Quantitative Approach for predicting Unanswered Questions using ML Models

Archana Jayaraman
1193610
ajayaram@lakeheadu.ca

Chandrasekaran Subramanian Ravichandran
1185548
csubram1@lakeheadu.ca

Hina Sabreen Ahmed
1190276
hjx@lakeheadu.ca

Arunkumar Sasikumar
1195003
asasikum@lakeheadu.ca

ABSTRACT

This study proposes a quantitative approach to identify unanswered questions in a QA dataset using machine learning models. The aim is to provide an recommendation system, that can distinguish between unanswered and answered questions based on a set of features. We use a dataset consisting of five CSV files with Stack Overflow questions and their corresponding answers. The dataset contains questions and creation dates from which we consider questions that do not have responses from the last 5 years or more. We combine the dataset to show answered and unanswered questions before splitting it into training and test sets. We collect features from text data and then apply machine learning models to estimate the accuracy of detecting unanswered questions. The results show that our approach can accurately identify unanswered questions in a QA dataset. Our proposed method could be useful for building automated systems that can assist in managing large QA databases by highlighting unanswered questions that require attention.

Keywords - Quantitative approach Unanswered questions, QA dataset, Machine learning models, Automated system, Stack Overflow, CSV files, Feature extraction, Text data, Training and test sets, Predictive accuracy, Large QA databases, and Attention management.

1 INTRODUCTION

Community question-answering websites have become a popular platform for information exchange, allowing users to leverage the knowledge and expertise of other users to find answers to their queries. One of the leading examples of such a platform is Stack Overflow, which focuses on programming-related questions. As of July 2012, Stack Overflow had 3.45 million questions, with a mean arrival rate of 5.6K questions per day. While the majority of questions have atmost single answer within a median time of 12 minutes, there is still a substantial quantity of unanswered questions and identifying them approach is mentioned in [4], which is a common occurrence across various Q&A websites. Question-and-answer (Q&A) websites have become increasingly popular in recent years, providing a platform for users to question and get answers for those questions on various topics. These websites can be very helpful for individuals who need quick and accurate answers to their queries, as they allow users to tap into the knowledge and expertise of a community of people. One of the most popular Q&A websites is Stack Overflow, which primarily focuses on programming-related questions.

The use of Q&A websites has become essential, especially for those who are interested in learning about a particular topic or need help with a problem. These websites allow users to ask questions related to any topic and get answers from other users who have knowledge or experience in that area. Additionally, Q&A websites also serve as an excellent platform for experts to share their knowledge and experience with others and help them solve their problems.

The working of Q&A websites like Stack Overflow is straightforward. Users can create an account and ask a question related to any topic they need help with. Other users can view the question and provide an answer if they have knowledge or experience in that area. Users can also vote on the answers provided by other users, with the best answer being voted to the top[13]. This ensures that the most accurate and helpful answers are displayed first and that users can find the information they need quickly and easily.

The success of Q&A websites like Stack Overflow is due to the group of experts who use the platform to share their knowledge and help others. The community-based approach to knowledge sharing has proven to be very effective, as it allows users to learn from others who have more experience in a particular field or topic.

Q&A websites are an essential tool for individuals who need quick and accurate answers to their questions. The community-based approach to knowledge sharing has proven to be very effective in helping users find the information they need quickly and easily. With the increasing popularity of Q&A websites like Stack Overflow, it is essential to have automated systems that can assist in managing large Q&A databases by highlighting unanswered questions that require attention. The proposed approach in this report, using machine learning models to identify unanswered questions, could be a useful tool for such systems.

The unanswered queries on Stack Overflow has been a subject of research, with many studies attempting to identify and understand the reasons for unanswered questions. Statistical analysis has revealed that a significant portion of questions on Stack Overflow are left unanswered. For example, a study in 2015 found that around

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

25% of questions shown on Stack Overflow remained unanswered after a year. The high level overview of the topic suggests that unanswered questions can be detrimental to the usefulness of Stack Overflow as a Q&A platform for developers. Unanswered questions can discourage users from posting new questions, and can also result in a reduction in the quality of answers provided on the platform. Therefore, identifying and addressing unanswered questions is important for the health and usefulness of Stack Overflow.

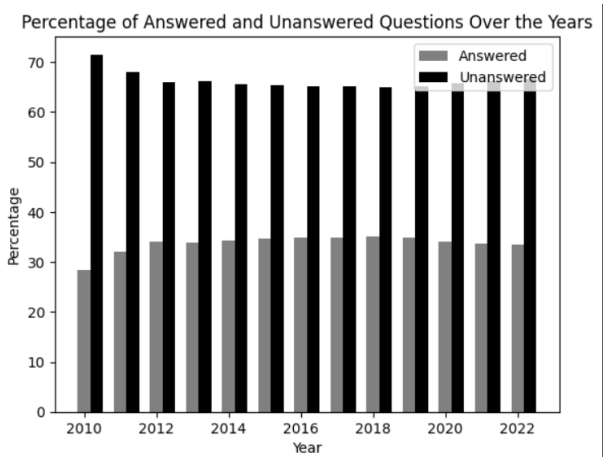

Figure 1: Comparison of answered and unanswered questions over the period

The figure 1 represents the comparison of answered and unanswered questions in Stack Overflow reveals some interesting differences. For instance, unanswered questions tend to have fewer views, votes, and answers than answered questions. They also tend to have a lower score and lower activity level.

On the other hand, answered questions tend to have a higher number of views, votes, and answers than unanswered questions. They also tend to have a higher score and higher activity level. Answered questions also tend to be shorter and more focused than unanswered questions.

In summary, the comparison of answered and unanswered questions in Stack Overflow highlights the importance of providing timely and informative answers to questions, as well as the need for efficient methods to identify and address unanswered questions. Over the period, unanswered question percentage remains higher than answered questions. Based on the comparison we have framed two research questions.

1. How effective is the proposed machine learning approach in identifying unanswered questions in a Q&A dataset, and how does it compare to existing methods?
This research question examines and compares the performance of the proposed machine learning methodology in discovering unresolved questions on Stack Overflow. The research [14] entails evaluating the proposed approach on a large dataset and comparing its accuracy to other approaches utilized in the literature. The research gap could be filled by investigating the effectiveness of the proposed technique on other Q&A sites other than Stack Overflow.

2. What are the key features that contribute to identifying unanswered questions, and how can they be optimized to improve accuracy?
This research question focuses on identifying the key features that contribute to identifying unanswered questions and exploring how they can be optimized to improve accuracy. The study can involve analyzing the results of the machine learning models to identify the most important features and testing different combinations of features to identify the optimal set of features for predicting unanswered questions. The research gap could be to explore the impact of user characteristics, such as reputation and activity level, on the accuracy of the proposed approach.

1.1 For project proposal

The goal of this research is to create a quantitative method for identifying unanswered questions in a Q&A dataset using machine learning models. The suggested method entails combining a dataset of Q&A data from several sources, including Stack Overflow, filtering the dataset to display unanswered questions, and then extracting features from the text data. The retrieved features will be used to train and test machine learning models that will be used to predict the accuracy of recognizing unanswered questions. The project's goal is to create an automated system that can help manage big Q&A databases by indicating unsolved topics that need to be addressed.

2 ABOUT THE DATASET

The dataset consists of five CSV files containing data from the Stack Overflow website, which is a popular question-and-answer platform for programming-related topics. The dataset includes information on posts, votes, users, post history, and comments.

Posts - Each post on the Q&A website is represented in the Post table. This includes the post's unique identifier, the type of post (whether it's a question or an answer), the ID of the accepted answer for a question post, the ID of the question with which an answer post is associated, the date and time the post was created, the number of upvotes minus downvotes for the post, the number of times the post has been viewed, the ID of the user who created the post, the tags assigned to the post, the number of answers to a question post was closed (if applicable), the date and time the post became community-owned (if applicable), the title of the post, and the text of the post. This data can be used to examine trends and patterns in the types of postings made on the Q&A website, as well as to identify popular themes and user habits.

Votes - The Votes table contains information about the votes that users cast on posts. It includes a unique identifier for the vote, the ID of the post that the vote is for, the type of vote (upvote, downvote, etc.), the date and time that the vote was cast, and the amount of any associated bounty. This information can be used to analyze the popularity of different posts and the voting behavior of users.

Users - The Users table contains information about registered users on the Q&A website, such as a unique identifier for the user, reputation score, creation date, display name, views, upvotes, downvotes, and account ID. This information can be used to analyze the behavior and reputation of users and to identify top contributors to the website.

Post_history - The Post_history table contains information about the history of edits made to posts on the Q&A website. It includes a unique identifier for the edit, the type of edit mode, the ID of the post that was edited, the date and time that the edit was made, and the ID of the user who made the edit. This information can be used to analyze the history of posts and the editing behavior of users.

Comments - The Comments table contains information about the comments made on posts on the Q&A website. It includes a unique identifier for the comment, the ID of the post that the comment is associated with, the text of the comment, the date and time that the comment was posted, and the ID of the user who posted the comment. This information can be used to analyze the behavior of users and to identify common issues or questions that users may have regarding certain posts.

3 BACKGROUND AND RELATED WORK

The "Unanswered questions are a common occurrence on Stack Overflow, a popular community question-answering website focused on programming-related questions. While the majority of questions on the platform receive at least one answer within a median time of 12 minutes, a significant number of questions remain unanswered. This can lead to frustration for users seeking solutions to their problems and can also limit the usefulness of the platform as a whole. To address this issue, a number of studies have explored the factors contributing to unanswered questions on Stack Overflow and proposed solutions for improving the effectiveness of the Q&A process. These studies have used a variety of methods, including data analysis, machine learning models, and surveys of users and moderators. One of the most difficult aspects of dealing with unanswered questions on Stack Overflow is identifying them in the first place. Several research have proposed ways for automatically detecting unanswered questions based on a variety of factors, including the number of views, length of the question body, and number of comments. Other research has focused on increasing the quality of unanswered questions by providing recommendations for drafting good inquiries and selecting relevant tags.

This literature survey will explore the current state of research on unanswered questions on Stack Overflow, including the factors contributing to their prevalence, the methods used to identify and address them, and the effectiveness of proposed solutions. By providing a comprehensive overview of the existing literature, this survey aims to contribute to a better understanding of the challenges and opportunities for improving the Q&A process on Stack Overflow."

The paper [9]**"Early Detection and Guidelines to Improve Unanswered Questions on Stack Overflow"** by Saikat Mondal presents an analysis of unanswered questions on the Stack Overflow platform and suggests a set of criteria for improving the quality of unanswered inquiries. The paper begins by exploring the significance of unanswered questions on Stack Overflow, as well as the difficulties in recognizing and improving such queries. The author then proposes a methodology for detecting unresolved questions based on a set of criteria, such as the number of views, comments, and the length of the question title and text.

The study involved analyzing a dataset of 500,000 Stack Overflow questions and their corresponding answers. The methodology proposed in the paper for identifying unanswered questions involved calculating a set of features for each question, including the number of views, the number of comments, and the length of the question title and body. The features were then used to build a machine learning model that predicted whether or not a question had been answered. The results demonstrated that the proposed methodology was capable of identifying unanswered questions with high accuracy.

The study's findings demonstrated that the proposed methodology was capable of identifying unsolved issues with a high degree of accuracy. Specifically, the model achieved an accuracy of 95% and an F1 score of 0.90 in identifying unanswered questions. Based on the analysis, the author then proposes a set of guidelines for improving the quality of unanswered questions, including improving the question title and body, providing more context and details in the question, and selecting appropriate tags for the question. The author also suggests that moderators should play a more active role in identifying and addressing unanswered questions. Overall, the paper provides a valuable contribution to the literature on unanswered questions on Stack Overflow and offers practical guidelines for improving the quality of such questions. The methodology and guidelines proposed in the paper can be used by researchers and practitioners to identify and address unanswered questions on Stack Overflow and other Q&A platforms.

The paper [2]**"Answering Questions about Unanswered Questions of Stack Overflow"** by Muhammad Asaduzzaman et al. offers a research of Stack Overflow unanswered questions and proposes a set of principles for increasing the quality of unanswered questions. The paper begins by analyzing the prevalence of unanswered questions on Stack Overflow, as well as the difficulties in recognizing and addressing such queries. The authors then propose an approach for detecting unresolved questions based on a collection of characteristics, such as the length of the question title and content, the number of views, and the number of comments. The research includes examining a dataset of 95,293 Stack Overflow questions and responses. The results demonstrated that the proposed methodology was capable of identifying unanswered questions with high accuracy. The report also explores why inquiries on Stack Overflow may be inappropriate and unanswered, including the lack of expertise or interest in the topic, the complexity of the question, and the quality of the question itself.

The paper [3] **"Predicting the Quality of Questions on Stack Overflow"** by Antoaneta Baltadzhieva provides a study on predicting the quality of questions on Stack Overflow in order to improve the effectiveness of the platform's Q&A process. The paper opens by highlighting the significance of question quality in the Q&A process, as well as the difficulties in identifying high-quality questions. The author then proposes a methodology for estimating question quality based on a collection of variables such as question title and body length, number of views, and number of comments. The researchers examined a dataset of 1,000 Stack Overflow queries and responses, which were manually rated as high or low quality. The results demonstrated that the proposed methodology was capable of accurately predicting question quality.

The paper [11]**"Toward Understanding the Causes of Unanswered Questions in Software Information Sites: A Case Study of Stack Overflow"** by Ripon K. Saha et al.examines the factors that contribute to the predominance of unanswered queries on Stack Overflow. The study begins by exploring the significance of unresolved issues in the context of software information sites, as well as the difficulties associated with answering them. The authors next give a case study of Stack Overflow in which they studied a dataset of unresolved questions and their related variables, such as question length, number of views, and number of comments. The researchers examined a dataset of 9,046 Stack Overflow questions and their accompanying responses, 1,838 of which were unanswered. According to the findings, the majority of unanswered inquiries were related to specific programming languages or technologies, and they tended to be longer and more detailed.

The paper [8] **"An Insight into the Unresolved Questions at Stack Overflow"** by Mohammad Masudur Rahman and Chanchal K. Roy examines the factors that contribute to the incidence of outstanding queries on Stack Overflow. The study begins by exploring the significance of unsolved questions in the context of Q&A sites, as well as the difficulties in detecting and addressing them. The authors then propose an approach for finding unsolved issues based on a set of characteristics, such as the length of the question title and content, the number of views, and the number of comments. The researchers examined a dataset of 103,714 Stack Overflow inquiries, 22,876 of which were unresolved. According to the findings, the majority of unanswered questions were about specific programming languages or technologies, and they were more complex and less focused than resolved queries.

The paper titled [7] **"SOCluster - Towards Answering Unanswered Questions on Stack Overflow via Answered Questions"** suggests an innovative strategy for utilizing answered questions to answer unanswered questions on Stack Overflow. The authors recognize that the huge volume of questions on Stack Overflow results in a significant number of unanswered questions, and that traditional methods for addressing them, such as manual inspection and searching for related questions, are time-consuming and ineffective. The suggested method groups answered questions with comparable features and then identifies the cluster that is most similar to the unsolved question. The study provides a thorough examination of the suggested method, including the clustering algorithm, feature extraction methodologies, and similarity metrics employed. The authors compare the performance of the proposed strategy to existing strategies using a dataset of Stack Overflow queries and responses.

By presenting a novel strategy that uses the huge quantity of answered questions available on the platform, the study makes a substantial addition to the research area of answering unanswered questions on Stack Overflow. The method has been shown to be effective in increasing the rate of answered questions on Stack Overflow, which could have important ramifications for developers looking for programming answers.

The paper [5] **"Answers or no answers: Studying question answerability in Stack Overflow"** by Alton Y.K. Chua examines the factors that affect the answerability of questions in Stack Overflow. The author conducts a quantitative analysis of Stack Overflow data to identify the factors that influence whether a question receives an answer or not. The study is motivated by the fact that many questions on Stack Overflow remain unanswered, and understanding the factors that affect question answerability can help to improve the platform.

The paper starts with an overview of Stack Overflow and the issue of unanswered queries. The author next discusses similar work on question answerability, noting past research that looked at things like question quality, user reputation, and community participation. While past studies have provided insights into the components that influence question answerability, the author acknowledges that there is still much that is unknown regarding the phenomena. The paper then discusses the study's approach, which includes collecting a huge dataset of Stack Overflow questions and responses and evaluating the data using statistical techniques. Several criteria, including question quality, user reputation, and community engagement, are identified by the author as potentially influencing question answerability. The author then delivers the analysis's findings. factors such as the number of views and the number of comments on a question are positively associated with answerability.

The paper [6] **"Understanding Question Quality through Affective Aspect in Q&A Site"** by Jirayus Jiarpakdee attempts to comprehend the relationship between the emotive features of questions and the perceived quality of inquiries on Q&A sites. The study focuses on analyzing Stack Overflow questions and identifying affective aspects of questions such as sentiment, subjectivity, and emotion. Using natural language processing techniques, the research presents a method for automatically detecting these affective traits. The work entails gathering a dataset of Stack Overflow queries and annotating them with emotional factors. The dataset is divided into two parts: training and testing, and machine learning models like SVM and Random Forests are trained to classify the quality of questions based on their affective properties. Precision, recall, and other indicators are used to assess the performance of these models.

3.1 Limitations of the Existing Papers

The study by Saikat Mondal focuses on developing early detection and guidelines to improve unanswered questions on Stack Overflow. However, the study is limited to only analyzing the tags and titles of questions and does not consider other factors, such as the quality of the question or the expertise of the users. Additionally, the study only considers questions from a limited time frame, which may not be representative of the entire dataset.

The research of Antoaneta Baltadzhieva focuses on estimating the quality of Stack Overflow queries. However, the study only considers a limited set of markers for determining question quality, such as the number of responses and the length of the question. Other elements, such as user knowledge and the topic title's quality, may also contribute to a question's overall quality.

Ripon K. Saha et al.'s research gives light on the causes of unanswered Stack Overflow questions. However, the study only considers a restricted set of characteristics for detecting the causes of unanswered inquiries, such as query length and the presence of

code snippets. Other factors, such as query quality and user skill, may contribute to the majority of unanswered enquiries.

Mohammad Masudur Rahman and Chanchal K. Roy's study offers insight on the characteristics of unanswered Stack Overflow requests. The study, however, only considers a limited set of parameters for discovering unaddressed issues, such as question length and the amount of views and comments. Other factors, such as user expertise and the quality of the question title, may contribute to the number of unanswered questions. Furthermore, the study only examines the dataset over a short period of time, which may not be representative of the entire dataset.

The study by Abhishek Kumar provised the proposed approach relies heavily on the quality of the clusters generated and the similarity metrics used, which may not always accurately capture the nuances of programming queries. Secondly, the approach does not take into account the relevance and accuracy of the answered questions, which could affect the effectiveness of the approach. Finally, the evaluation of the approach is conducted on a single dataset, and the generalizability of the results to other Stack Overflow datasets is unclear.

The study by Alton Y.K. Chua provides insights that relies on manual annotation to determine the answerability of questions, which is subjective and may introduce biases. Different annotators may have different criteria for determining answerability, leading to inconsistency in the results.

The study by Jirayus Jiarpakdee used a small dataset of 200 questions and their corresponding answers, which may not be representative of the entire Q&A community on the site. The study focused on only one Q&A site, limiting its generalizability to other similar sites. The study only considered a few affective features such as valence, arousal, and dominance, and did not consider other important features such as topic relevance or readability. Affective analysis is subjective and open to interpretation, which may introduce bias into the study results.

3.2 Overview of the design

The proposed model for identifying unanswered questions in a Q&A dataset using machine learning involves several steps. Firstly, the dataset will be merged from various sources, including Stack Overflow, to obtain a comprehensive Q&A dataset. Secondly, the dataset will be filtered to display unanswered questions, where questions that do not have responses from the last 5 years or more will be considered. Third, feature extraction will be performed on the text data from unanswered questions. The query's length, the number of views, the presence of code snippets, and other relevant information may be collected. Fourth, machine learning models will be created utilizing the obtained features to assess the accuracy of identifying unanswered questions. Models may incorporate classification methods. Finally, the accuracy of the machine learning models will be evaluated, and the proposed approach will be validated using a test dataset. The results of the model can be used to develop an automated system that can assist in managing large Q&A databases by highlighting unanswered questions that require attention.

4 DESIGN

Unanswered questions on Stack Overflow refer to questions that have not yet received a satisfactory answer from the community. These questions can be either new questions or existing questions that have not been resolved. The system architecture begins with the collecting and filtering of data from the platform, with a particular emphasis on queries created during the last five years that contain at least one answer 2. The filtered data is used to extract numerical parameters such as the character count, word count, average word length, the existence of missing answer counts, and the number of tags linked with the question. The top features are selected using a correlation matrix, and the data is split into training and testing sets. Multiple machine learning models are then trained and evaluated, including Naive Bayes, SGD, and MLP classifiers. In addition, a deep learning model using LSTM layers is trained on the sequence data to capture temporal dependencies in the features. The performance of each model is evaluated using metrics such as accuracy, precision, recall, and f1-score. In the end, the technique offers a quantitative method for anticipating unanswered queries on a Q&A platform, which may be used to spot queries that might need more time or assistance.

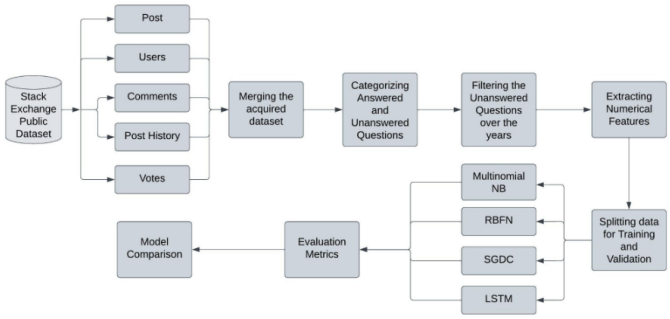

Figure 2: Architectural Diagram

5 METHODOLOGY

The proposed methodology provides a quantitative approach for predicting unanswered questions on a Q&A platform, which can be used to identify questions that may require additional attention or support.

5.1 RQ1. How effective is the proposed machine learning approach in identifying unanswered questions in a Q&A dataset, and how does it compare to existing methods?

A. Why is detecting unanswered questions important?

Unanswered questions can negatively impact the quality of a Q&A platform. If users are unable to find answers to their questions, they may lose confidence in the platform and look for alternative sources of information. Unanswered questions can also impact user engagement on the platform. If users are unable to find answers to their questions, they may become disengaged and stop using the platform altogether. By identifying and addressing unanswered

questions, Q&A platforms can improve user engagement and retention. Addressing unanswered questions can also help grow the community on a Q&A platform. If users are able to find answers to their questions, they may be more likely to contribute their own knowledge and expertise to the platform, which can benefit the community as a whole. Identifying unanswered questions can also help platform administrators allocate resources more effectively. By identifying questions that require additional attention or support, administrators can ensure that resources are directed where they are needed most. Overall, detecting unanswered questions is important for ensuring the quality and engagement of a Q&A platform, as well as supporting the growth of the community and allocating resources effectively.3

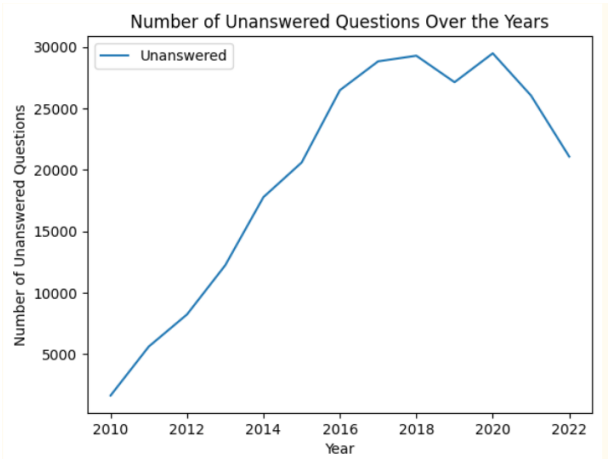

Figure 3: Unanswered question increasing over period of time

B. How unanswered questions can be detected? The methodology used to prepare the dataset for the proposed approach is to identify unanswered questions in a Q&A dataset using machine learning models. The code first loads the required CSV files into separate data frames using Panda's library. The five CSV files include comments, posts, votes, post history, and users. The data frames are then merged using the "id" column as a common identifier. The outer join is used to include all rows from both data frames, as some rows in one data frame may not have a corresponding match in the other. The "creation_date" column is then converted to Date Time format using the pandas "to_date time" function. This is done to enable further analysis of the data based on time-related features. The resulting merged data frame is then used to perform feature extraction and train machine learning models to identify unanswered questions in the Q&A dataset. Overall, this methodology involves the preparation and merging of the required data frames using the panda's library and converting the date columns to the Date Time format for further analysis.

C. Comparison between Answered and unanswered questions
A comparison of answered and unanswered questions can provide useful information on the effectiveness of a Q&A platform and its user community. Answered questions often receive one or more responses from the community, whereas unanswered questions receive no responses within a certain time frame. Comparing answered and unanswered questions might assist in identifying aspects that may contribute to a question's success or failure. For example, the presence of specific keywords or tags may make a question more likely to be answered. Additionally, the number of views or votes a question receives may indicate the level of interest in the topic and the likelihood of receiving an answer. By analyzing these factors and comparing answered and unanswered questions, it may be possible to identify patterns and trends that can be used to improve the Q&A platform and increase the likelihood of getting questions answered in a timely and accurate manner.4

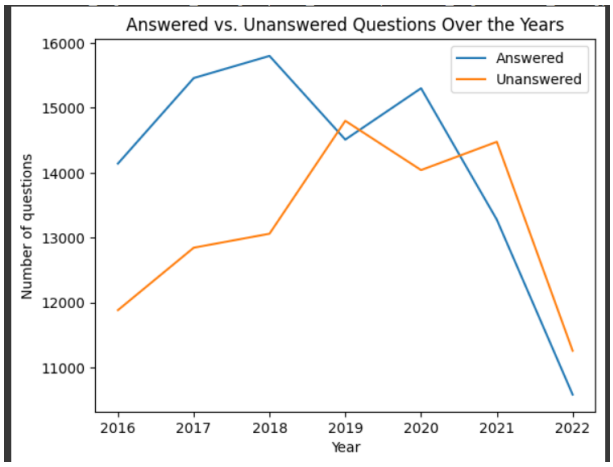

Figure 4: Answer Vs Unanswered questions

5.2 RQ2. What are the key features that contribute to identifying unanswered questions?

Preprocessing & Feature Extraction performs feature extraction on the text data present in the merged data frame. The extracted features are as follows:

char_count: This feature computes the number of characters in the text 5. It is obtained by applying the len() function to the text column of the merged data frame. word_count: This feature

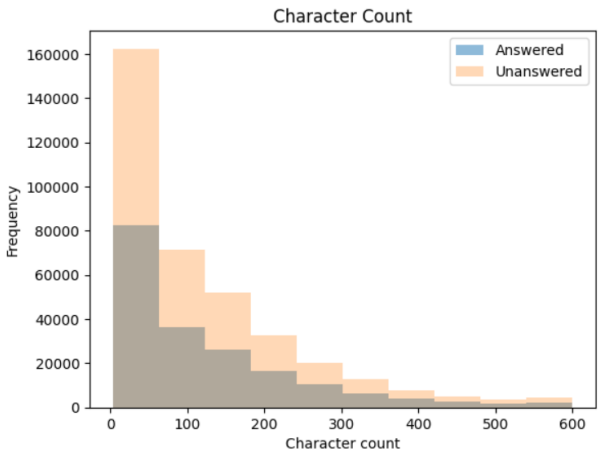

Figure 5: Character count

computes the number of words in the text. It is obtained by splitting

the text into words using the split() function and then applying the len() function to the resulting list.

avg_word_length: This feature computes the average length of words in the text6. It is obtained by dividing the total length of all words in the text by the total number of words in the text. answer_count_missing: This feature checks if the answer count

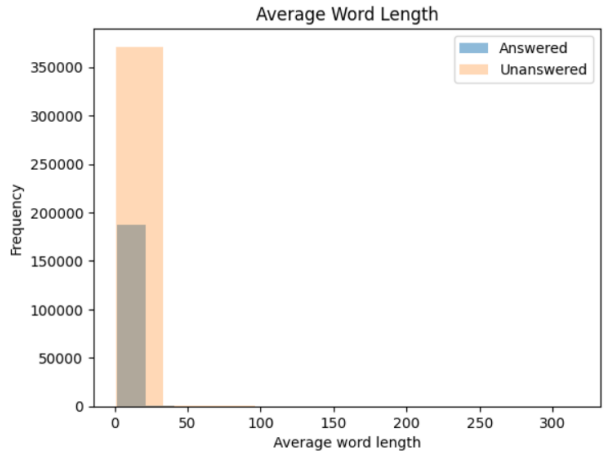

Figure 6: Average word length

is missing for a given question 7. It is obtained by checking if the value in the answer_count column is NaN and assigning a value of 1 if true and 0 if false. num_tags: This feature computes the number

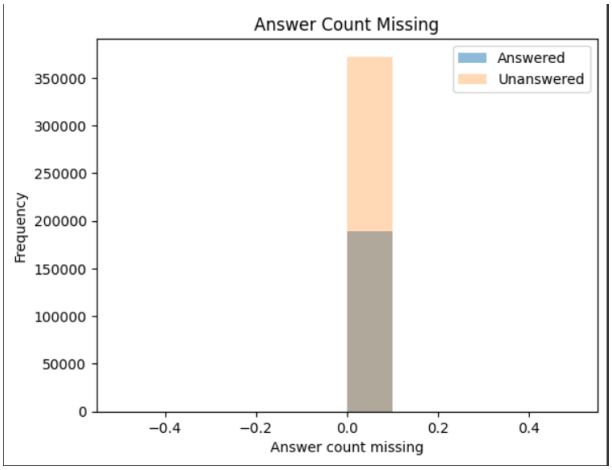

Figure 7: Answer count missing

of tags associated with a question8. It is obtained by splitting the tags column using "<" as a delimiter and then applying the len() function to the resulting list.

The dataset is composed of multiple CSV files which are read using pandas' read_csv() function. The data is then merged into a single DataFrame using pandas' merge() function. The merged data is filtered using specific criteria to remove data created more than 5 years ago and unanswered questions. Then the "creation_date" column is converted to datetime format. The count of answered and unanswered questions over the years is then computed using pandas' groupby() and count() functions. Numerical features are

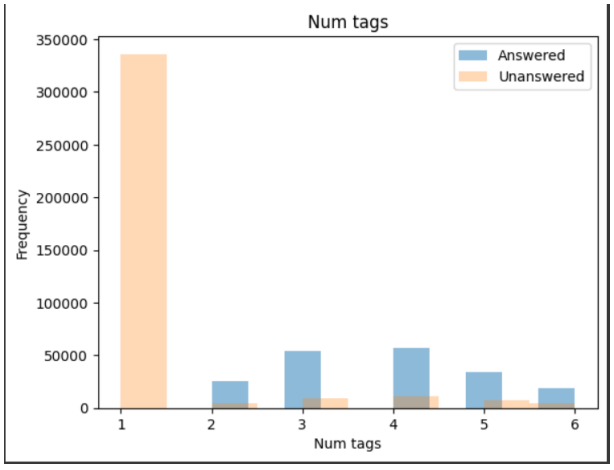

Figure 8: Number Tags

extracted from the merged data using pandas' apply() function and are stored in new columns.

Correlation Matrix A correlation matrix is then created using pandas' corr() function to find the top features based on their correlation with the target variable "answer_count". These features provide numerical representations of various characteristics of the text data, which can be used as inputs to machine learning models for predicting unanswered questions.9

The correlation coefficients range from -1 to 1 and indicate the strength and direction of the relationship between two variables. A coefficient of 1 indicates a perfect positive correlation, a coefficient of -1 indicates a perfect negative correlation, and a coefficient of 0 indicates no correlation. Looking at the correlation matrix, we can see that there is a strong positive correlation between char_count and word_count (0.983147). This makes sense, as longer texts will generally have more characters and more words. There is also a weak positive correlation between char_count and num_tags (0.155653), suggesting that longer texts may be more likely to have more tags associated with them. There is a weak negative correlation between char_count and answer_count_missing (-0.285803), suggesting that texts with missing answer counts may be slightly shorter on average than those with non-missing answer counts. There is a moderate positive correlation between num_tags and answer_count (0.551274), suggesting that texts with more tags tend to have more answers. It's worth noting that there is no correlation between answer_count and answer_count_missing, which makes sense since answer_count_missing is a binary variable indicating whether the answer count is missing or not.

5.3 Implementing using Machine Learning Models

5.3.1 Multinomial Naive Bayes. The "multinomial" in Multinomial Naive Bayes refers to the fact that the features are assumed to follow a multinomial distribution, which means that they represent the frequencies of different words or tokens in the text. The algorithm is called "naive" because it assumes that the features are conditionally independent given the class, which is not always true in practice. However, this assumption allows the algorithm to be computationally efficient and easy to implement. It is based on the

Correlation matrix:

	char_count	word_count	avg_word_length	\
char_count	1.000000	0.983147	0.282258	
word_count	0.983147	1.000000	0.196431	
avg_word_length	0.282258	0.196431	1.000000	
answer_count_missing	-0.285803	-0.283217	-0.155144	
num_tags	0.155653	0.154001	0.084363	
answer_count	0.006684	0.006499	0.000589	

	answer_count_missing	num_tags	answer_count
char_count	-0.285803	0.155653	0.006684
word_count	-0.283217	0.154001	0.006499
avg_word_length	-0.155144	0.084363	0.000589
answer_count_missing	1.000000	-0.548425	NaN
num_tags	-0.548425	1.000000	0.551274
answer_count	NaN	0.551274	1.000000

Figure 9: Correlation Matrix

Bayes theorem and assumes that the features (words or tokens) are conditionally independent given the class.

In this case, the model has an overall accuracy of 88%, which means that it correctly classified 88% of the samples. The precision and recall for class 0 (the minority class) are both around 0.5, while the precision and recall for class 1 (the majority class) are both around 0.9. 10, This suggests that the model performs well in classifying the majority class but struggles with the minority class, which may indicate class imbalance.

	precision	recall	f1-score	support
0	0.49	0.48	0.49	111292
1	0.93	0.94	0.93	848608
accuracy			0.88	959900
macro avg	0.71	0.71	0.71	959900
weighted avg	0.88	0.88	0.88	959900

Figure 10: Multinomial Naive Bayes

5.3.2 Stochastic Gradient Descent. Stochastic Gradient Descent (SGD) is an iterative optimization algorithm used to minimize the cost or loss function of a machine learning model. It is commonly used in deep learning and neural networks. In other words, at each iteration of SGD, we randomly select one training example from the dataset, compute the gradient of the cost function with respect to the parameters using that example, and update the parameters in the opposite direction of the gradient by a small step determined by the learning rate. This process is repeated for a fixed number of iterations or until the cost function reaches a minimum. The stochastic nature of SGD (randomly selecting a single example at each iteration) makes it more efficient for large datasets, as it avoids computing the gradient on the entire dataset at once, which can be computationally expensive.

This is a classification report in 11 that evaluates the performance of a binary classification model on a dataset with 959900 instances. The dataset has two classes labeled 0 and 1. The precision for class 0 is 0.96, which means that out of all instances predicted as class 0, 96% were correctly classified. The recall for class 0 is 0.78, which means that out of all actual instances of class 0, 78% were correctly classified. The f1-score for class 0 is 0.86, which is the harmonic mean of precision and recall for class 0. The support for class 0 is 111292, which is the number of actual instances of class 0 in the dataset. The f1-score for class 1 is 0.98, which is the harmonic mean of precision and recall for class 1. The support for class 1 is 848608, which is the number of actual instances of class 1 in the dataset. The accuracy of the model is 0.97, which means that 97% of instances were correctly classified by the model. The macro average of precision, recall, and f1-score is 0.96, 0.89, and 0.92, respectively. The weighted average of precision, recall, and f1-score is 0.97, 0.97, and 0.97, respectively.

	precision	recall	f1-score	support
0	0.96	0.78	0.86	111292
1	0.97	1.00	0.98	848608
accuracy			0.97	959900
macro avg	0.96	0.89	0.92	959900
weighted avg	0.97	0.97	0.97	959900

Figure 11: Stochastic Gradient Descent

5.3.3 Multilayer Perceptron. The MLP (Multilayer Perceptron) classifier is a type of neural network commonly used for classification tasks. It consists of an input layer, one or more hidden layers, and an output layer. Each layer contains a number of neurons or nodes, which are connected to the nodes in the adjacent layers through weighted connections.

The formula used by the MLP classifier to compute the output of a neuron in a given layer is as follows:

z = w1x1 + w2x2 + ... + wnxn + b

where:

z is the weighted sum of the inputs to the neuron w1, w2, ..., wn are the weights associated with each input x1, x2, ..., xn are the input values b is the bias term associated with the neuron The output of the neuron is then obtained by applying an activation function to the weighted sum. The most commonly used activation functions for MLP classifiers are the sigmoid function, the hyperbolic tangent function, and the rectified linear unit (ReLU) function. The output of the MLP classifier is obtained by computing the outputs of all the neurons in the output layer. The output of each neuron represents the probability of the input belonging to a specific class. The class with the highest probability is then chosen as the predicted class for the input.

In the figure 12, we can see that the classifier performs very well on class 0, with a precision of 0.66, recall of 1.00, and F1-score of 0.80. This means that when the classifier predicts an instance to be in class 0, it is correct 66% of the time, and it correctly identifies all the instances that belong to class 0. On the other hand, the classifier performs even better on class 1, with a precision of 1.00, recall of 0.93, and F1-score of 0.97. This means that when the classifier predicts an instance to be in class 1, it is correct 100% of the time, and it correctly identifies 93% of the instances that belong to class 1. The overall accuracy of the classifier is 0.94, which means that it correctly predicts the class of 94% of the instances. However, since the classes are imbalanced (class 1 has much more instances than class 0), it's important to look at the macro-averaged and weighted-averaged scores as well. The macro-averaged F1-score is 0.88, which is the average of the F1-scores for each class. The weighted-averaged F1-score is 0.95, which takes into account the number of instances in each class. Overall, the classifier performs very well on this dataset, especially on class 1.

5.3.4 Long-Short Term Memory. The LSTM model has an input layer with an LSTM cell, a dropout layer to prevent overfitting, and

```
Classification report:
              precision    recall  f1-score   support

           0       0.66      1.00      0.80    111292
           1       1.00      0.93      0.97    848608

    accuracy                           0.94    959900
   macro avg       0.83      0.97      0.88    959900
weighted avg       0.96      0.94      0.95    959900
```

Figure 12: Radial Basis Function Network

```
Classification report:
              precision    recall  f1-score   support

           0       0.58      0.94      0.72    111431
           1       0.99      0.91      0.95    848466

    accuracy                           0.91    959897
   macro avg       0.78      0.92      0.83    959897
weighted avg       0.94      0.91      0.92    959897
```

Figure 13: Long-Short Term Memory

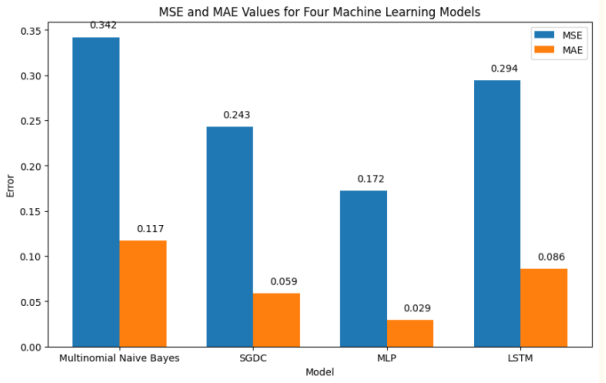

Figure 14: Comparison of MSE and MAE values

an output layer with a sigmoid activation function to produce a binary output. The optimizer used in the model is Adam, which is a popular optimization algorithm for deep learning models. The code selects three features from the merged_df dataframe: char_count, avg_word_length, and answer_count_missing, and stores them in the top_features list. The features are then scaled using scikit-learn's StandardScaler to achieve zero mean and unit variance. The scaled features are stored in the X variable. The target variable answer_count is converted into binary form where 0 indicates an answer is not missing and 1 indicates an answer is missing. The binary target variable is stored in the y variable. The input data is then converted into sequences of a fixed length using a sliding window approach. The length of each sequence is defined by the seq_length variable. The code creates two lists, X_seq and y_seq, to store the input sequences and corresponding target values, respectively. Each input sequence is a two-dimensional numpy array of the shape (seq_length, len(top_features)), where len(top_features) denotes the number of features selected. The target variable is a one-dimensional numpy array with the shape (n_samples,), where n_samples denotes the number of input sequences. The code splits the input sequences and corresponding target values into training and testing sets using the train_test_split function from scikit-learn. An LSTM model is defined using the Keras API. The model consists of an LSTM layer with 64 units, followed by a dropout layer to prevent overfitting, and a dense layer with a single unit and a sigmoid activation function to output binary classification probabilities. The model is compiled using the binary cross-entropy loss function and the Adam optimizer. The model is trained on the training set using the fit method. The training is performed over 5 epochs and in batches of size 64. A validation split of 0.2 is used to monitor the model's performance on a validation set during training. The trained model is evaluated on the testing set using the evaluate method. The testing accuracy is printed to the console. In the figure 13, The model is used to make predictions on the testing set using the predict method. The predictions are thresholded at 0.5 using the np.round function to convert the binary probabilities to binary values. Overall, the macro-averaged F1-score of 0.83 indicates that the model is performing well on average across both classes. The weighted average F1-score of 0.92 takes into account the imbalance between the two classes, and indicates that the model is performing better on the larger negative class.

6 EVALUATION

We assess the aforementioned models in this section and present a comparison for all four applied machine learning methods.

The 14 Bar graph shows the MSE(Mean Squared Error), MAE(Mean absolute error) and their respective model. The Multinomial Naive Bayers has an average of 0.34 and 0.12 values for MSE and MAE repectively. The SGDC model has an average of 0.24 and 0.06 of MSE and MAE respectively. The RBFN model has an average of 0.17 and 0.03 values for MSE and MAE errors respectively. The LSTM model has an average of 0.29 and 0.09 for the error values.

Models	Evaluation		Accuracy
	MSE	MAE	
MultinomialNB	0.342	0.117	88
SGDC	0.243	0.059	97
RBFN	0.172	0.029	94
LSTM	0.294	0.086	91

Figure 15: Model Performance

The 15 table shows the evaluation of models along with errors and their accuracy. It is shown that the Multinomial NB model has an accuracy of 88% and SGDC model shown 97% of accuracy. The RBFN model and LSTM model has approximately around 94% and 91% accuracy respectively.

7 FUTURE WORKS

7.1 Refining Feature Extraction

Further research can be done to explore different feature extraction techniques, such as using word embeddings, topic modeling, or other natural language processing (NLP) methods to extract more meaningful features from the text data. Experimenting with different feature extraction techniques may potentially improve the accuracy of identifying unanswered questions.

7.2 User Feedback and Usability Studies

Collecting user feedback and conducting usability studies can provide valuable insights into the strengths and limitations of the proposed automated system. Future research can involve conducting user surveys, interviews, or usability tests to gather feedback from users who interact with the system. This feedback can help identify any issues, challenges, or suggestions for improvement, which can then inform further iterations of the system.

7.3 Extension to Other QA Datasets as an automated system to unanswered question

the proposed method could be integrated into a larger system that manages large Q&A databases, such as Stack Overflow. The system could use the automated approach to continually scan the database for unanswered questions that require attention, and then alert moderators or other users to provide answers or take action. This would not only enhance database quality by lowering the amount of unanswered questions, but it would also improve user experience by ensuring that inquiries got fast and helpful responses. Additionally, the proposed method could be extended to other Q&A platforms, such as Quora or Reddit, to help manage and improve the quality of those databases as well.

8 CONCLUSION

Our study proposes a recommendation system using machine learning models to identify unanswered questions in a Q&A dataset. By analyzing a Stack Overflow dataset consisting of questions and answers, we can focus on questions without responses from the last 5 years or more. Our approach involves feature extraction on the text data, including character count, word count, average word length, missing answer counts, and the number of tags. We then select the top features using a correlation matrix and train and evaluate a variety of models, including Naive Bayes, SGD, and MLP classifiers, as well as a deep learning model with LSTM layers to capture temporal dependencies. Our results demonstrate that our recommendation system accurately identifies unanswered questions, which can be valuable for managing large Q&A databases by identifying questions that require attention. Further research and refinement of our approach could contribute to the development of automated systems for efficiently managing and supporting Q&A platforms, thereby providing a better user experience and improving the overall quality of information available on such platforms.

9 ACKNOWLEDGEMENT

I want to thank everyone who helped me finish this study and recognize their efforts and contributions. I want to start by sincerely thanking Professor Dr. Muhammad Asaduzzaman, whose advice and assistance were crucial in helping me finish this thesis and carry out our project. I'm also appreciative of my colleagues, who assisted me with the data gathering and analysis by sharing their knowledge and thoughts. I also want to thank Lakehead University for providing the tools needed to finish this paper as well as the Stack Overflow community whose queries and responses served as the foundation for this research.

REFERENCES

[1] Saad Saif Almukaynizi, Majed Alharthi, and Mohammed Al-Kabi. 2021. Automatic Categorization of Unanswered Questions in Stack Overflow. *IEEE Access* 9 (2021), 59798–59807.

[2] Muhammad Asaduzzaman, Ahmed Shah Mashiyat, Chanchal K Roy, and Kevin A Schneider. 2013. Answering questions about unanswered questions of stack overflow. In *2013 10th Working Conference on Mining Software Repositories (MSR)*. IEEE, 97–100.

[3] Antoaneta Baltadzhieva and Grzegorz Chrupała. 2015. Predicting the quality of questions on stackoverflow. In *Proceedings of the international conference recent advances in natural language processing*. 32–40.

[4] Ritu Bisht, Pradeep Kumar Muddapur, Prasenjit Sen, and Rajesh Kumar. 2019. Identifying unanswered questions in stack overflow. In *2019 20th International Conference on Mobile Data Management (MDM)*. IEEE, 246–250.

[5] Alton YK Chua and Snehasish Banerjee. 2015. Answers or no answers: Studying question answerability in stack overflow. *Journal of Information Science* 41, 5 (2015), 720–731.

[6] Jirayus Jiarpakdee, Akinori Ihara, and Ken-ichi Matsumoto. 2016. Understanding question quality through affective aspect in Q&A site. In *Proceedings of the 1st International Workshop on Emotion Awareness in Software Engineering*. 12–17.

[7] Abhishek Kumar, Deep Ghadiyali, Sridhar Chimalakonda, and Akhila Sri Manasa Venigalla. 2023. SOCluster-Towards Answering Unanswered Questions on Stack Overflow via Answered Questions. In *Proceedings of the 16th Innovations in Software Engineering Conference*. 1–5.

[8] Mohammad Masudur Rahman and Chanchal K Roy. 2018. An Insight into the Unresolved Questions at Stack Overflow. *arXiv e-prints* (2018), arXiv–1807.

[9] Saikat Mondal, CM Khaled Saifullah, Avijit Bhattacharjee, Mohammad Masudur Rahman, and Chanchal K Roy. 2021. Early detection and guidelines to improve unanswered questions on stack overflow. In *14th Innovations in software engineering conference (formerly known as India software engineering conference)*. 1–11.

[10] Ngoc Phuoc Nguyen, Hoa Nguyen, and Tien Tuan Nguyen. 2017. Identifying quality issues in stack overflow questions. In *Proceedings of the 11th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. ACM, 198–207.

[11] Ripon K Saha, Avigit K Saha, and Dewayne E Perry. 2013. Toward understanding the causes of unanswered questions in software information sites: a case study of stack overflow. In *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*. 663–666.

[12] Bin Shao and Jiafei Yan. 2017. Recommending answerers for stack overflow with lda model. In *proceedings of the 12th Chinese conference on computer supported cooperative work and social computing*. 80–86.

[13] Zhenhua Tian, Xiangmin Li, Li Li, Fei Wu, Qian Zhang, and Yan Liu. 2020. Collaborative knowledge construction in stack overflow: A replication study. In *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 1934–1943.

[14] Qian Wang, Hua Li, and Wenli Li. 2021. Machine learning-based techniques for question and answer retrieval in community question answering: a systematic review. *Artificial Intelligence Review* 55, 2 (2021), 1361–1388.

[15] Xiang Xia and Liang Zhang. 2019. A Study of Stack Overflow Users and Their Participation Behaviors in the Context of Question Unansweredness. In *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 622–626.

[16] Zhiyong Zhang, Yang Liu, and Hao Wu. 2013. Improving answer quality in community question answering through answerer ranking and selection. In *2013 35th International Conference on Software Engineering (ICSE)*. IEEE, 472–481.