Nigerian Journal of Technology (NIJOTECH)
SI-2026A: Advances in Modelling, Simulation, and AI/ML for Multi-Disciplinary Engineering Applications, *2026. pp. 76- 86*
www.nijotech.com

Print ISSN: 0331-8443
Electronic ISSN: 2467-8821
https://dx.doi.org/10.4314/njt.v45i1.6s

ANALYZING ENSEMBLE LEARNING TECHNIQUES FOR DETECTING REDUNDANT QUESTIONS ON QUORA

AUTHORS:
*B. Yelure[1], S. Pawar[2], S. Thorat[2], S. Patil[3], A. Patokar[2] and P. Jawade[4]

AFFILIATIONS:
[1]Department of Computer Science & Engg., Government College of Engineering Kolhapur, INDIA
[2]Department of Information Technology, Government College of Engineering Karad, INDIA
[3]Department of Computer Engineering, DYPCOE, Kolhapur, INDIA
[4]Department of Computer Engineering, Government College of Engineering Nagpur, INDIA

***CORRESPONDING AUTHOR:**
Email: bhushanyelure2008@gmail.com

ARTICLE HISTORY:
Received: December 07, 2025.
Revised: March 25, 2026.
Accepted: April 01, 2026.
Published: May 4, 2026.

KEYWORDS:
Redundant Questions, Machine Learning, Natural Language Processing, Feature Extraction.

ARTICLE INCLUDES:
Peer review

DATA AVAILABILITY:
On request from author(s)

EDITORS:
Sagar Shelare
Sameer Sheshrao Gajghate

FUNDING:
None

Abstract
Redundant questions are a major concern faced by community question-answering platforms such as Quora, Stack Overflow. Redundant questions reduce the efficiency of information retrieval, hinder effective statistical categorization, and limit the availability of diverse responses for users. Therefore, this research focuses on identifying duplicate or redundant questions by applying techniques from Machine Learning and Natural Language Processing. The dataset of more than 400,000 question pairings retrieved from Quora is preprocessed using tokenization and stop word removal. Feature extraction is performed on this preprocessed dataset. The proposed approach utilizes Bag-of-Words (BoW) for feature extraction, transforming raw question pairs into structured numerical vectors to optimize the performance of the ensemble classifiers. The algorithms, including Decision Tree, Random Forest, XGboost, and Adaboost are applied on the dataset for detecting duplicate questions. Random Forest outperformed Decision Tree, XGboost, and Adaboost classifiers with an accuracy of 81.69 %.

1.0 INTRODUCTION

1.1 Background and Motivation

Community question and answer platforms are online discussion boards where users can post questions or inquiries and receive professional responses. Some of the most widely used websites for answering questions include Quora [1], [6], [8], Stack Overflow [10], and Stack Exchange [1], [9]. All of the questions and answers that are being discussed on the forum are kept in a database that is maintained by these question & answer websites. The queries are becoming more frequent as the number of users on

HOW TO CITE:
Yelure, B., Pawar, S., Thorat, S., Patil, S., Patokar, A., and Jawade, P., "Analyzing Ensemble Learning Techniques for Detecting Redundant Questions on Quora", *Nigerian Journal of Technology*, SI-2026A: Advances in Modelling, Simulation, and AI/ML for Multi-Disciplinary Engineering Applications, 2026. pp. 76-86.
https://dx.doi.org/10.4314/njt.v45i1.6s

these platforms are rising. The issue of duplicate questions is growing due to digitization and popularity of these informative sites among the users. A 2025 case study on the Quora dataset highlights that identifying this semantic similarity remains a primary technical hurdle in managing community forum content [26]. Users may find it challenging to locate the best response to their inquiries due to this redundancy. Also, the redundancy of questions makes it difficult for experts to answer, since they have to give same answer at multiple places. Furthermore, question redundancy creates significant inefficiency for subject matter experts, as they are compelled to provide identical responses across multiple threads, thereby diluting their productivity and fragmenting the knowledge base. Addressing this issue is critical for maximizing the volume and quality of responses available to users on Q&A websites [24], [25]. Duplicate questions are defined as inquiries that have been previously submitted and for which answers already exist on the forum; the recurrence of such entries results in significant data redundancy within the database. As argued by Kumar et al. [23] eliminating duplicate copies of data reduces storage requirements while increasing bandwidth efficiency.

Furthermore, addressing redundancy lowers storage expenses since fewer drives are utilized. To improve the user experience and reduce redundancy in the database, it's important to detect questions with the same semantic meaning. Users are encouraged to mark questions as duplicates as soon as they notice them. However, since this is a manual process, it is often inefficient and inconsistent [11]-[15]. The objective is to add extra features to the existing data set to assist ML algorithms in recognizing the underlying pattern of duplicate questions. The ensemble learning technique is selected for this problem since it combines the result of weak learners and forms a strong learner and improves accuracy. The performance analysis of ensemble learning techniques such as DTC, RFC, XGB classifier, and ADB classifier are compared in terms of their accuracy. The data set used for experimentation is an open-source data set that is available on Kaggle and it was posted by Quora. Table 1 describes notations and abbreviations used throughout the paper.

Several academics have used ML and DL to detect duplicate questions. The review covers data set, methodology, performance metric, and research objective.

Due to semantic loss, earlier redundant question detection methods that automatically identify repetitive queries by extracting text features are limited. Word2vec CNN, RNN, and LSTM eliminate duplicate questions [1]. The experiment dataset is from Stack Overflow. The recall rate is used to evaluate the model across six question sets. For duplicate detection across domain-specific datasets, semantic matching models (SMM) capture sentence-level similarity [2]. Researchers in [3] use machine learning models and TF-IDF word weights to detect duplicate questions. Four algorithms—LR, DT, DT with Bagging, Adaboost, and RF—are tested on Quora. The experiment shows that Adaboost has the highest accuracy rate (81.73%), followed by RF (81.72%), DT (79.29%), and LR (79.21%). DT, Naive Bayes, KNN, and LR are used in [23] to solve the Quora question duplication problem. These algorithms are tested for duplicate question detection. Experimental results show that DT is the best algorithm for detecting duplicate questions over Naive Bayes, KNN, and LR[13].

Table 1: Notation table

Notation	Significance
NLP	Natural Language Processing
RFC	Random Forest Classifier
DTC	Decision Tree Classifier
ML	Machine Learning
DL	Deep Learning
TFIDF	Term Frequency-Inverse Document Frequency
BoW	Bag Of Words
EDA	Exploratory Data Analysis
XGB	XGBoost
ADB	Adaboost
LR	Logistic Regression
KNN	K Nearest Neighbour
TP	True Positives
TN	True Negatives
FP	False Positives
FN	False Negatives

Using Google News Vector and FastText Crawl, the Siamese MaLSTM was assessed for its ability to identify duplicate question pairs [4]. This model has an 80% accuracy rate when working with the Google News Vector and a 90% accuracy rate when working with the FastText Crawl, which means it is more effective than any previously published models. Moreover, this particular model demonstrates the effectiveness of the word embedding technique and

https://dx.doi.org/10.4314/njt.v45i1.6s

how it can be applied to duplicate question pair detection.

Quora duplicate questions have been identified using machine learning (ML) and deep learning (DL) techniques [26]. In this case, ML and DL are employed to identify duplicate questions; however, while BOW, LR, XGBoost, RNN and Word2vec models were used in previous studies, the present work found that, with 80.99% accuracy, XGBoost outperforms RNN (79.63%). The authors also conclude that the use of Word2vec leads to improved performance over BOW models.

SVM deep neural models have been employed to detect redundant questions on Quora [7]. The SVM achieved an accuracy rate of 82.4% with a corresponding F1 score of 80.21%. The study utilized grid search to find optimal hyperparameter values; thus showing both SVM's effectiveness for natural language processing (NLP), and the relevance of hyperparameter selection for optimal model performance.

2.0 METHODOLOGY

Methodology for the detection of redundant questions comprises various stages such as question pairing, preprocessing on selected pairs, cleaning of the data, data splitting, model training and analysis of results. The mentioned steps are depicted in Figure 1.

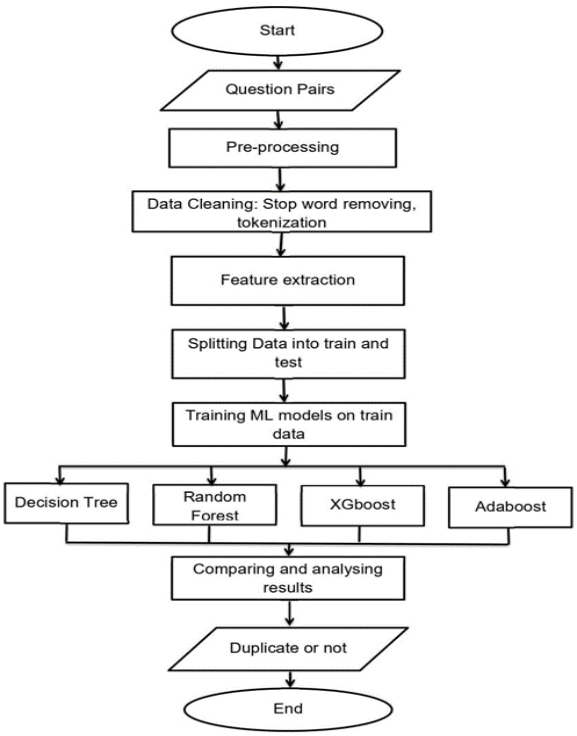

Figure 1: Process for Detecting Redundant Questions

The workflow in the figure 1 for redundant questions detection starts with gathering pairs of questions and getting them ready for analysis through proper pre-processing. In this stage, the text is cleaned by removing stop words, applying tokenization, and performing other essential normalization steps. Once the data is cleaned, important features are extracted so that the model can understand how similar the two questions are. The dataset is then split into training and testing parts. Several machine learning algorithms such as Decision Tree, Random Forest, XGBoost, and AdaBoost are trained using the training data [16], [18]. The selection of aforementioned ensemble techniques is supported by recent trends in automated model design, which suggest that ensemble configurations are highly effective for capturing complex semantic similarities [27]. Their results are evaluated on the test data to identify which model performs the best in predicting similarity. Based on the top-performing model, each question pair is finally labelled as duplicate or non-duplicate, completing the entire process.

2.1 Exploratory Data Analysis

EDA [5] is a method for comprehending the data and spotting hidden patterns. There are roughly 404,290 question pairs in the dataset. The dataset exhibits a moderately imbalanced class distribution of duplicate and non-duplicate questions, as shown in Figure 2: roughly 36.9% of the samples are in the duplicate class (label = 1), and roughly 63.1% are in the non-duplicate class (label = 0). This distribution reflects a realistic situation that is frequently seen in community question-answering platforms, where there are inherently more unique queries than duplicates.

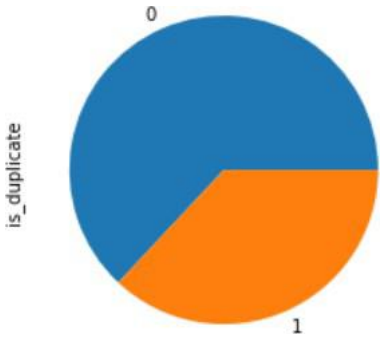

Figure 2: Distribution of redundant (1) and non-redundant (0) questions in the dataset

2.2 Data Cleaning and Preprocessing

The EDA analysis results contain repetitive data that needs to be removed from the dataset. Removing duplicate data reduces the dataset size, which, in turn, shortens the time required to train the model.

https://dx.doi.org/10.4314/njt.v45i1.6s

The preprocessing phase removes stop words from the dataset because they don't contribute to the semantic meaning of the sentence.

2.3 Feature Extraction

The data set comprises of six features, which are inadequate for determining the duplication of question pairs so enhancement in the performance of model is required. The number of features increases when basic features like length q1, length q2, words q1, words q2, a word common, and word sharing are retrieved from the data set and added. The basic features added are as follows:

•q1 length & q2 length represent the total characters in question1 & question2 respectively:

$$q1\ length = length(q1) \quad (1)$$

$$q2\ length = length(q2) \quad (2)$$

•q1 words & q2 words represent the total words in question1 & question2 respectively:

$$q1\ words = WordCount\ (q1) \quad (3)$$

$$q2\ words = WordCount\ (q2) \quad (4)$$

The Figure 3 demonstrates the significance of word similarity extracted from the dataset when assessing the redundancy of questions. When the word similarity count is greater than 5, there is a higher probability that a question will be redundant. This distinction between duplicate and non-duplicate questions is observed in the Figure 3.

Algorithm 1: Word Count
function WORDCOUNT (sentence)
words ← Tokenize(sentence) word
count ← lengthof (words) return
word count
end function

• word total represent the total words in both question1 and question2:

$$word\ total = q1\ words + q2\ words \quad (5)$$

• word common represent the number of words common to both questions:

$$word\ common = CountCommonWords\ (q1, q2) \quad (6)$$

Algorithm 2: Count Common Words
function COUNTCOMMONWORDS(sentence1, sentence2)
words1 ← Tokenize (sentence1)
words2 ← Tokenize (sentence2)
common words ← 0
for word1 in words1 do
if word1 in words2 then
common words ← common words + 1
end if
end for
return common words
end function

Word share is the ratio of similar words to the total number of words in question1 and question2. It is clear from the Fig. 4 that there is a very minimal chance of questions being redundant when the word share between them is less than 0.2 and a high probability if it is greater than 0.2 [23].

Advance Features such as cwcmin and cscmin are also considered.

The feature cwcmin represents the ratio of the intersection of non-stop-word sets between two questions over the minimum length of the two sets. Theoretically, this captures subset similarity, identifying cases where one question's intent is entirely encapsulated within another, regardless of additional descriptive modifiers in the longer query.

$$CWCmin = Tcw/Mw \quad (7)$$

The cscmin feature calculates the commonality of functional stop-words. Similar distributions of stop-words often indicate similar grammatical structures, which, when combined with semantic features, improve the model's ability to distinguish between question types (e.g., 'how-to' vs. 'what-is').

$$CSCmin = TSw/Msw \quad (8)$$

Where, Tcw: Total number of common words between question1 and question2.

Tsw: Total number of common stop words between question1 and question2.

Mw: Minimum number of words among question1 and question2.

Msw: Minimum number of stop words among question1 and question2.

https://dx.doi.org/10.4314/njt.v45i1.6s
http://creativecommons.org/licenses/by-nc-nd/4.0/

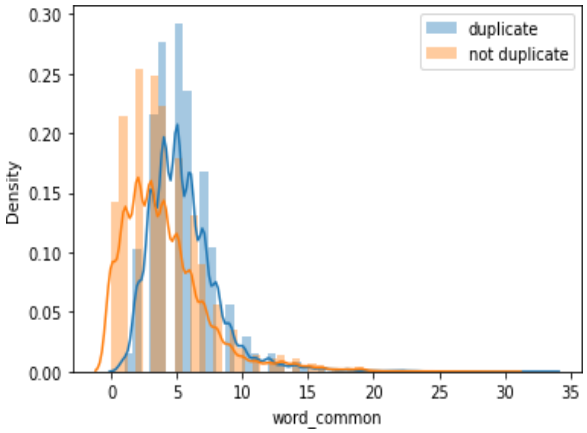

Figure 3: Plot of common words in redundant and non-redundant questions

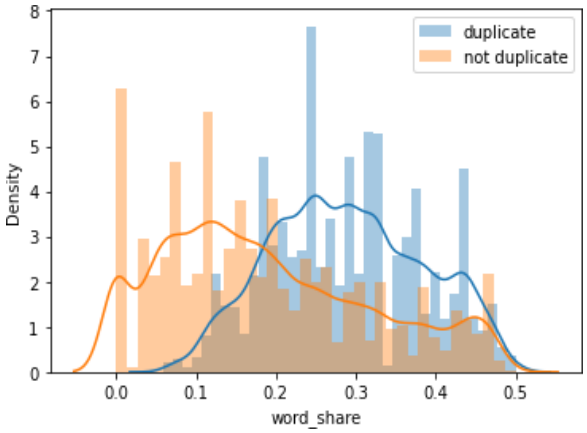

Figure 4: Plot of word share feature in redundant and non-redundant questions

From Figure 5, it is observed that the chance of a question being duplicated increases when cwcmin is more than 0.5. Similarly from the graph of cscmin feature indicates that cscmin is also a major feature in deciding duplication of question.

Let:

CWCmax = Tcw/ MWmax (9)

CSCmax = Tsw/ MSWmax (10)

Algorithm 3: Check if First Words are Equal
function IsFirstWordEqual(*sentence*1, *sentence*2)
*words*1 ← *Tokenize*(*sentence*1)
*words*2 ← *Tokenize*(*sentence*2)
*f irst word*1 ← *words*1[0]
*f irst word*2 ← *words*2[0]
return *f irst word*1 == *first word2*
end function

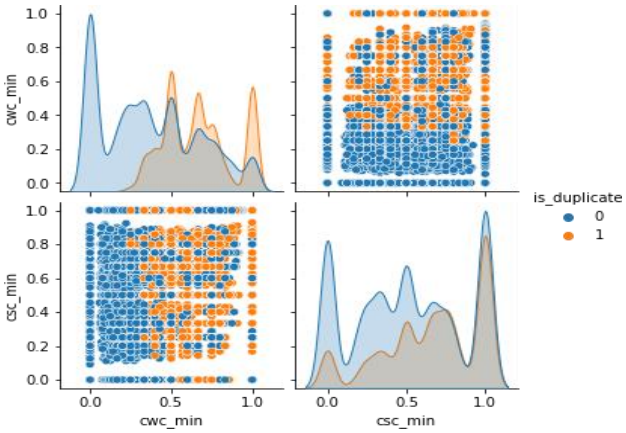

Figure 5: Pair plot of cwcmin and cscmin in redundant and non-redundant questions

Algorithm 4: Check if Last Words are Equal
Function ISLASTWORDEQUAL(*sentence*1, *sentence*2)
*words*1 ← *Tokenize* (*sentence*1)
*words*2 ← *Tokenize* (*sentence*2)
*last word*1 ← *words*1
[*LastIndex*(*words*1)] *last word*2 ← *words*2
[*LastIndex*(*words*2)] return *last word*1 == *last word*2
end function

The Figure 6 shows when the last word and the first word of questions are not equal there is more probability of no redundancy among them therefore last word eq and first word eq are important features in deciding the redundancy of questions.

Longest Substring Ratio = length (LongestCommonSubstring)/min (length (question1), Length (Question2)) (11)

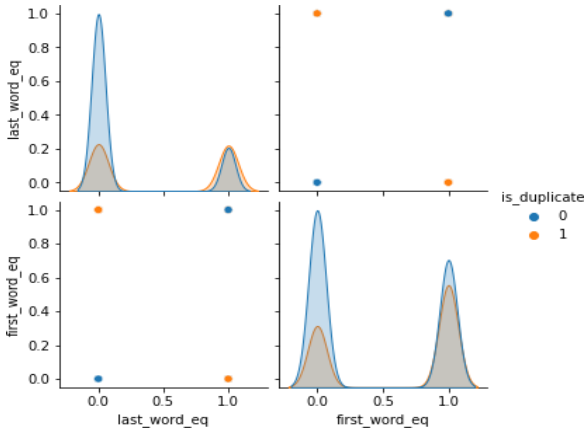

Figure 6: Pair plot of first word equation and last word equation feature in redundant and non-redundant questions

https://dx.doi.org/10.4314/njt.v45i1.6s

Longest substr ratio is the ratio of the longest common substring for redundant and non- redundant questions, as shown in the Figure 7.

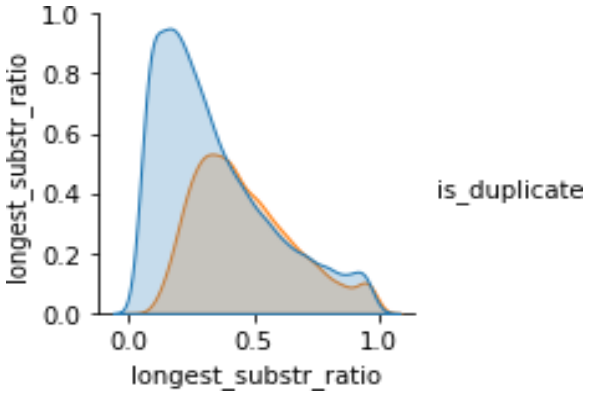

Figure 7: Pair plot of longest common substring in redundant and non-redundant questions

The bag of word technique is used for the vectorization [26] of text data. ML algorithms cannot work directly with raw text data, so it's needed to transform the text data into numbers, especially vectors of numbers before a ML model is trained on text input. 3000 bags of word features for both the questions have been computed and added to the dataset. The train-test split approach is used to calculate how well machine learning algorithms perform when predicting using data that was not utilized for model training. Splitting the data is necessary to ensure that every data set has a high amount of training data. Prior to the training ML models on available data, the dataset is split as 70 percent for training and 30 percent for testing [23] - [26].

2.4 Machine Learning Models

There are various ML algorithms are available. Here for the comparative analysis DT, Adaboost, RF, and XGBoost are used [19]-[20].

Decision Tree (DT): The DT is the most powerful tool for classification problems. The DT is a tree-based classification algorithm in which intermediate nodes represent the decision node and leaf nodes represent the predicted output. Decisions are taken by splitting the nodes based on attribute selection measures [23].

Random Forest (RF): An ensemble ML technique called RF combines a number of decision trees. Building blocks of RF is the decision trees. A number of DT operate as an ensemble independent of each other and take decisions by majority voting [23], [30].

XGBoost (XGB): Extreme gradient boosting is known as XGB. The algorithm is based on ensemble DT and employs a gradient-boosting framework. It ensembles weak learners to build strong learners by minimizing errors made by weak learners [22], [29].

Adaboost (ADB): is one of the boosting technique that ensembles multiple weak classifiers into strong classifiers. A poorly performing classifier is termed a weak classifier. Adaboost combine such weak classifier to form a strong classifier thus improving performance [23].

2.5 Hyperparameter Configuration and Cross-Validation Strategy

For the purpose of comparison and to attain reproducibility in the study, hyperparameters were optimally configured for all machine learning classifiers. The hyperparameters were configured to have default settings to begin with and then were empirically configured to attain optimal performance with high computational efficiency.

The hyperparameters used in this study are summarized below:

- **Decision Tree (DT):**
  - Criterion = Gini
  - Max Depth = 12
  - Min Samples Split = 5
  - Min Samples Leaf = 2
- **Random Forest (RF):**
  - Number of Estimators (n_estimators) = 100
  - Criterion = Gini
  - Max Depth = 20
  - Max Features = sqrt
  - Bootstrap = True
  - Min Samples Split = 5
  - Min Samples Leaf = 2
- **XGBoost (XGB):**
  - n_estimators = 100
  - Learning Rate = 0.1
  - Max Depth = 6
  - Subsample = 0.8
  - Colsample_bytree = 0.8
- **AdaBoost (ADB):**
  - n_estimators = 100
  - Learning Rate = 1.0
  - Base Estimator = Decision Tree (Max Depth = 1)

https://dx.doi.org/10.4314/njt.v45i1.6s

As mentioned in Section 2.3, the training and test sets were split at a 70-30 ratio. The cross-validation technique of 5-fold cross-validation was employed only on the training set of 70% data for hyperparameter tuning. The model's performance reported in the Section 3.0 was on the unseen data of the test set. In this technique, for each fold, four sets were used for training, and one set was used for validation. This was repeated five times so that all sets could be used for validation at least once. The performance of the model was obtained by averaging all the sets.

3.0 RESULTS AND DISCUSSION

The dataset used for this work was obtained from Kaggle and provided by Quora[21]. There are total 6 features in the dataset. The distinct id for each row is contained in the first column. qid1 and qid2, which correspond to questions 1 and 2, are found in the second and third columns, respectively. The question 1 and question 2 are in the fourth and fifth columns, respectively, and the category values 1 and 0 for redundant and non-redundant questions are in the last column. Sample dataset is depicted in Table 2. The confusion matrix is created for the performance evaluation of the methods used after classification. It is a two-dimensional matrix in which one axis represents the actual class labels and the other represents the predicted class labels. Figure 8 to 11 represents the confusion matrix for Decision Tree, Random Forest, XGboost, Adaboost respectively. To ensure that the ensemble classifiers are not biased towards the majority class, evaluation metrics such as precision, recall and F1-score were evaluated in addition to accuracy during the evaluation phase. By doing so, we are able to handle the moderate imbalance between the two classes in the Quora dataset. The formulas for calculating these parameters are expressed as follows. Table 3 represents the comparative analysis of the system.

Table 3 clearly indicates that the Random Forest model performs the strongest, with all its evaluation metrics hovering around 82%. XGBoost also shows solid and dependable performance above 80%. In comparison, the Decision Tree gives moderate results near 75%, while AdaBoost records the lowest scores among the four. Overall, it is evident that ensemble methods—especially Random Forest—deliver more accurate and reliable outcomes for this classification task.

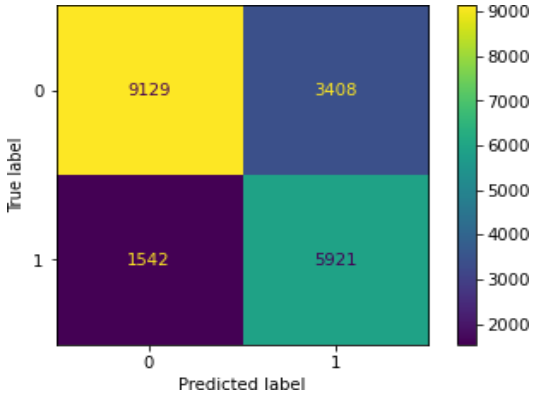

Figure 8: Decision tree

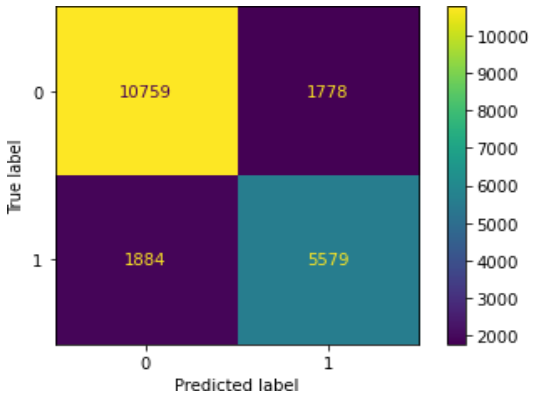

Figure 9: Random forest

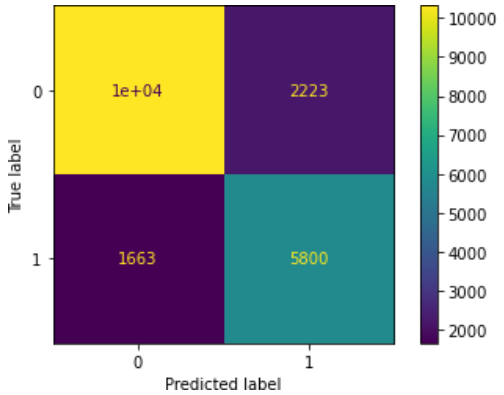

Figure 10: XGBoost

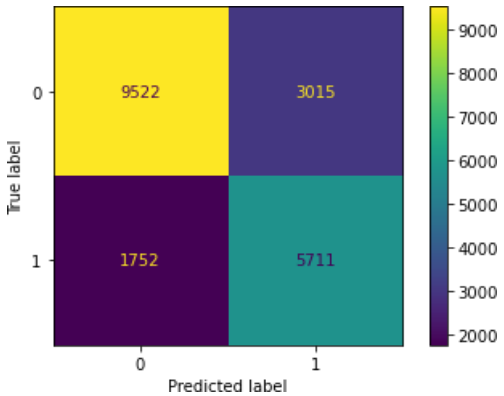

Figure 11: AdaBoost

https://dx.doi.org/10.4314/njt.v45i1.6s
http://creativecommons.org/licenses/by-nc-nd/4.0/

Table 2: Sample questions in the dataset with qid's

id	qid1	qid2	question1	question2	Is Duplicate
1	665522	665523	Why did tata sons sacked Cyrus Mistry?	Why Cyrus Mistry was removed from Tata Sons?	1
2	64553	64554	What is web application?	What is web application framework?	0
3	568878	568879	Why did Quora marked my question as incomplete?	Why does Quora detected my question incomplete?	1
4	46532	46533	What is the age of marriage in india?	what is the minimum age limit for voting in india?	0

Table 3: Result comparison

Parameter/Methodology	DT	RF	XGB	Adaboost
Accuracy (%)	75.25	81.69	80.57	74.98
Precision (%)	77	82	81	74
Recall (%)	75	82	81	76
F1 (%)	76	82	81	74

3.1 Results Interpretation and Discussion

The Random Forest (RF) model provides an accurate classification of 81.69% based on the experiments conducted; as such, it outperformed all of the other models tested. This success is attributed to the ability of RF's ensemble bagging procedure to provide superior performance within high-dimensional and sparse feature spaces that are generated via Bag-of-Words (BoW) vectorization. A collection of randomly generated and uncorrelated trees is created by the ensemble bagging procedure employed by the RF; as a result, the overall variance of the random sample of decision trees is reduced, relative to a single Decision Tree model, that generally has a tendency to overfit the data based on a certain number of occurrences of keywords that exist in the Quora dataset. One of the creative design elements for this methodology was the coupling of traditional feature engineering and ensemble learning in the development of a computationally efficient, scalable framework for question classification. In comparison to the case study results provided by various Quora case studies [17]; where deep learning methods were used to classify questions (which required high amounts of computational resources and/or high performance hardware), the profiling classification methods used in this study used well developed engineered feature set based on structural and similarity based features that close to the level of accuracy of the classification results of the other two approaches, but required less computational expenses.

Furthermore, a critical comparison with prior studies highlights the efficiency of our proposed framework. While researchers such as [23], [26], [28] and [12] utilized complex Deep Learning architectures, like LSTMs or Transformers, to achieve marginal accuracy gains, our results indicate that an ensemble of traditional machine learning models remains highly competitive. While deep learning models often require massive computational overhead, our "least-cost" ensemble approach provides a scalable and efficient solution.

3.2 Error Analysis

Error analysis was completed to further understand model's functionality; the confusion matrix was used to review how pairs of questions were labeled incorrectly. Some instances were incorrectly predicted to be redundant when they included a strong lexical overlap between the questions but the intent was quite different. Conversely, many instances were misclassified due to semantic similarity expressed using different vocabulary. Examples of misclassified pairs are found in Table 4 as a reference.

https://dx.doi.org/10.4314/njt.v45i1.6s

Table 4: Examples of misclassified question pairs

S/N	Question 1	Question 2	Actual Label	Predicted Label	Error Type	Possible Reason
1	What is the best way to lose weight fast?	How can I reduce body fat quickly?	1	0	False Negative	Low lexical overlap despite semantic similarity (paraphrased expression).
2	What is the age limit for voting in India?	What is the minimum age for marriage in India?	0	1	False Positive	High word overlap ("age", "India") but different intent/context.

4.0 CONCLUSION

The classification of question pairs is major work of this article. This work proposes a useful method for identifying duplicate inquiries and afterwards discovering high-quality solutions due to the usage of least cost architecture and the selection of carefully engineered structural and similarity-based features from the questions. Random Forest outperforms the proposed ML models comprising Decision Tree, XGboost, and Adaboost. Additionally, Random Forest lowers the false positive rate, which is crucial for identifying redundant questions. In future, this model can be extended for question paper moderation at various institutes that reduces the burden of moderators by time saving.

There are future opportunities to enhance the model through the application of contextual embedding strategies (such as Word2Vec and transformer models like BERT) that capture a greater number of the meanings behind the words. A key approach is to build hybrid ML/DL ensemble models, and/or develop automated ensemble methods, that combine deep contextual embedding's with linguistically engineered features. In the future, these methods may continue to improve our semantic comprehension, while keeping the same computational balance and interpretability as outlined in this paper.

REFERENCE

[1] L. Wang, L. Zhang and J. Jiang, "Duplicate Question Detection With Deep Learning in Stack Overflow," *IEEE Access*, 8, pp. 25964–25975, 2020. doi: 10.1109/access.2020.2968391.

[2] Z. Xu and H. Yuan, "Forum Duplicate Question Detection by Domain Adaptive Semantic Matching," *IEEE Access*, 8, pp. 56029–56038, 2020. doi: 10.1109/access.2020.2982268.

[3] D. Basavesha and Y. S. Nijagunaraya, "Detecting Duplicate Questions in Community Based Websites Using Machine Learning," in *Proc. Int. Conf. Innovative Computing & Communication (ICICC)*, University of Delhi, India, 2021. doi: 10.2139/ssrn.3835083.

[4] S. K. Panda, V. Bhalerao and A. R. Sathya, "A Machine Learning Model to Identify Duplicate Questions in Social Media Forums," *Int. J. Innovative Technology and Exploring Engineering*, 9(4), pp. 370–373, 2020. doi: 10.35940/ijitee.D1362.029420

[5] Z. Imtiaz, M. Umer, M. Ahmad, S. Ullah, G. S. Choi and A. Mehmood, "Duplicate Questions Pair Detection Using Siamese MaLSTM," *IEEE Access*, 8, pp. 21932–21942, 2020. doi: 10.1109/access.2020.2969041.

[6] C. Saedi, J. Rodrigues, J. Silva, A. Branco and V. Maraev, "Learning Profiles in Duplicate Question Detection," in *Proc. 2017 IEEE Int. Conf. Information Reuse and Integration (IRI)*, San Diego, CA, USA, pp. 544–550, 2017. doi: 10.1109/iri.2017.39.

[7] V. M. Tambakhe and K. P. Wagh, "Review on Exploring Similarity between Two Questions using Machine Learning," *Int. J. Scientific Research in Computer Science, Engineering and Information Technology*, 7(20), pp. 287–293, 2021. doi: https://doi.org/10.32628/cseit217360

[8] R. R., R. P. Kumar, A. R. S. and A. N. Khan "Identification of Duplication in Questions Posed on Knowledge Sharing Platform Quora using Machine Learning Techniques," *Int. J. Innovative Technology and Exploring Engineering*, 8(12), pp. 2444–2451, 2019. doi: 10.35940/ijitee.L3017.1081219.

[9] S. Rani, A. Kumar, N. Kumar and S. Kumar "Deep Neural Model for Duplicate

Question Detection Using Support Vector Machines (SVM)," *Turkish Journal of Computer and Mathematics Education*, 12(6), pp. 4024–4033, 2021.

[10] K. Sharma and S. K. Tadepalli "Detecting Duplicate Questions in Online Forums Using Machine Learning Techniques," *Int. J. Research in Applied Science & Engineering Technology*, 10, pp. 4775–4778, 2022. https://doi.org/10.22214/ijraset.2022.45072

[11] Y. Zhang, D. Lo, X. Xia and J.-L. Sun "Multi-Factor Duplicate Question Detection in Stack Overflow," *Journal of Computer Science & Technology*, 30(5), pp. 981–997, 2015. doi: 10.1007/s11390-015-1576-4.

[12] T. Addair, "Duplicate Question Pair Detection with Deep Learning," Department of Computer Science, Stanford University, Stanford, CA, USA, 2017. Available: Stanford University PDF.

[13] N. Ansari and R. Sharma "Identifying Semantically Duplicate Questions Using Data Science Approach: A Quora Case Study," *arXiv preprint, abs/2004.11694*, 2020. [Online]. Available: https://arxiv.org/abs/2004.11694

[14] N. N. Qomariyah, E. Heriyanni, A. N. Fajar and D. Kazakov "Comparative Analysis of Decision Tree Algorithm for Learning Ordinal Data Expressed as Pairwise Comparisons," in *Proc. 8th Int. Conf. Information and Communication Technology (ICoICT)*, Yogyakarta, Indonesia, pp. 1–4, 2020. doi: 10.1109/Icoict49345.2020.9166341.

[15] J. K. Jaiswal and R. Samikannu "Application of Random Forest Algorithm on Feature Subset Selection and Classification and Regression," in *Process 2017 World Congress on Computing and Communication Technologies (WCCCT)*, Tiruchirappalli, India, pp. 65–68, 2017. doi: 10.1109/wccct.2016.25.

[16] C. N. Obiora, A. Ali and A. N. Hasan "Implementing Extreme Gradient Boosting (XGBoost) Algorithm in Predicting Solar Irradiance," in *Proc. 2021 IEEE PES/IAS PowerAfrica*, Nairobi, Kenya, pp. 1–5, 2021. doi: 10.1109/Powerafrica52236.2021.9543159.

[17] R. Wang "AdaBoost for Feature Selection, Classification and Its Relation with SVM: A Review," *Physics Procedia*, 25, pp. 800–807, 2012. doi: 10.1016/j.phpro.2012.03.160.

[18] C. Tang, B. Xu and H. Liu "The Application of the AdaBoost Algorithm in Text Classification," in *Proc. 2nd IEEE IMCEC*, Xi'an, China, pp. 1792–1796, 2018. doi: 10.1109/imcec.2018.8469497.

[19] T. P. Nagarhalli, V. Vaze and N. K. Rana "Impact of Machine Learning in Natural Language Processing: A Review," in *Proc. Third Int. Conf. Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, Tirunelveli, India, pp. 1529–1534, 2021. doi: 10.1109/ICICV50876.2021.9388380.

[20] S. Li, J. Zhang and Y. Chen "Further Improvement of AdaBoost Algorithm," in *Proc. Seventh Int. Conf. Measuring Technology and Mechatronics Automation (ICMTMA)*, pp. 499–501, 2015. doi: 10.1109/icmtma.2015.127.

[21] "Quora Question Pairs Dataset" *Kaggle*. [Online]. Available: https://www.kaggle.com/competitions/quora-question-pairs/data. Accessed on 2025.

[22] P. A. Kumar, E. Pugazhendhi and K. V. Lakshmi "Cloud Data Storage Optimization by Using Novel De-Duplication Technique," in *Proc. 2022 4th Int. Conf. Smart Systems and Inventive Technology (ICSSIT)*, Tirunelveli, India, pp. 436–442, 2022. doi:10.1109/icssit53264.2022.9716508.

[23] S. Tarek, H. M. Noaman and M. Kayed "Enhancing Question Pairs Identification with Ensemble Learning: Integrating Machine Learning and Deep Learning Models," *Int. J. Advanced Computer Science and Applications*, 14(11), pp. 600–608, 2023, doi:10.14569/ijacsa.2023.01411100

[24] N. B. Korade, M. B. Salunke et al. "Exploring NLP Techniques for Duplicate Question Detection to Maximizing Responses on Q&A Websites," *Int. J. Intelligent Systems and Applications in Engineering*, 12(3), pp. 11–20, 2024. doi: 10.51463/ijisae.v12i2s1.5218.

[25] R. P. Kumar, B. M. G., R. Elakkiya and V. Druva "Exploring Machine Learning Models for Duplicate Question Detection in Online Communities," in *Proc. 2023 Inter Conference Computer Science and*

Engineering, Bengaluru, India, 2023. doi: 10.1109/icaecc59324.2023.10560222.

[26] A. Bhardwaj, R. Hasan and S. Mahmood "Semantic similarity in community forum questions: Case study on Quora dataset," *Journal of Umm Al-Qura University for Engineering and Architecture*, 16, pp.1719–1728, 2025. doi: 10.1007/s43995-025-00206-0.

[27] J. Martinez-Gil "Automatic Design of Semantic Similarity Ensembles Using Grammatical Evolution," *arXiv preprint arXiv:2307.00925v8*, 2025. doi:10.48550/arxiv.2307.00925.

[28] L. Yu, C. Che, B. Liu, Q. Lin and X. Zhao "Semantic Similarity Matching for Patent Documents Using Ensemble BERT-related Model and Novel Text Processing Method," *arXiv preprint arXiv:2401.06782*, 2024. doi:10.48550/arxiv.2401.06782.

[29] James, I. I. ., & Osubor, V. I. "Hostile social media harassment: A machine learning framework for filtering anti-female jokes," *Nigerian Journal of Technology*, 41(2), pp. 342–350, 2022. doi: https://doi.org/10.4314/njt.v41i2.13

[30] B. S. Yelure, N. S. Deokule, S. S. Mane, M. V. Bhosale, A. B. Chavan and V. C. Satpute "Remote monitoring of Covid-19 patients using IoT and AI," *2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, Coimbatore, India, pp. 73-80, 2022. doi: 10.1109/icais53314.2022.9742750