Duplicate Question Pair Identification on Quora

Shwetank Chaudhary (*Author*)
Department of Information Technology
Bharati Vidyapeeth's College of Engineering
New Delhi

Dr. Surinder Kaur (*Mentor*)
Department of Information Technology
Bharati Vidyapeeth's College of Engineering
New Delhi

***Abstract*— Determining semantic equivalence between pairs of questions is crucial due to the vast number of users on Quora, leading to numerous questions with similar intents and potential duplicates. Efficiently identifying duplicate questions is essential for improving the quality of answers and saving users' time, thereby enhancing the overall user experience on Quora for both writers and readers. In this research, we investigate methods to address this challenge using a dataset from Quora. Our approach involves employing various classifiers, such as Random Forest and XGBoost, along with comprehensive feature engineering techniques. By leveraging ensemble methods that combine these classifiers, we achieve superior performance in distinguishing between related and duplicate questions compared to using individual classifiers alone. This study contributes to advancing the field of semantic equivalence detection in question pairs, with practical implications for enhancing user experience on question-and-answer platforms like Quora.**

Keywords*— *Quora, Similarity Detection, XGBoost, Light GBM, Logistic Regression, Random Forest, Ensemble models, Hyperparameter tuning, EDA

I. INTRODUCTION

One of the most crucial tasks which can be achieved by effective natural language processing is the task of evaluating sentence similarity. Determining sentence similarity between a pair of sentences finds many applications such as automated short-answer grading and machine translation [7] to name a few. However, it has been gaining traction in many question-answer and knowledge-sharing applications such as Quora [15], Reddit [16] and Stack Overflow [24] as they gain popularity. As the user base of these applications grow, the magnitude of questions and answers in their archives also grows to the possibility that a question asked by a user has already been asked before and answered. In such a scenario, it would be ideal for the application to suggest previously asked questions similar to the one which the user wishes to ask in order to improve user experience and enable efficient knowledge-sharing. This would save waiting time for the user and reduce frustration which can occur if the user sees different versions of the same question on his feed thus preventing display of other, more relevant content. Additionally, ensuring that duplicate or similar questions are not asked repeatedly reduces the burden on the storage and processing infrastructure of the application. However, it is difficult to find similar questions due to limitations of current search engines. Measuring question similarity using traditional document similarity techniques, which rely heavily on overlap, is not suitable for small question pairs due to their limited overlap. Additionally, questions can be phrased in multiple ways because of the nature of language generation, leading to further challenges. Therefore, understanding different phrasings of questions can enhance search and discovery methods.

In this report, we discuss different methods for assessing the similarity of question pairs sourced from Quora, considering a pair duplicate if they convey the same intent. Our initial approach employs established classifiers like XGBoost [20], Logistic Regression, and Random Forests to categorize question pairs based on similarity. We utilize a diverse feature set comprising similarity-based, graph-based, and language-based features as inputs to these classifiers, achieving notable performance. Additionally, we investigate an ensemble model that combines these classifiers to assess question pair similarity effectively.

The report is structured as follows: first, we review previous work on sentence similarity measurement and outline the techniques we will use in our methods. Then, we present detailed descriptions of our models and approaches, focusing on XGBoost [20], Random Forest, and Logistic Regression. Finally, we discuss our results and observations from these methods.

II. RELATED WORK

Understanding and classifying duplicate questions presents a challenge due to language variability, akin to the paraphrase identification problem, extensively studied in Natural Language Processing (NLP) [5]. Natural Language Sentence Matching (NLSM) [22] is applied to discern if sentences with similar intents are expressed differently. Traditional methods emphasize feature engineering, leveraging bag of words (BOW), TF-IDF [5], and n-grams. Support Vector Machines (SVM) are commonly used in text categorization, employing various feature extraction techniques. Recent advancements in deep learning have excelled in NLP tasks, particularly semantic text similarity, utilizing pre-trained word embeddings to capture semantic symmetries effectively. Detecting equivalent sentences or questions has been a longstanding challenge in NLP. Traditional machine learning algorithms, like SVMs, excel with handcrafted features and preprocessed data. Deep learning methods often employ Siamese neural networks, such as Siamese MaLSTM [21], encoding input sentences using the same neural network. Techniques combining lexical, syntactic, and semantic features, along with deep learning models like Recursive Neural Networks and Recurrent Neural Networks,

have shown promise in sentence similarity tasks, demonstrated in SemEval [12] competitions. Successful systems include ASOBEK [11] using SVM with lexical and character n-gram features and MITRE [14] employing a recurrent neural network with string matching features.

III. Dataset Details

For this project we make use of the publicly available Quora Question Pairs dataset available on Kaggle [8][17]. For our experiments, we make use of the training set provided by the website. The training set consists of 4,04,290 labeled question pairs. The fields in the dataset are shown in TABLE 2 Of all the question pairs, 149302 are duplicates, or roughly 37% of the full dataset. We assume that questions marked as duplicates in the Quora dataset are semantically equivalent since Quora's duplicate question policy concurs with our definition of semantic equivalence above. The dataset has been labeled manually by humans. Hence there is bound to be some noise in the labels. We have split our data into three sets of training, validation, and testing sets. The training set has 3,04,290 entries whereas the validation and test sets contain 50,000 entries each.

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:

TABLE 1: Field Description of our dataset

Fields	**Description**
id	unique identifier for the question pair
qid1	qid1 unique identifier for the first question
qid2	qid2 unique identifier for the second question
question1	question1 full unicode text of the first question
question2	question2 full unicode text of the second question
is duplicate	is duplicate 1 if questions are duplicates, 0 otherwise

IV. Dataset And Preprocessing

Since the classifier is only concerned with "question1", "question2" and "Is_duplicate", the rest of the attributes of the dataset are ignored.

The steps required for organizing the data in understandable format by handling the missing, inconsistent and redundant values is called preprocessing. Various pre-processing steps are performed on experimental dataset. Several NLP techniques are used such as conversion to lower letters of text, stopwords removal, stemming, and tokenization, with the help of freely available libraries such as NLTK [5] and keras's.

TABLE 2: Sample Date in our Dataset

id	**qid**	**qid2**	**question1**	**question** 2	**is duplicate**
234	256	257	How can I be a good geologist?	What should I do to be a great geologist?	1
345	356	357	How do I read and find my YouTube comments?	How can I see all my Youtube comments?	1
678	456	467	Which is the best digital marketing institution in banglore?	Which is the best digital marketing institute in Pune?	0
876	789	800	Do smart people brag?	Why do smart people think dumb?	0
890	921	922	What is f and o in a stock market?	What is F&O in stock market?	1

V. Our Approaches and Work Flow

A. Basic Data Analysis

Basic Data Analysis, often referred to as Exploratory Data Analysis (EDA), is a fundamental process in data science that involves thoroughly examining and understanding a dataset before proceeding to more complex analyses or modeling tasks. The primary goal of EDA [17] is to extract insights and uncover hidden patterns, relationships, and anomalies within the data.

The workflow of basic data analysis begins with data collection and understanding, where the dataset is obtained and its structure is comprehended, including the types and meanings of variables. Following this, data cleaning and preprocessing are performed to handle missing values, remove duplicates, and standardize formats, ensuring that the dataset is ready for analysis.

Descriptive statistics are then calculated to summarize the dataset, providing initial insights into the distribution of numeric variables and the frequency of categorical variables. Visualization [17] plays a crucial role in EDA, allowing for the creation of graphical representations such as histograms, box plots, scatter plots, and heatmaps to explore relationships and identify trends, outliers, and potential patterns.

Feature engineering is conducted to derive new features or transform existing ones based on insights gained during analysis. Correlation analysis helps understand relationships between variables, identifying highly correlated features that may impact modeling. Additionally, exploratory techniques like clustering [17] or anomaly detection [19] can reveal meaningful patterns or anomalies within the data.

B. Feature Engeering

Feature engineering is a crucial technique in data science aimed at improving model accuracy by creating new, meaningful features from existing data. This process involves transforming, scaling, extracting, and encoding features to enhance predictive power.

In our next step, we will augment our dataset by adding seven new features using the bag of words model applied to pairs of questions. The bag of words [5] model converts textual data (questions in this case) into numerical feature vectors based on word frequencies. For each question pair, this model will generate unique features representing the similarity or overlap of words between the questions. These new features will undergo exploratory data analysis (EDA) to understand their impact and relationship with other variables. Subsequently, they will be incorporated into our machine learning model to improve its predictive capability, leveraging the insights gained from EDA.

This iterative process of feature engineering and analysis plays a pivotal role in refining our dataset and optimizing model performance for effective prediction and decision-making.

i. Question Length: The size of the question is a critical feature because when we vectorize it, the question gets split by words, so having the length feature is good. The length we are having is the character-wise length. So, it will create 2 new features for the length of questions 1 and 2.

ii. Number of words: The number of words in both questions is another feature that should impact the model performance. So, it will add 2 new features for questions 1 and 2. To add the feature, split the sentence with space and extract the length of the list.

iii. Common words: Another feature is to know how many common words there are in both questions. It helps identify the similarity between both questions. Calculating where you only need to apply the intersection between both questions is simple. For this, we find the number of unique words in both questions and apply the set intersection to the set length.

iv. Total words: The sum of the total number of unique words in each question. In simple terms, find the number of unique words in both questions and return their sum.

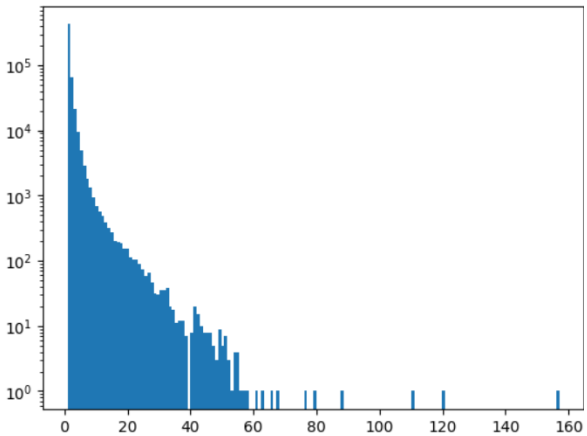
v. Word share: It is one exciting feature and simple to add. To calculate, divide the common words by the total number of words

C. Exploratory Analysis of Newly Added Features

We have introduced some new features in the dataset, and it is an excellent time to analyze the relationship and their spread with the target variable.

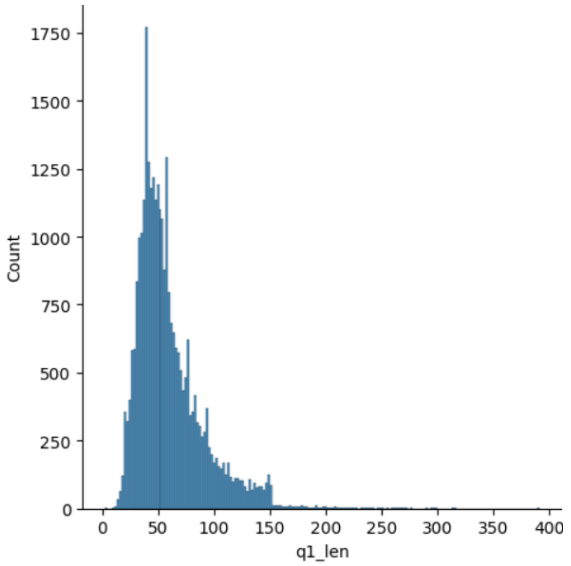
i. Distribution of Question

We will analyze the length of the question, like the average length of each question in 1 and 2. Minimum and maximum number of characters. So, plot the distribution plot for questions 1 and 2.



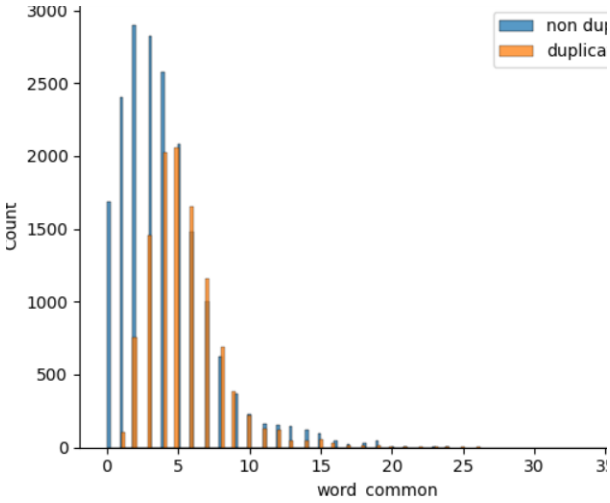
ii. Distribution of Number of words

Generate the same graphical analysis for the average, minimum, or maximum number of words in each question we have done.



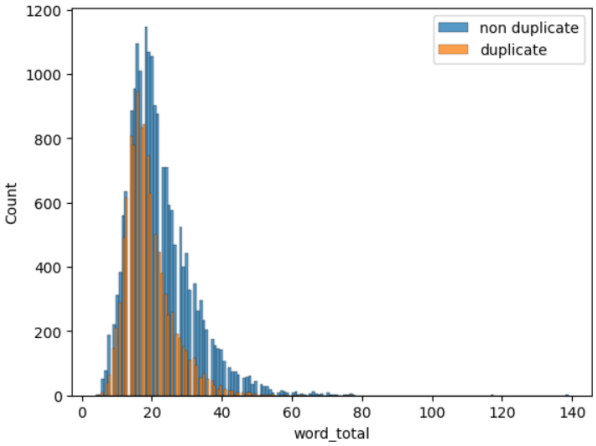
iii. Common word Analysis

We will plot the distribution of common words in questions 1 and 2. In this distribution plot, we will have a separate curve of non-duplicate and vice-versa



iv. Total word Analysis

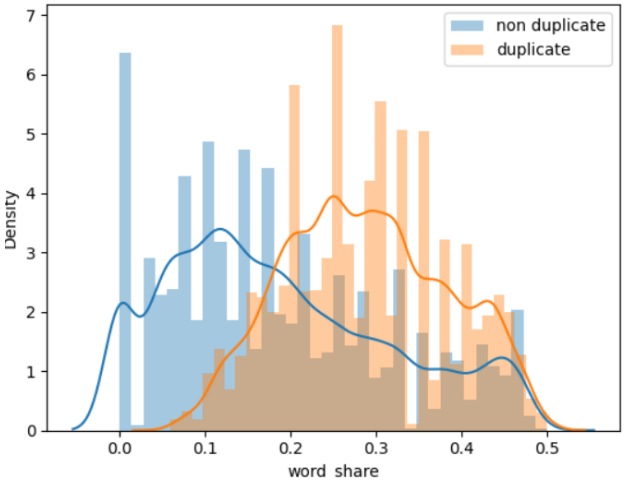
The same analysis for total words against target variable unique entries.



Records will be duplicated if the total word count is between 0 and 20, but if it is more significant than 40, the model gives high weightage to non-duplicates.

v. Word share Analysis

Plot the distribution plot for duplicate and non-duplicate against the words share column.



A non-duplicate is likely to occur if the words share value is less than 0.2, but duplication occurs if the word's share value is more significant than 0.2.

VI. Machine Learning Model Creation

After performing the above EDA, we gain the confidence to keep the features in our dataset and move to the modelling part.

1. Separate the Independent and Dependent features

First, we need to drop the unnecessary columns and pick the columns needed for training and one target feature (dependent). So, we will pick questions in different data frame for vectorizing and other features in another data frame with the target variable. And concat them after vectorizing.

2. Vectorizing the Feature

We need to turn the questions into numerical ones because we can't provide the string to the model. To do this, we employ a variety of feature vectorizing techniques; for the moment, we'll use a bag of words (BOW). The bow is a technique for extracting characteristics from text input for machine learning algorithms. It displays the text that describes words' behavior in the corpus, which entails two things. The first is words' vocabulary (unique words in the corpus added as a new feature), and the second is a way to count the number of known words (represent the word's presence in that query using binary format).

3. Train – Test Split

For calculating the performance of the model, we need some amount of data that the model has not seen as a test set, so we will split the final data frame into two parts, training and test set, where 80 percent of data in the train set and 20 percent for the test set.

4. Train the Machine Learning Models

To determine which model works the best, we will train XGboost, and Random Forest both models.

VII. Initial Model Performance

TABLE 3: Initial Performance

Model Type	Mode Detail	Accuracy (%)	Recall (%)
Classifier	Random Forest	74	73
	XGBoost	73	72

VIII. Optimizing the Current Model Performance

The main work now is increasing the model performance and getting a well-generalized model. therefore far, we have not performed any text preprocessing, so from now we will perform text mining and analyze the text in different ways to generate some advanced features, which we can say is Advance feature engineering. The feature I have found through multiple notebooks available on Kaggle is that the features are represented as Magic. So, for this, we will again load the data in a new data frame because we will clean the text and calculate all features again.

IX. Text Preprocessing

The first step is to clean up the text and rectify the dataset by removing the irregularities in regular NLP Projects. As a result, we will do the following text-cleaning processes.

i. **Lowercase:** If the text falls into one case, it is simple to vectorize and interpret because the vectorizer considers token and Token to be different words. So, we will convert the entire text into lowercase.

ii. **String Equivalents:** The text contains multiple symbols, so we will replace them with corresponding string words.

iii. **Expand Contraction:** Contraction is written communication in human language to write words in short form. For example, don't stand for do not so there are multiple contractions which we need to change to corresponding complete forms.

iv. **Remove HTML tags:** The text contains some unnecessary HTML tags, so that we will remove them.

v. **Remove Punctuation:** Punctuation is unnecessary and does not convey meaning, so it is better to remove them.

X. Advance Feature Engineering

In the domain of identifying duplicate question pairs on Quora, advanced feature engineering plays a pivotal role in augmenting the performance of machine learning models. By employing sophisticated techniques tailored to the characteristics of textual data, we can extract nuanced features that effectively capture the subtle nuances and distinctions between questions. Advanced feature engineering for duplicate question pair identification involves leveraging techniques such as text embeddings, where words are represented as dense vector representations using pre-trained models like Word2Vec [3] or GloVe [2] These embeddings encode semantic relationships between words and can be aggregated to represent entire questions in a meaningful vector space.

A. Tokken Based Feature

Tokens are chunks of words that unite to form a sentence. A single word is a Token so we will add some of the features at the token level or that will help to calculate some valuable columns.

i. **Cwc min** – It represents the number of common words with a minimum number of words in more minor questions.
ii. **Cwc max** - represents the number of common words to a maximum number of words in a larger question.
iii. **Csc min** – It represents the number of common stop words to the smaller stop word count among two questions.
iv. **Csc max** - represents the number of common stop words to the larger count among two questions.
v. **Ctc min** – The number of standard tokens to count more minor tickets among two questions.
vi. **Ctc max** – The number of standard tokens to count more significant tokens among two questions.
vii. **Last word equal** – The binary feature includes the value 1 if the last word of both questions is the same, else 0.
viii. **The First equal** - If the first word of both questions is equal then the value is 1. Else 0.

B. Length Based Feature

i. **Mean Length** – It is an essential but crucial feature where we will find the average of both questions' length, known as the mean length of both questions. For example, if the first question is 8 characters in length and the second is 6 characters in length, then the average becomes 14 divided by 2 equals 7, so the mean is 7.
ii. **Absolute Length Difference** - The absolute difference between the length of the two questions (length of words)
iii. **Longest Substring Ratio** - The ratio of the length of the longest substring between two questions is divided by the length of the smaller questions. The first thing is you need to find the substring in both questions and determine the longest one and then divide it with a length of a small token sentence.

C. Fuzzy Feature

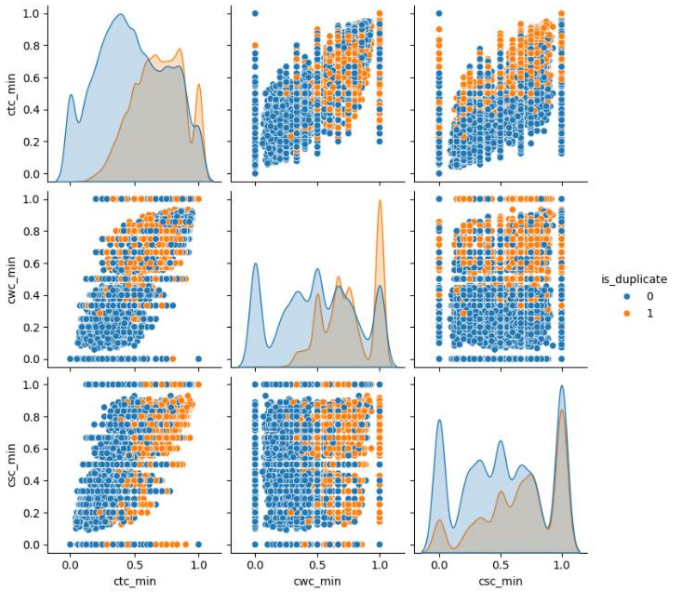
These are some features that are generated using the fuzzy-wuzzy library. To get important about this feature, you can recommend this blog written by the founders of the library itself, where in detail, they explain how these features are calculated.

XI. EDA of Newly Created Features

We have added nearly 23 new features to our dataset, so before modeling, we want to be confident that these features would dominate the prediction of output variables, so we run some analysis to identify specific patterns.

I. Minimum Variables with Target Variable

The feature we have added as a minimum calculation against the target feature to see how they affect record duplication.

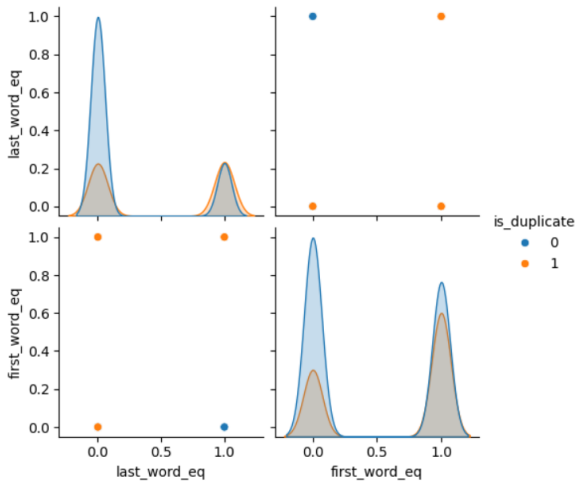


II. Maximum Variable with Target Variable

We have also appended the maximum calculation, so let us plot them against target variables.

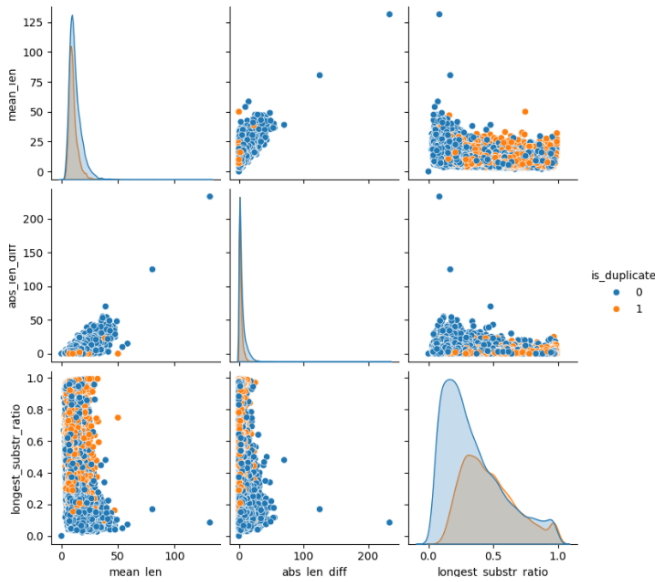
III. Last Word and First Word Analysis

We plot the first and last word match against the target variable.



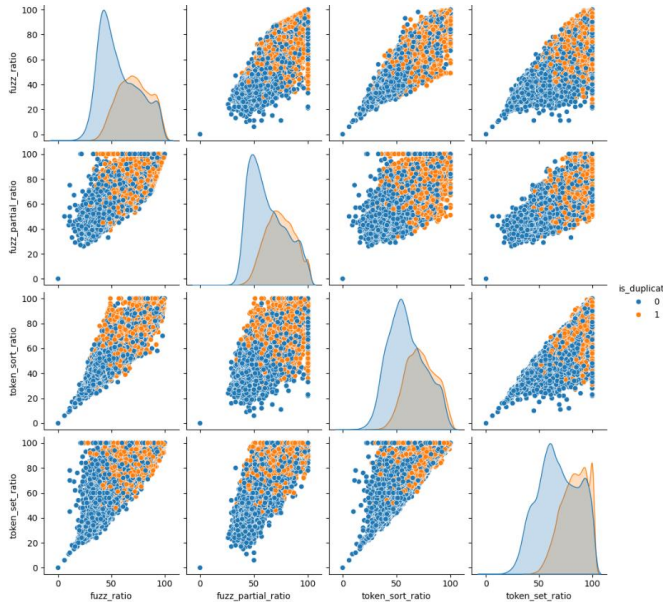
IV. Length-based Feature Analysis

The mean length and the absolute curve do not give much information because both the curves are moving almost together, but the most extended substring feature is beneficial where a blue curve is dominating.



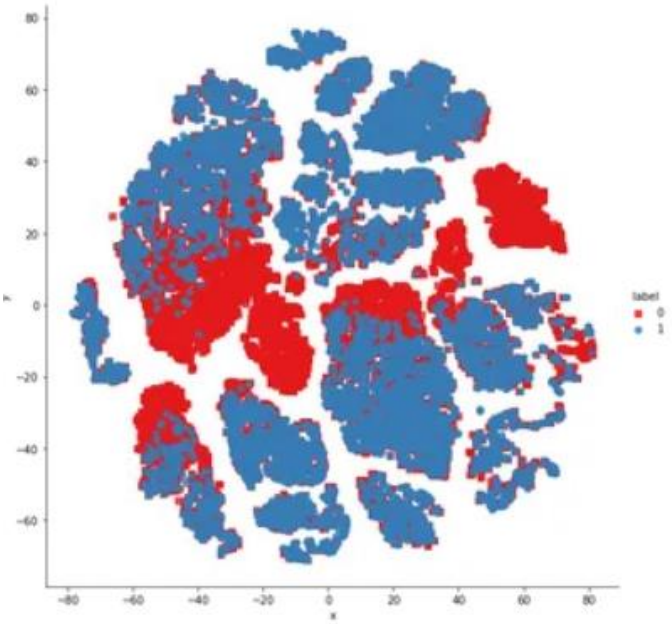
V. Fuzzy Feature Analysis

All 4 features give a good understanding of the output variables, which can be useful.



XII. Dimensionality Reduction

We will employ TSNE (T-distributed stochastic neighbor Embedding) [23], a non-linear unsupervised dimensionality reduction technique for data exploration and visualization. First, we'll plot the data on the 2D graph. Then, we'll use it to visualize the data in 3-D, allowing you to see the impact of 15 features on the target variable. Visit the notebook to get the code for the plot 3-D view.



XIII. RESULTS

The data is now ready, and you must repeat the steps above to train the Random Forest and XGboost models for NLP Project. You can rerun the cell or copy and paste the code. Only the difference here is we have 15 more features which total become 6023 by adding 15 more features. The random forest accuracy is approximately 78.7, and XGboost gives 80 percent. So, by doing this much optimization, we could boost the performance by 2 to 2.5 percent. We believe that the performance will only increase slowly after the model gets the main explanation.

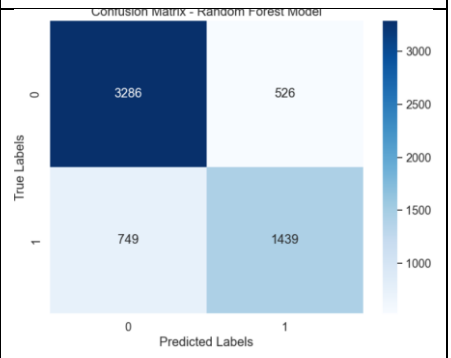
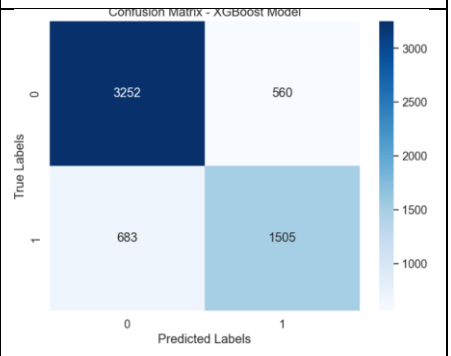
A. Selecting the best Model

This is where many engineers and practitioners make the mistake of picking the best model for deployment. So, we have two underlined scenarios to consider while selecting between the models.

B. Observation

i. When the real value is non-duplicate, but the model reports it as such.
ii. When the actual value is duplicated, the model predicts it is not.

Table 4 RESULTS ANALYSIS

Model Type	Model Detail	Initial (%)	F1 Score After Adding 7 new Features (%)	F1 Score After Adding 15 new Feature (%)	Confusion Matrix
1.	Random Forest	74	77	78	
2.	XGboost	73	76	80	

XIV. Confusion Matrix

.A confusion matrix is a table used to evaluate the performance of a classification model by summarizing the counts of correct and incorrect predictions made by the model on a dataset. It presents a clear and concise summary of the model's performance in terms of predicted class labels versus actual class labels. The confusion matrix is typically a square matrix with dimensions equal to the number of classes in the dataset. For a binary classification problem (two classes: positive and negative), the confusion matrix is a 2x2 matrix. For multiclass classification problems (more than two classes), the matrix size corresponds to the number of classes.Figures and Tables.

	Predicted Negative	*Predicted Positive*
Actual Negative	*TN*	*FP*
Actual Positive	*FN*	*TP*

References

[1] Achananuparp, Palakorn, et al. "Utilizing sentence similarity and question type similarity to respond to similar questions in knowledge-sharing community." *Proceedings of QAWeb 2008 Workshop*. Vol. 214. 2008.

[2] Mueller, Jonas, and Aditya Thyagarajan. "Siamese Recurrent Architectures for Learning Sentence Similarity." *AAAI*. 2016.

[3] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781* (2013).

[4] Dey, Kuntal, Ritvik Shrivastava, and Saroj Kaushik. "A paraphrase and semantic similarity detection system for user generated short-text content on microblogs." *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. 2016.

[5] Bird, Steven, and Edward Loper. "NLTK: the natural language toolkit." *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*. Association for Computational Linguistics, 2004.

[6] Achananuparp, Palakorn, Xiaohua Hu, and Xiajiong Shen. "The evaluation of sentence similarity measures." *International Conference on data warehousing and knowledge discovery*. Springer, Berlin, Heidelberg, 2008.

[7] Sanborn, Adrian, and Jacek Skryzalin. "Deep learning for semantic similarity." *CS224d: Deep Learning for Natural Language Processing*. Stanford, CA, USA: Stanford University (2015).

[8] Quora Question Pairs | Kaggle [Online] Available: https://www.kaggle.com/c/quora-question-pairs

[9] Langville, Amy N., and Carl D. Meyer. *Google's PageRank and beyond: The science of search engine rankings*. Princeton University Press, 2011.

[10] Brychcín, Tomáš, and Lukáš Svoboda. "UWB at SemEval-2016 Task 1: Semantic textual similarity using lexical, syntactic, and semantic information." *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. 2016.

[11] Eyecioglu, A. and Keller, B. (2015). "ASOBEK: Twitter paraphrase identification with simple overlap features and SVMs." In *Proceedings of SemEval*.

[12] Zarrella, G., Henderson, J., Merkhofer, E. M., and Strickhart, L. (2015). "MITRE: Seven systems for semantic similarity in tweets." In *Proceedings of SemEval*.

[13] Peters, Matthew E., et al. "Deep contextualized word representations." *arXiv preprint arXiv:1802.05365* (2018).

[14] Zarrella, G., Henderson, J., Merkhofer, E. M., and Strickhart, L. (2015). "MITRE: Seven systems for semantic similarity in tweets." In Proceedings of SemEval.

[15] Home - Quora [Online] Available: https://www.quora.com/

[16] reddit: the front page of the internet [Online] Available: https://www.reddit.com/

[17] Kaggle: Your Home for Data Science [Online] Available: https://www.kaggle.com/

[18] Ke, Guolin, et al. "LightGBM: A highly efficient gradient boosting decision tree." *Advances in Neural Information Processing Systems*. 2017.

[19] Ho, Tin Kam. "Random decision forests." *Document analysis and recognition, 1995., proceedings of the third international conference on*. Vol. 1. IEEE, 1995.

[20] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 2016.

[21] Wieting, John, et al. "Learning Paraphrases with Siamese Recurrent Networks." Proceedings

[22] Wang, Zhen, and Jianwen Zhang. "Sentence Similarity Learning by Lexical Decomposition and Composition." Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). 2015.

[23] Maaten, Laurens van der, and Geoffrey Hinton. "Visualizing Data using t-SNE." Journal of Machine Learning Research, vol. 9, pp. 2579-2605, 2008.

[24] Stackoverflow: Every developers Home [Online] Available: https://stackoverflow.com/