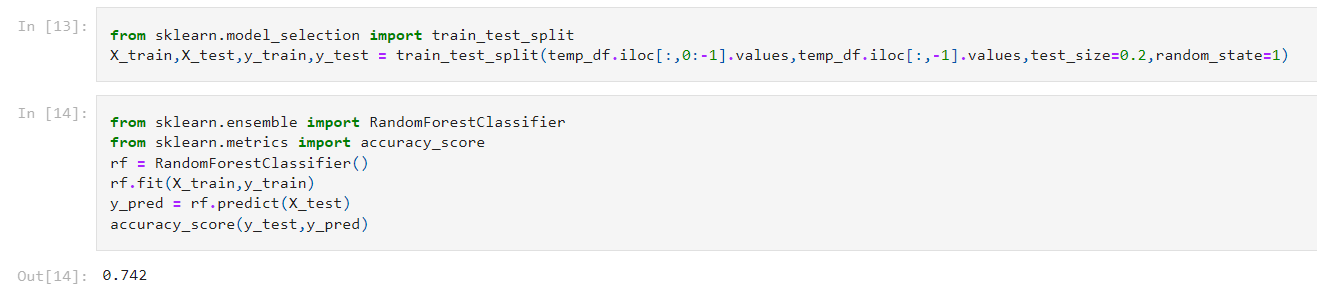
**BEFORE FEATURE ENGINEERING:**

**FEATURE EXTRCTION PROCESS:**

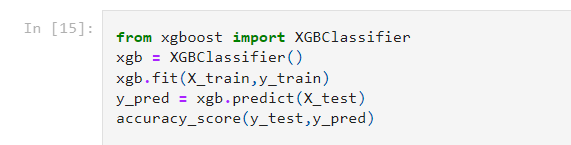


**TRAINING AND ACCURACY PREDICTION:**

**ACCURACY USING RANDOMFOREST**:

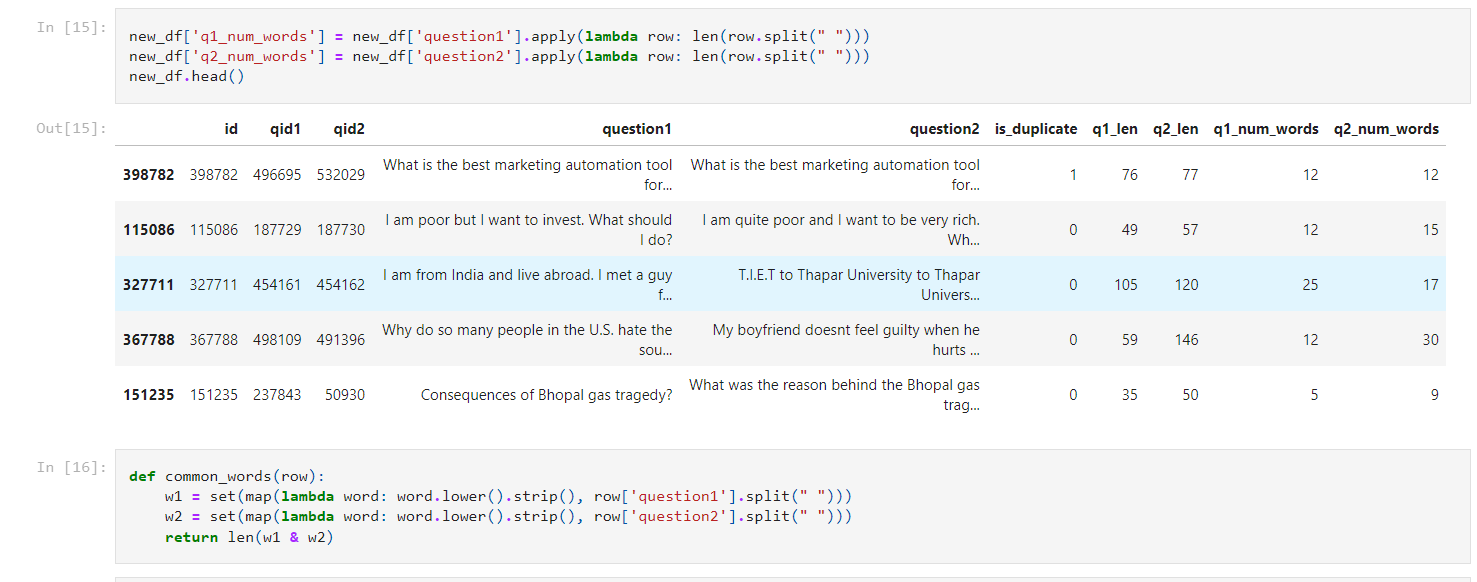


**ACCURACY USING XGBOOST:**

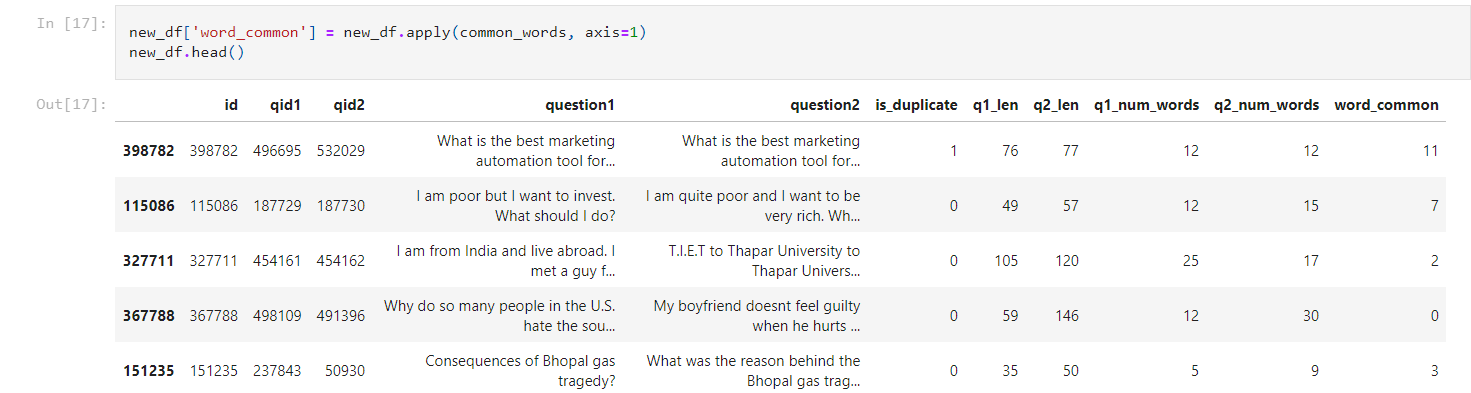




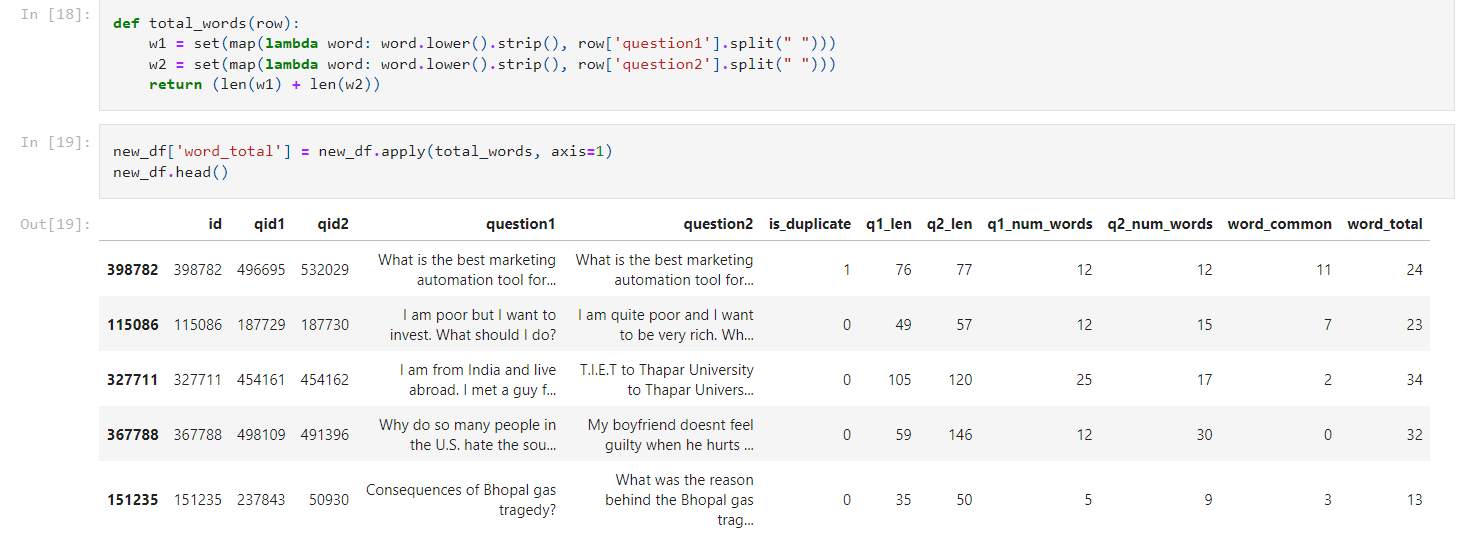
**FEATURE ENGINEERING PROCESS:**

**Adding features like Length of question1 and Length of question2:**

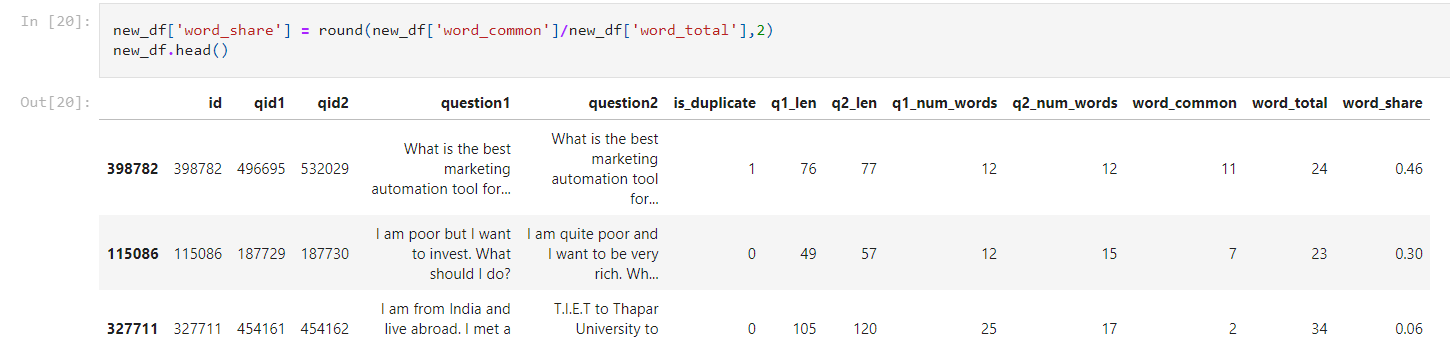
**Words common:**



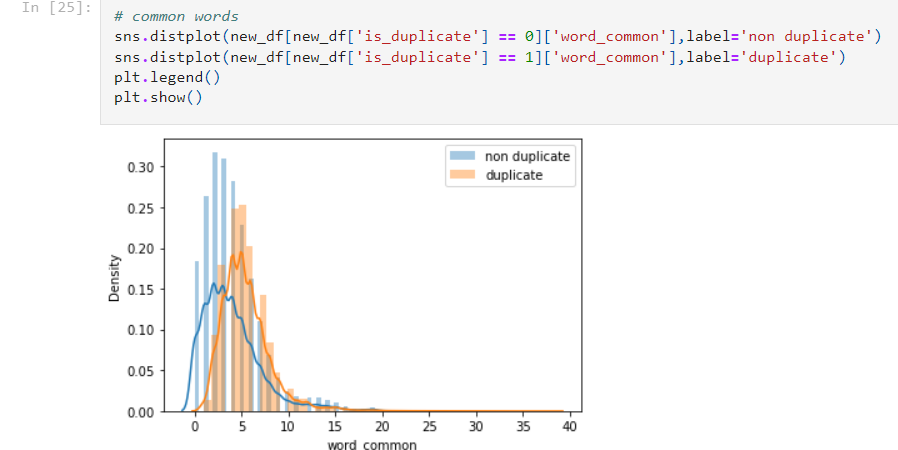
**Total number of words:**

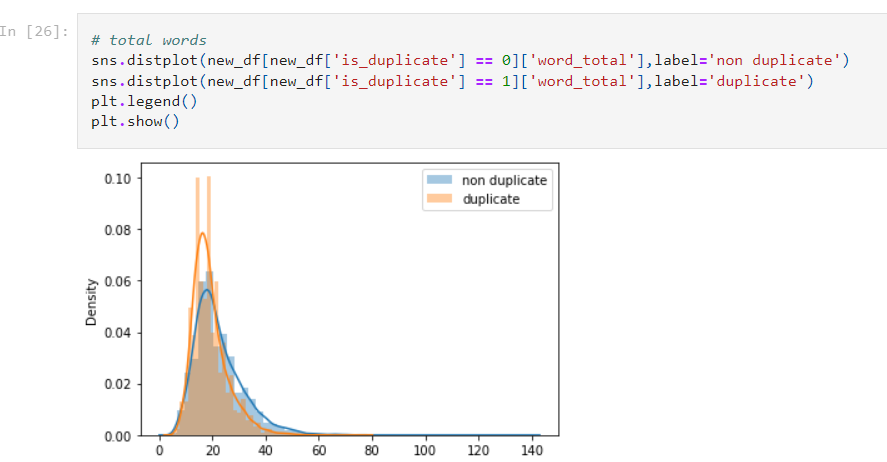


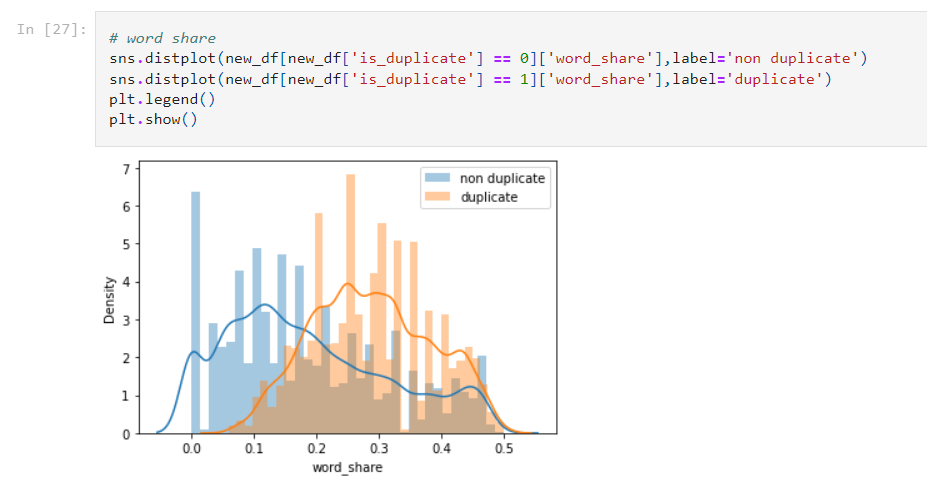
**Number of words shared:**



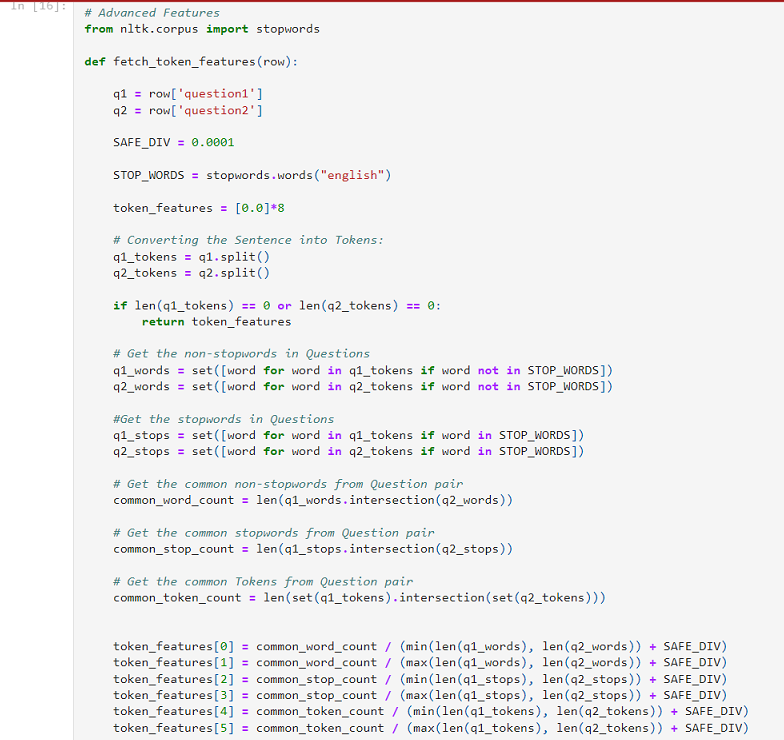
**FEATURE ANALYSIS:**



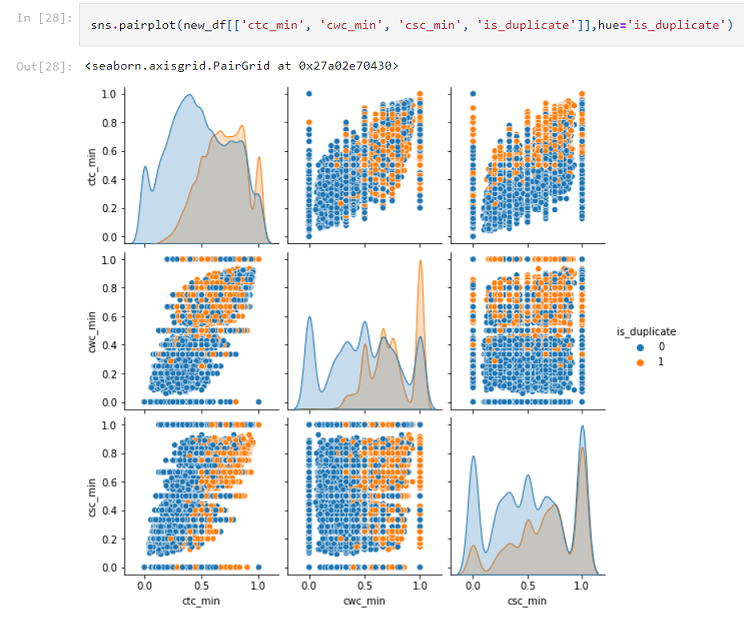


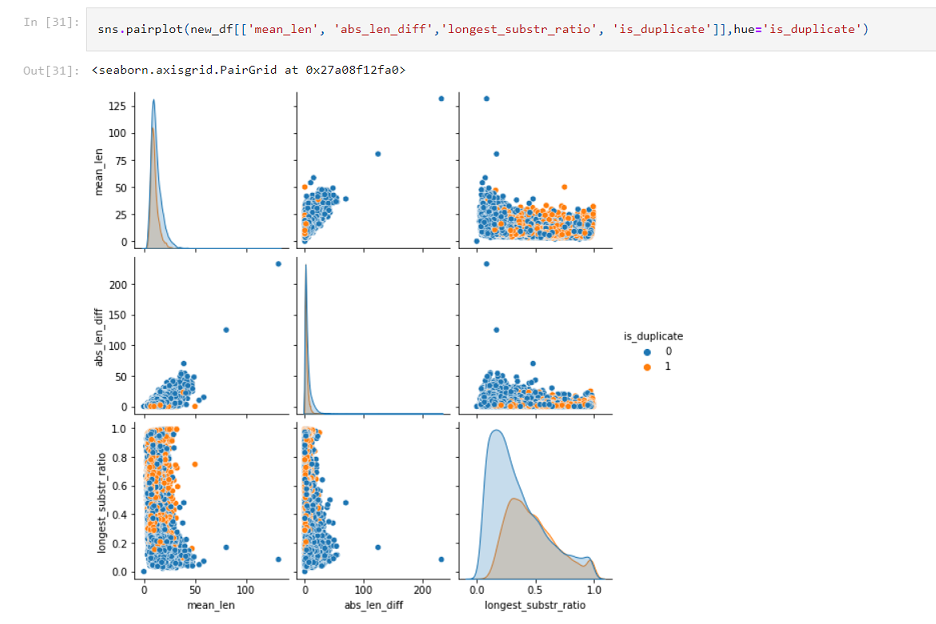


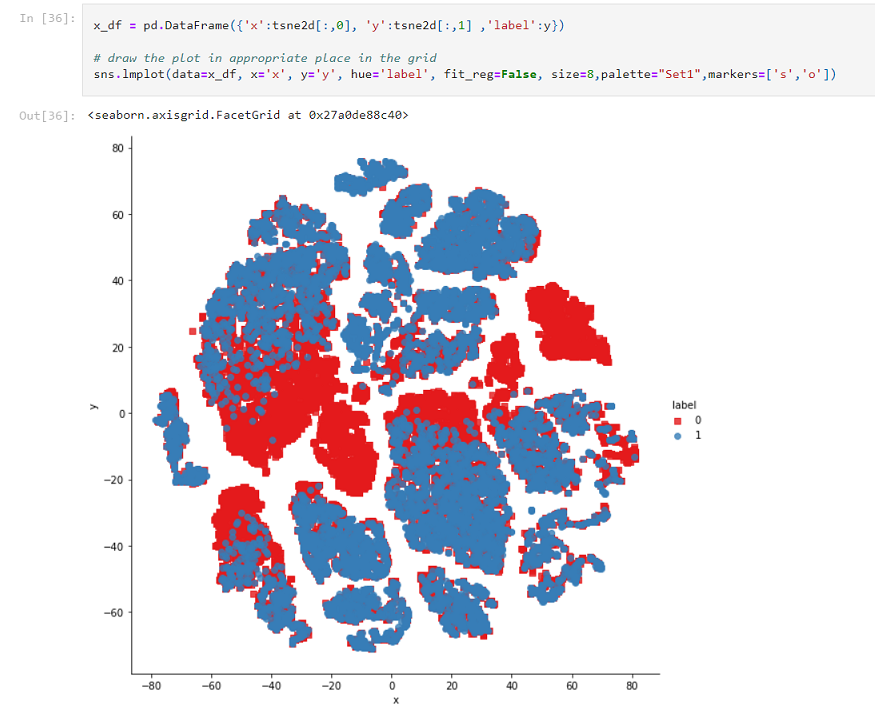
**ADDING ADVANCED FEATURES:**



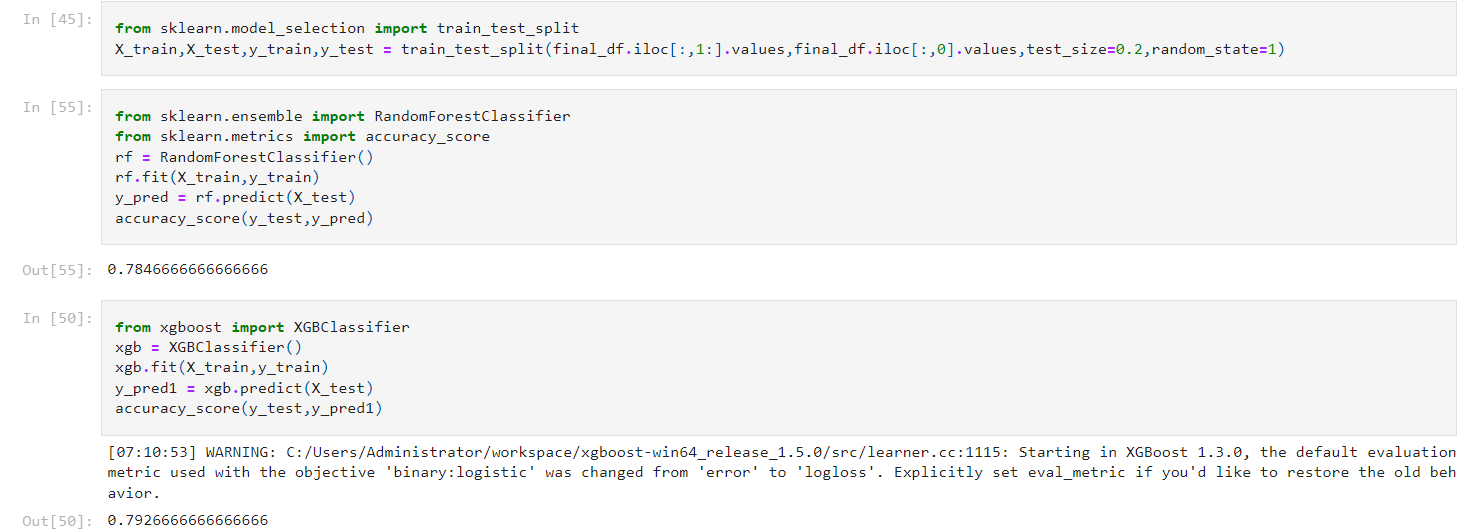
**DATA ANALYSIS OF ADVACED FEATURES:**



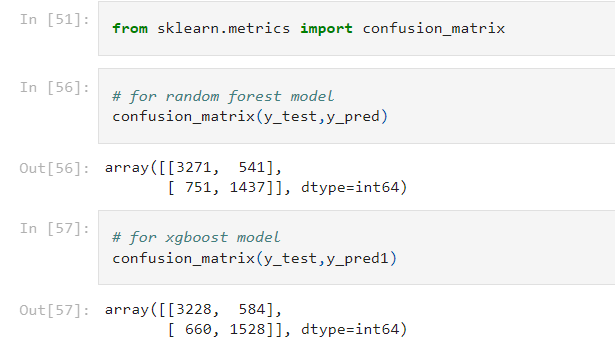




**ACCURACIES AFTER FEATURE ENGINEERING:**



**CONFUSION MATRIX:**



1. **CONCLUSION**

In this paper, the prediction accuracy of XGBoost and Bag-of-words to find the duplicate question pair has shown significant improvement that can be noted regarding repeating questions in the given dataset, it was observed that repeating questions occurred at various intervals. The similitude of the questions in the given sample data have been satisfactory with meeting required results.

1. **REFERENCES**
2. Zainab Imtiaz, Muhammad Umar, Muhammad Ahmad, Saleem Ullah, Gyu Sang Choi,Arif Mehmood,” Duplicate Questions Pair Detection Using Siamese MaLSTM”,IEEE, Access, Vol.8, 21, 21932 \_ 21942, Jan, 2020.
3. Kratika Sharma, Satya Kiranmai Tadepalli, ” Detecting Duplicate Questions in Online Forums Using Machine Learning Techniques ”, IJRASET, Access, Vol.10,16, 29, June, 2021.
4. Huong T. Le, Dung T. Cao,” Improve Quora Question Pair Dataset for Question Similarity Task”, IEEE, Access, Vol.2, 18, 20, Dec, 2021.
5. V K Raj Anishaa, P Sathvika, Sandeep Rawat, ” Identifying Similar Question Pairs Using Machine Learning Techniques”,IJST,Access,vol.4,20,1635\_1641,May,2021.
6. P. Sravanthi and B. Srinivasu, ``Semantic similarity between sentences,'' Int. Res. J. Eng. Technol., vol. 4, no. 1, pp. 156\_161, 2017.
7. R. Johnson and T. Zhang, ``Supervised and semi-supervised text categorization using LSTM for region embeddings,'' in Proc. ICML, 2016, pp. 1\_9.
8. D. Tang, B. Qin, and T. Liu, ``Document modeling with gated recurrent neural network for sentiment classi\_cation,'' in Proc. Conf. Empir-ical Methods Natural Lang. Process., Lisbon, Portugal, Sep. 2015, pp. 1422\_1432.
9. K. S. Tai, R. Socher, and C. D. Manning, ``Improved semantic representations from tree-structured long short-term memory networks,'' in Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process., Beijing, China, vol. 1, Jul. 2015, pp. 1556\_1566.
10. R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler, ``Skip-thought vectors,'' CoRR, vol. abs/1506.06726, pp. 1\_11, Jun. 2015.
11. Y. Kim, ``Convolutional neural networks for sentence classi\_cation,'' in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), Doha,Qatar, 2014, pp. 1746\_1751