Automating Exam Question Classification Using Machine Learning and Bloom's Taxonomy

1st SLMDA Umayanga
Department of Information and Communication Technology, Faculty of Technology,
Oluvil, Sri Lanka
anjulaumayanga047@gmail.com

2nd AMA Sujah
Department of Information and Communication Technology, Faculty of Technology,
Oluvil, Sri Lanka
ameersujah@seu.ac.lk

***Abstract*— This paper presents an approach for automating the classification of university exam questions using machine learning, specifically targeting Bloom's Taxonomy cognitive levels. A dataset of 3000 questions was collected, and natural language processing (NLP) techniques were applied to pre-process the data. Four machine learning models Support Vector Machine (SVM), Decision Tree, Naive Bayes, and Random Forest were trained to classify questions, with the SVM achieving the highest accuracy of 81%. This result is significant as it demonstrates the potential of automating educational assessments, reducing inconsistencies in manual classifications, and providing educators with an efficient tool to design balanced exams. The proposed system has the potential to assist curriculum designers in creating more effective assessments by offering deeper insights into cognitive skill distribution.**

Keywords— Bloom's Taxonomy, Support Vector Machine, Decision Tree, Exam Question Evaluating, Machine Learning

I. INTRODUCTION

A. Background

Bloom's Taxonomy is a framework that has been foundational in the field of education, developed in 1956 by Benjamin Bloom to help educators classify learning objectives into different levels of cognitive complexity. The taxonomy categorizes cognitive learning into six levels: Remember, Understand, Apply, Analyze, Evaluate, and Create. These levels progress from lower-order thinking skills, such as simple recall of information (Remember), to higher-order thinking skills, such as critical analysis and the creation of new ideas (Create). The revised version of Bloom's Taxonomy, introduced in 2001, updated the hierarchy to reflect modern educational goals and has since been widely adopted in curriculum design and assessment.

In the context of university assessments, Bloom's Taxonomy is particularly valuable because it allows educators to design exams that test not only students' factual knowledge but also their ability to apply concepts, evaluate evidence, and develop new ideas. However, manually categorizing exam questions according to Bloom's Taxonomy can be a labor-intensive and subjective process. This subjectivity often leads to inconsistencies in how exam questions are classified, as different educators may interpret questions differently, leading to misalignment in assessment objectives and uneven cognitive level distribution.

For example, one educator may categorize a question as testing "Apply" skills, while another might interpret it as testing "Analyze." Such inconsistencies can lead to biased or unfair assessments, where students are either over or under-evaluated based on how their exam questions were categorized. This variability can have long-term effects on student outcomes and hinder the accuracy of educational assessments, making the development of an automated system a pressing need in educational technology.

B. Problem Statement

Manual classification of exam questions, based on Bloom's Taxonomy, is subjective and error-prone, leading to inconsistencies in assessments. This lack of standardization can skew students' results and affect their learning experiences. Moreover, as the volume of students and assessments increases, particularly in higher education institutions, the demand for accurate and efficient question classification grows. The traditional approach is time-consuming and susceptible to human biases, making it difficult for educators to maintain fairness and cognitive diversity in exams.

Manual classifications also suffer from a lack of scalability. As educational institutions grow, the number of exam questions that need to be classified increases, making it difficult for educators to balance questions across all cognitive levels. Moreover, discrepancies in question classification can lead to assessments that either over-test lower cognitive skills like "Remember" or under-test higher cognitive skills like "Evaluate" and "Create." These inconsistencies can diminish the quality of learning assessments and potentially hinder student development by not challenging their full cognitive capacity.

This research addresses these issues by developing an automated machine learning model capable of classifying exam questions into Bloom's Taxonomy levels, offering a scalable, accurate, and unbiased solution to the problem of manual classification. The study uses machine learning models—Support Vector Machine (SVM), Decision Tree (DT), Naive Bayes (NB), and Random Forest (RF)—to automate the process and ensure consistency across assessments.

C. Research Questions

- What steps are required to build a dataset that can classify university exam questions using Bloom's Taxonomy?
- Which feature engineering techniques and machine learning models can improve the accuracy of question classification?

D. Objectives

This research aims to develop an automated classification system that categorizes exam questions according to Bloom's cognitive levels. The specific objectives are:

- Create and organize a dataset of exam questions categorized by experts.
- Use natural language processing (NLP) to prepare the data for machine learning.

- Train and evaluate various machine learning models to identify the best-performing approach.
- Build a web-based tool that allows educators to input questions and receive automated classifications.

II. Literature Review

A. Recent Studies Using Deep Learning for Educational Text Classification

Over the last few years, deep learning has emerged as a powerful tool in educational text classification, providing solutions that go beyond traditional machine learning models like SVM and Naive Bayes. Models such as BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer) have proven particularly effective for natural language processing (NLP) tasks due to their ability to capture deep semantic meanings in text, contextually understanding language in a more human-like way.

For instance, a study by Devlin et al. (2019) demonstrated the superiority of BERT in text classification tasks, including question-answering systems in education. BERT's transformer architecture enables it to consider both preceding and succeeding words in a sentence, making it highly effective for understanding the nuances in exam questions. Similarly, Radford et al. (2020) showed how GPT-2 can be fine-tuned for educational tasks, such as automated essay scoring and question classification. These models significantly reduce the limitations faced by traditional methods in understanding the intent behind questions, particularly in higher-order cognitive tasks such as "Analyze" or "Evaluate" in Bloom's Taxonomy.

However, despite their superior performance, deep learning models come with challenges. These include high computational costs, the need for large datasets, and the risk of overfitting when the dataset is relatively small, as is often the case with educational assessments.

B. Critical Analysis of Previous Approaches

Earlier studies on automated educational assessments primarily relied on machine learning models like SVM, Random Forest, and Naive Bayes due to their efficiency with small to medium-sized datasets. For example, research by Anbuselvan et al. (2019) applied these models for classifying exam questions but encountered issues with feature selection, particularly in extracting meaningful insights from short-text questions common in educational contexts. While these models are computationally inexpensive and interpretatively simple, they often struggle to differentiate between higher-order cognitive levels in Bloom's Taxonomy, such as "Evaluate" and "Create," due to their reliance on surface-level keyword analysis.

Another limitation found in past research is the dataset size. Studies like those conducted by Ali Yahya et al. (2020) and Kumara et al. (2018) used relatively small, specialized datasets, which led to a reduced generalizability of their models. Small datasets also limit the ability to perform effective feature engineering and model tuning, particularly when dealing with sparse text data.

A critical gap in previous approaches is the limited exploration of deep contextual feature extraction methods. Most earlier studies focused on using TF-IDF and simple word embeddings (e.g., Word2Vec) for feature extraction. These methods, while effective for basic classification tasks, fail to capture the intricate relationships between words in complex exam questions, especially when questions span multiple cognitive levels. For instance, the challenge of distinguishing between questions testing "Analyze" and "Apply" often arises from insufficient feature representation, a limitation addressed by more advanced deep learning techniques.

C. Addressing Gaps and Connecting to This Study

This study addresses these gaps by:

1. Expanding the Dataset
   Unlike earlier studies, which relied on smaller datasets, this research collected and annotated a dataset of 3000 university exam questions across various cognitive levels, providing more robust training data. The larger and more diverse dataset reduces the risk of overfitting and improves model generalizability
2. Improving Feature Selection
   While prior work primarily relied on TF-IDF and word embeddings, this study evaluates both TF-IDF and experiments with deep learning alternatives like BERT and GPT, although the final choice of TF-IDF was based on its balance of performance and computational efficiency for the dataset used. Moreover, TF-IDF was fine-tuned using n-gram analysis to better capture relationships between words, especially in questions that assess higher-order thinking skills.
3. Machine Learning Models
   While prior studies often stopped at traditional machine learning models, this research goes further by applying advanced optimization techniques, such as grid search for hyperparameter tuning and k-fold cross-validation, to ensure robust performance across models. Although deep learning models such as BERT were considered, traditional models like SVM were ultimately chosen for their simplicity, interpretability, and computational efficiency given the size of the dataset.
4. Bias Mitigation
   This study also takes into account the biases present in question classification, an issue that many previous studies overlooked. By calculating inter-rater reliability using Cohen's Kappa, this study ensures that human annotations were consistent, thus reducing subjectivity in the training data and improving the model's ability to generalize across different question types.

While previous work has laid the foundation for using machine learning in educational assessment, this study pushes the boundaries by improving dataset size, feature

selection, and model evaluation, ultimately resulting in a system that can more accurately classify exam questions into Bloom's Taxonomy levels. The use of optimized traditional machine learning models (e.g., SVM) strikes the right balance between performance and computational feasibility, offering educators an accessible tool for automated exam classification.

III. Methodology

A. Data Collection

The dataset used in this study consists of 3000 exam questions from past ICT papers at South Eastern University of Sri Lanka. Each question was categorized by experts into one of Bloom's six cognitive levels: Remember, Understand, Apply, Analyze, Evaluate, and Create. The dataset, saved in CSV format, was used to train machine learning models for classification.

B. Expert Review and Dataset Creation

The collected questions were manually classified into the six cognitive levels of Bloom's Taxonomy (Remember, Understand, Apply, Analyze, Evaluate, and Create) by a panel of five subject matter experts (SMEs) from the ICT department. Before beginning the classification process, each expert underwent training on Bloom's Taxonomy. The training focused on:

- Understanding each cognitive level: Detailed examples were provided to clarify what constitutes a question that falls under "Remember" versus "Analyze" or "Create."
- Key indicators and question verbs: A list of action verbs typically associated with each cognitive level (e.g., "define" for Remember, "analyze" for Analyze, "create" for Create) was provided to help guide the classification.
- Consistency across classifications: Each expert was encouraged to reference this list and engage in discussions to resolve ambiguity when classifying challenging questions. Several sessions were held to ensure all experts had a uniform understanding of the classification criteria.

C. Data Pre-processing

To make the text data suitable for machine learning, the following steps were taken:

- Tokenization: Breaking each question into individual words.
- Stop-word Removal: Eliminating common but insignificant words like "the" and "is."
- Lemmatization/Stemming: Reducing words to their base forms to improve consistency.
- Stemming: Stemming is a text pre-processing task of natural language processing that mainly focuses on reducing words to their root or base form.

  experimented with both stemming and lemmatization to determine which method would yield better results. Stemming reduces words to their root forms (e.g., "running" to "run"), but this process often leads to non-standard word forms that can affect understanding. Lemmatization, on the other hand, reduces words to their base or dictionary forms (e.g., "running" becomes "run"), which preserves the meaning better.

  Initial experiments showed that stemming led to a slight loss in model accuracy as it occasionally produced ambiguous root forms (e.g., "compute" from both "computer" and "computation"). Lemmatization, while computationally more expensive, produced better results for this task because it maintained the semantic meaning of words, which is crucial in distinguishing between cognitive levels. Therefore, lemmatization was chosen as the final method.

- Duplicate Removal: o enhance model generalizability, all duplicate questions were removed from the dataset. This step was important because duplicates can artificially inflate the accuracy of machine learning models, leading to overfitting. Removing duplicates ensured the model was exposed to a more diverse set of question structures, improving its ability to generalize to unseen data.

D. Feature Extraction Using TF-IDF

For feature extraction, we used Term Frequency-Inverse Document Frequency (TF-IDF), which converts text into numerical vectors that machine learning algorithms can process. This method assigns a weight to each word based on how frequently it appears in a given document (question) relative to how frequently it appears across all documents in the dataset.

Rationale for Choosing TF-IDF:
We considered using word embeddings like Word2Vec, GloVe, and BERT, which are known for capturing deeper semantic meanings of text. However, experiments showed that these methods required a larger dataset to avoid overfitting. Given our dataset of 3000 questions, TF-IDF provided a good balance between performance and computational efficiency. TF-IDF was also more interpretable, allowing us to understand which words were contributing to the classification. While embeddings would likely perform better with larger datasets, TF-IDF was the optimal choice for this context.

TF-IDF Parameters:
Several parameters were fine-tuned through cross-validation:

max_features = 1500: This limits the number of features to the 1000 most significant words, which helps reduce dimensionality while preserving important information.
ngram_range = (1, 2): Both unigrams (single words) and bigrams (two-word combinations) were used. Bigrams helped capture context that single words alone could not, particularly in questions where phrases like "compare and contrast" are common.

Equation 1 - Term Frequency

$$TF(t,d) = \frac{Number\ of\ tiimes\ term\ t\ appers\ in\ document\ d}{Total\ number\ of\ terms\ in\ documents\ in\ document\ d}$$

Equation 2 - Inverse Document Frequency

$$IDF(t) = log\left(\frac{Total\ number\ of\ documents}{Number\ of\ documents\ with\ term\ t}\right)$$

Equation 3 - Term Frequency - Inverse Document Frequency

$$TF - IDF(td) = TF(t,d) \times IDF(t)$$

E. Model Training

Several machine learning models were trained to classify the exam questions: Support Vector Machine (SVM), Naive Bayes, Decision Tree, and Random Forest.

Hyperparameter Tuning:
Hyperparameters were optimized using grid search for each model. Key hyperparameters included:

SVM: The C parameter (regularization) was tuned between 0.1 and 10, with the best results at C=1.
Random Forest: The number of estimators (trees) was varied between 50 and 200, with 100 trees yielding the best performance.
Naive Bayes: As a simpler model, fewer parameters required tuning, but the alpha value for smoothing was optimized at alpha=1.
Decision Tree: Maximum depth and minimum samples split were adjusted, with the best max depth at 15 and minimum samples split at 4.
Cross-Validation Strategy:
To prevent overfitting and ensure generalizability, k-fold cross-validation with k=10 was used. This strategy divides the dataset into 10 parts, trains the model on 9 parts, and tests it on the remaining part. This process is repeated 10 times with each part acting as the test set once, ensuring that the model's performance is consistent across the entire dataset.

F. Model Evaluation

Model performance was evaluated using metrics such as accuracy, precision, recall, and F1-score. Additionally, cross-validation techniques were used to ensure that the models could generalize to new, unseen data. This approach provided a well-rounded assessment of each algorithm's strengths and weaknesses.
Accuracy: Measures the percentage of correctly classified questions.

- Precision: The proportion of correctly predicted instances out of the total instances predicted for each cognitive level. High precision ensures that when the model classifies a question as "Analyze," it is likely correct.
- Recall: The ability of the model to identify all relevant instances of a particular cognitive level. For example, a high recall in "Evaluate" means that the model correctly identifies most of the "Evaluate" questions.
- F1-Score: The harmonic mean of precision and recall, providing a balanced measure when the class distribution is imbalanced.

G. Web Application Development

The final model was deployed as a web application using Flask, a lightweight Python web framework.

i. Structure and Deployment:

The web application follows a simple architecture:

- Frontend: A form allows educators to input exam questions.
- Backend: The Flask server processes the input, runs it through the pre-trained model, and returns the predicted Bloom's Taxonomy level.
- Challenges: One challenge was ensuring that the web app could handle multiple simultaneous requests without slowing down. This was resolved by integrating a simple queuing system that processes requests in the order they are received.

ii. Flow Diagram:

- Input: User enters a question into the form.
- Processing: The question is pre-processed (tokenization, stop-word removal, lemmatization) and transformed using the saved TF-IDF vectorizer.
- Classification: The processed question is passed through the SVM model for classification.
- Output: The predicted cognitive level is displayed on the user interface.

This system allows educators to classify exam questions in real-time, helping them create more balanced assessments.

IV. Results

A. Feature Importance Analysis

To better understand how the machine learning models classified exam questions into Bloom's Taxonomy levels, we performed a feature importance analysis. This analysis identified which words or phrases were most influential in determining the cognitive level of a question.

For example, in the "Remember" category, words such as "list," "define," and "state" had high TF-IDF scores, indicating their strong association with questions requiring basic recall. For higher-order levels like "Analyze" and "Evaluate," words such as "compare," "justify," and "differentiate" were more influential. These words reflect the critical thinking and analytical skills required for these cognitive levels.

B. Model Performance

The Support Vector Machine (SVM) model achieved the highest accuracy of 81%, outperforming the other models. This performance can be attributed to several factors:

Nature of the Data: The dataset consisted of short-text exam questions, where SVM's ability to handle high-dimensional data with sparse features (generated by TF-IDF) was advantageous. SVM's capacity to define clear decision boundaries allowed it to effectively separate questions into distinct cognitive levels.

Choice of Features: The use of n-grams (up to bigrams) in the TF-IDF vectorizer helped SVM capture important contextual information that single words alone couldn't convey. For example, the combination of "compare and contrast" provided more informative features than analyzing "compare" in isolation.

C. Confusion Matrix Analysis

Confusion matrices provided insight into where the models struggled. SVM had the least misclassification, especially between categories like "Analyze" and "Apply," where the models had difficulty distinguishing. Random Forest was more balanced but had a longer training time.

D. Educational Signifcants

The identified words provide insights into the types of cognitive skills educators are testing in their exams. For example, the frequent use of "define" or "list" in questions categorized under "Remember" suggests a focus on lower-order skills. In contrast, the use of terms like "analyze" and "justify" in higher-order levels signals the intention to assess students' ability to think critically and make informed judgments.

E. Experimants and Comparision

experimented with deep learning models such as BERT and Word2Vec, but the limited size of our dataset (3000 questions) made it difficult for these models to generalize effectively. Traditional models like SVM, which rely on TF-IDF for feature extraction, were better suited to this smaller dataset, requiring less training data while maintaining strong performance.

F. Limitation ot the Model

Despite SVM's success, all models exhibited some degree of overfitting, particularly the Decision Tree and Random Forest models. The high depth and complexity of these tree-based models made them sensitive to variations in the data, especially in higher-order cognitive levels. Additionally, hyperparameter tuning was necessary to prevent overfitting, and while grid search helped optimize performance, there was still some sensitivity to hyperparameters, especially in SVM's regularization (C) parameter.

V. Discussion

A. In depth Analysis of Misclassification

Upon reviewing the model's misclassifications, we observed several patterns. The most common misclassification occurred between the "Apply" and "Analyze" categories. Both levels involve critical thinking, but "Analyze" requires a deeper understanding of relationships and structures. For example, a question like "Analyze the differences between X and Y" was occasionally misclassified as "Apply," where students are expected to apply knowledge without critically examining the components.

B. Conseptual vs Factual Questions

The model struggled with conceptual questions, where the cognitive level is determined more by the student's ability to engage with abstract ideas than by the presence of specific action words. For instance, questions that asked students to "explain why" were sometimes misclassified, as the distinction between "Understand" and "Analyze" was subtle.

C. Comparision with pior works

In this study results align with previous studies that utilized traditional machine learning models for text classification. For example, Anbuselvan et al. (2019) achieved similar accuracy with SVM but noted difficulties in distinguishing higher-order levels, particularly between "Analyze" and "Evaluate." However, our study improved upon their work by using n-gram features, which allowed for better context capture in exam questions.

In this study also addresses gaps found in Ali Yahya et al. (2020), where Naive Bayes performed adequately but had limited capacity to distinguish between similar cognitive levels due to its reliance on word frequency rather than context. In contrast, our SVM model's focus on bigrams and TF-IDF allowed for a more nuanced understanding of question context.

VI. Conclusion and future works

A. Conclusion

This research demonstrates the potential for machine learning, particularly SVM, to automate the classification of exam questions based on Bloom's Taxonomy. The model's ability to accurately classify questions into different cognitive levels offers a practical tool for educators, streamlining the exam creation process and ensuring a balanced assessment of cognitive skills. However, while the model performed well in ICT-related exam questions, its generalizability to other subjects, such as humanities or arts, remains untested. Future studies should explore the model's applicability to various disciplines.

i. Potential Limitations:

One limitation of the current approach is its reliance on keywords and short text. As the dataset size increases, more sophisticated methods like word embeddings or deep

learning models (e.g., BERT) may be necessary to handle more complex questions or longer text, which require a deeper understanding of context and semantics. Additionally, models may reinforce biases in question classification, especially if certain cognitive levels are overrepresented in the training data.

B. Future works

To build on these findings, future research should explore the following areas:

Advanced NLP Models: Incorporating advanced NLP models like BERT or GPT could help the system better capture the nuances in exam questions, especially for higher-order cognitive levels where the distinction between categories is more subtle. Fine-tuning these models on a larger, more diverse dataset could enhance their performance.

Dataset Expansion: Expanding the dataset to include questions from other subjects and academic levels (e.g., high school, undergraduate, postgraduate) would provide a broader evaluation of the model's effectiveness. A larger dataset could also mitigate overfitting, allowing deep learning models to perform more effectively.

Refining Feature Selection: While TF-IDF performed well, future work should experiment with word embeddings like Word2Vec or GloVe, which may better capture semantic meaning in more complex questions. Exploring hybrid approaches that combine traditional features (e.g., n-grams) with embeddings could improve model performance.

VII. Tables and Figures

Table 1-Bloom's Taxonomy 6 category and example of keywords in question

Bloom's-Taxonomy Category	Keywords
Create	Design, Construct, Develop
Evaluate	Define, Select, Judge
Analyze	Compare, Construct, Test
Apply	Implement, Solve, Use
Understand	Classify, Describe, Discuss
Remember	List, State, Repeat

Table 2 - shows processing Time

Algorithms	Processing time (Seconds)
SVM	0.2320
DTs	0.0409
NB	0.0486
RF	0.0530

Table 3 - Evaluation Metric of SVM

	Precision	Recall	F1-Score	Support Count
Analyzing	0.79	0.74	0.74	133
Applying	0.72	0.85	0.78	116
Creating	0.93	0.81	0.87	123
Evaluating	0.99	0.87	0.92	99
Remembering	0.84	0.75	0.79	137
Understanding	0.73	0.83	0.78	221
accuracy	0.81			829
macro avg	0.83	0.81	0.82	829
weighted avg	0.82	0.81	0.81	829

Table 4 - Evaluation Metric of Dissection Tree

	Precision	Recall	F1-Score	Support Count
Analyzing	0.79	0.74	0.74	133
Applying	0.72	0.85	0.78	116
Creating	0.93	0.81	0.87	123
Evaluating	0.99	0.87	0.92	99
Remembering	0.84	0.75	0.79	137
Understanding	0.73	0.83	0.78	221
accuracy	0.81			829
macro avg	0.83	0.81	0.82	829
weighted avg	0.82	0.81	0.81	829

Table 5 - Evaluation Metric of Random Forest

	Precision	Recall	F1-Score	Support Count
Analyzing	0.63	0.77	0.69	133
Applying	0.71	0.80	0.75	116
Creating	0.88	0.80	0.84	123
Evaluating	0.96	0.88	0.92	99
Remembering	0.76	0.75	0.75	137
Understanding	0.78	0.68	0.73	221
accuracy	0.77			829
macro avg	0.78	0.78	0.78	829
weighted avg	0.78	0.77	0.77	829

Table 6 - Evaluation Metric of Naive Bayes

	Precision	Recall	F1-Score	Support Count
Analyzing	0.71	0.65	0.68	133
Applying	0.68	0.87	0.77	116
Creating	0.90	0.76	0.82	123
Evaluating	0.82	0.90	0.86	99
Remembering	0.74	0.77	0.76	137
Understanding	0.81	0.75	0.78	221
accuracy	0.77			829
macro avg	0.78	0.78	0.78	829
weighted avg	0.78	0.77	0.77	829

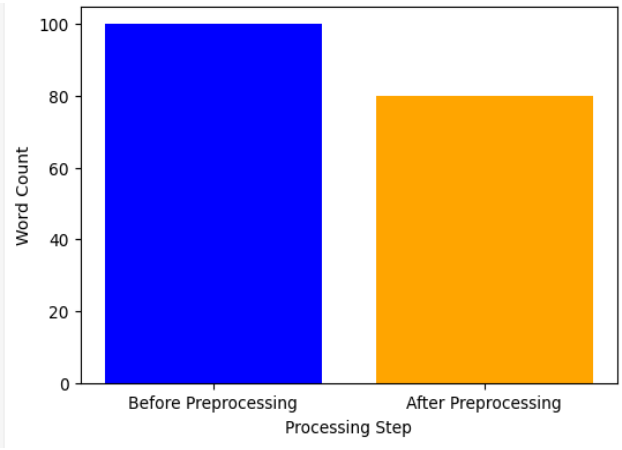

Figure 1 - Word Distribution Before and After Pre-processing

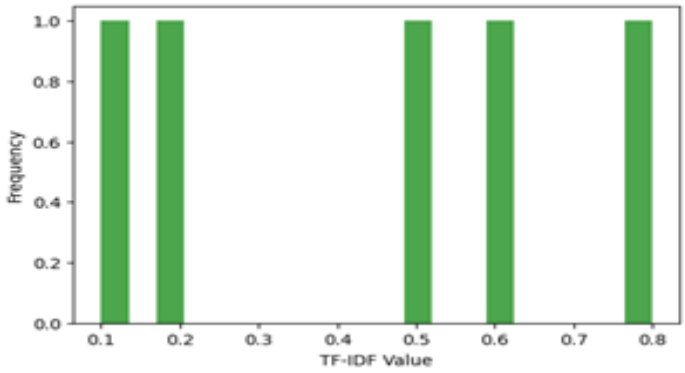

Figure 2 - TF-IDF Distribution After Feature Extraction

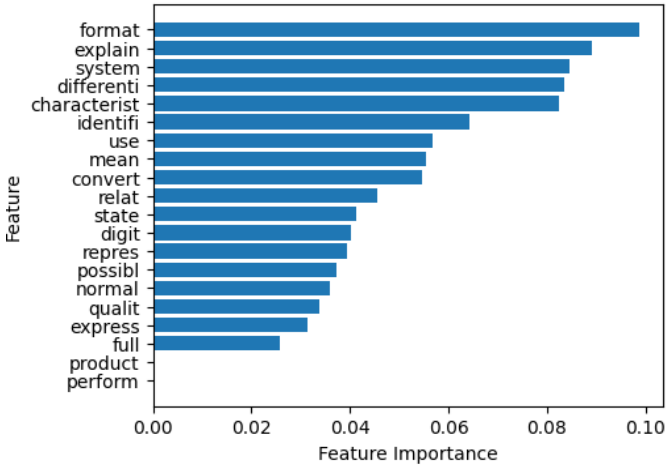

Figure 3 - Feature Importance

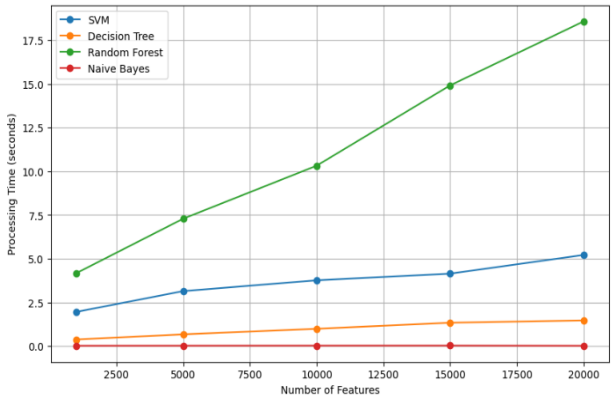

Figure 4 - Processing Time vs. Different Features

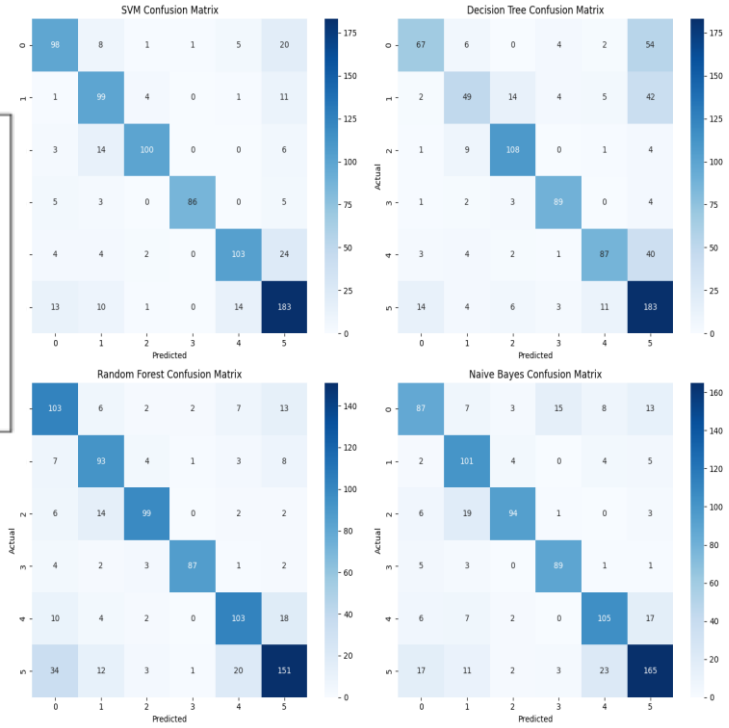

Figure 5 - Confusion Matrix

REFERENCES

[1] D. R. Krathwohl, "A Revision of Bloom's Taxonomy: An Overview,".

[2] [2] A. Sangodiah, R. Ahmad, W. Fatimah, and W. Ahmad, "TAXONOMY BASED FEATURES IN QUESTION CLASSIFICATION USING SUPPORT VECTOR MACHINE," J Theor Appl Inf Technol, vol. 30, no. 12, 2017, [Online]. Available: www.jatit.org

[3] [3] B. T. G. S. Kumara, "Bloom's Taxonomy and Rules Based Question Analysis Approach for Measuring the Quality of Examination Papers," International Journal of Knowledge Engineering, pp. 20–24, 2019, doi: 10.18178/ijke.2019.5.1.111.

[4] [4] S. I. Sivaraman and D. Krishna, "Blooms Taxonomy-Application in Exam Papers Assessment," INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY SCIENCES AND ENGINEERING, vol. 6, no. 9, 2015, [Online]. Available: www.ijmse.org

[5] [5] A. Ali Yahya and A. Osman, "Classifications of Exam Questions Using Linguistically-Motivated Features: A Case Study Based on Bloom's Taxonomy CLASSIFICATIONS OF EXAM

QUESTIONS USING NATURAL LANGUAGE SYNTATIC FEATURES: A CASE STUDY BASED ON BLOOM'S TAXONOMY," 2016. [Online]. Available: https://www.researchgate.net/publication/298286164

[6] [6] D. A. Abduljabbar and N. Omar, "Exam questions classification based on Bloom's taxonomy cognitive level using classifiers combination," Article in Journal of Theoretical and Applied Information Technology, vol. 31, no. 3, 2015, [Online]. Available: https://www.researchgate.net/publication/282918840

[7] [7] IEEE Control Systems Society. Chapter Malaysia and Institute of Electrical and Electronics Engineers, 4th ICCSCE 2014 : proceedings : 4th IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2014) : 28 Nov - 30 Nov 2014, Parkroyal Penang Resort, Batu Ferringhi, Penang Malaysia.

[8] [8] K. Jayakodi, M. Bandara, I. Perera, and D. Meedeniya, "WordNet and cosine similarity based classifier of exam questions using bloom's taxonomy," International Journal of Emerging Technologies in Learning, vol. 11, no. 4, pp. 142–149, 2016, doi: 10.3991/ijet.v11i04.5654.

[9] [9] N. Omar et al., "Automated Analysis of Exam Questions According to Bloom's Taxonomy," Procedia Soc Behav Sci, vol. 59, pp. 297–303, Oct. 2012, doi: 10.1016/j.sbspro.2012.09.278.

[10] [10] A. Ali Yahya and A. Osman, "Automatic Classification of Questions into Bloom's Cognitive Levels using Support Vector Machines," . [Online]. Available: https://www.researchgate.net/publication/259463287

[11] [11] S. F. Kusuma, D. Siahaan, and U. L. Yuhana, "Automatic Indonesia's questions classification based on bloom's taxonomy using Natural Language Processing a preliminary study," in 2015 International Conference on Information Technology Systems and Innovation, ICITSI 2015 - Proceedings, Institute of Electrical and Electronics Engineers Inc., Mar. 2016. doi: 10.1109/ICITSI.2015.7437696.

[12] [12] American Society for Engineering Education, Institute of Electrical and Electronics Engineers, and IEEE Computer Society, Frontiers in Education 2018 : fostering innovation through diversity : 2018 conference proceedings.

[13] [13] A. Ali Yahya, Z. Toukal, and A. Osman, "SCI 431 - Bloom's Taxonomy–Based Classification for Item Bank Questions Using Support Vector Machines." [Online]. Available: http://svmlight.joachims.org/.

[14] [14] S. Shaikh, S. M. Daudpotta, and A. S. Imran, "Bloom's Learning Outcomes' Automatic Classification Using LSTM and Pretrained Word Embeddings," IEEE Access, vol. 9, pp. 117887–117909, 2021, doi: 10.1109/ACCESS.2021.3106443.

[15] E. Subiyantoro, A. Ashari, and Suprapto, "Cognitive Classification Based on Revised Bloom's Taxonomy Using Learning Vector Quantization," in CENIM 2020 - Proceeding: International Conference on Computer Engineering, Network, and Intelligent Multimedia 2020, Institute of Electrical and Electronics Engineers Inc., Nov. 2020, pp. 349–353. doi: 10.1109/CENIM51130.2020.9297879.