EXAM QUESTIONS CLASSIFICATION BASED ON BLOOM'S TAXONOMY: A HYBRID APPROACH USING TRADITIONAL MACHINE LEARNING AND NEURAL NETWORKS

[1]Mr. Shreyas N. Chandurkar
Department of Information Technology
BRACT Vishwakarma Institute of Information Technology Pune, India
shreyas.22120264@viit.ac.in

[2]Mr. Chetan G. Ingle
Department of Information Technology
BRACT Vishwakarma Institute of Information Technology Pune, India
chetan.22010416@viit.ac.in

[3]Mr. Akash G. Gulge
Department of Information Technology
BRACT Vishwakarma Institute of Information Technology Pune, India
akash.22010608@viit.ac.in

[4]Prof. Chaitali Shewale
Prof. Department of Information Technology
BRACT Vishwakarma Institute of Information Technology Pune, India
chaitali.shewale@viit.ac.in

***Abstract* – This study presents a groundbreaking solution to the arduous task of classifying exam questions according to Bloom's Taxonomy, a cornerstone of educational assessment. Manual classification, prone to subjectivity and time constraints, is revolutionized through the introduction of an innovative hybrid approach, amalgamating Neural Network and traditional machine learning techniques. The primary objective is to automate question categorization, furnishing educators with a more efficient and impartial means to evaluate students' cognitive abilities. Leveraging machine learning augments the precision and expediency of this categorization process.**

The hybrid model intricately intertwines the interpretability of traditional machine learning models with the nuanced pattern recognition capabilities of Neural Networks. Methodologically, the model is trained on a meticulously labeled dataset of exam questions, harnessing features extracted from traditional machine learning algorithms such as K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Naive Bayes (NB), and Random Forest. Notably, the Random Forest algorithm demonstrates superior accuracy among the ensemble.

The implications of this hybrid model reverberate across diverse educational domains. Its applications span from automated grading systems to personalized learning platforms and detailed performance analytics. Within educational technology, the model's integration promises advancements in adaptive learning environments, question generation tools, and tailored feedback mechanisms. Furthermore, its utility extends to research and development endeavors, facilitating tasks such as educational data mining, the formulation of novel assessment methodologies, and the scrutiny of cognitive demands within educational materials.

In essence, this hybrid approach transcends the limitations of manual classification, heralding a new era in educational technology and research methodologies. It furnishes educators with a standardized and efficacious tool, fostering a deeper comprehension of students' cognitive progression. The model's automation capabilities streamline assessment procedures, furnishing a valuable resource for educators and researchers alike, thereby revolutionizing the educational landscape.

I. INTRODUCTION

1.1 What is Blooms Taxonomy?:

Bloom's Taxonomy is a hierarchical framework that classifies educational objectives and cognitive skills into a structured hierarchy. It was developed by educational psychologist Benjamin Bloom in the 1950s and later revised by Anderson and Krathwohl in 2001. The taxonomy organizes cognitive skills into six levels, ranging from lower-order thinking skills to higher-order thinking skills:

1. **Remembering**: Recalling facts, information, or concepts.
2. **Understanding:** Explaining ideas or concepts and demonstrating comprehension.
3. **Applying:** Using information or knowledge in a new situation or context.
4. **Analyzing**: Breaking down information into components, identifying patterns, and understanding relationships.
5. **Evaluating:** Making judgments based on criteria and standards.

6. Creating: Generating new ideas, products, or ways of viewing things.

Bloom's Taxonomy provides a framework for educators to design learning objectives and assessments that target specific cognitive skills, aiding in the evaluation and improvement of students' understanding and application of knowledge.

1.2 Aim or Purpose of Project:

The aim of the project is to create a system that automates the classification of exam questions based on Bloom's Taxonomy. This automation aims to bring efficiency and objectivity to the grading process, mitigating the time-consuming nature of manual classification and reducing the influence of personal opinions. The overarching goal is to enhance the accuracy of question classification using a combination of traditional machine learning methods and Neural Networks.

1.3 Brief Introduction of Project:

The project focuses on leveraging technology, specifically machine learning, to automate the classification of exam questions according to Bloom's Taxonomy. By combining traditional machine learning models and Neural Networks, the system aims to provide a reliable and precise classification mechanism. This approach addresses the challenges associated with manual grading, such as time constraints and subjective biases. The project not only seeks to streamline the grading process but also aligns with broader educational goals, contributing to the improvement of educational technology and research methods.

1.4 How Proposed System Will Help Users:

The proposed system offers several benefits to users in the education domain:

- **Efficient Grading:** The system enables automated grading, saving time for educators and facilitating a quicker turnaround for assessments.

- **Consistency and Objectivity:** By utilizing machine learning, the system reduces personal biases in question classification, ensuring more consistent and objective grading across different assessors.

- **Personalized Learning Experiences:** The model's accurate classification can contribute to creating personalized learning experiences for students, aligning assessments with their cognitive abilities and learning styles.

- **Performance Analysis**: The system provides detailed performance analyses, offering insights into students' strengths and weaknesses, which can inform targeted interventions and improvements in teaching strategies.

- **Educational Technology Enhancement**: In the realm of educational technology, the system supports adaptive learning tools, improves question creation processes, and offers personalized feedback, enhancing the overall learning experience.

- **Contribution to Research and Development:** Beyond immediate applications, the system's insights can contribute to advancements in educational data analysis, the development of new assessment methods, and a deeper understanding of cognitive aspects related to educational content.

II. LITERATURE SURVEY

[1] Paper Named “Classifying Question Papers with Bloom’s Taxonomy Using Machine Learning Techniques” Published in year 2019. The authors named Minni Jain, Rohit Beniwal, Aheli Ghosh, Tanish Grover, and Utkarsh Tyagi employed a methodology that involved using various Machine Learning (ML) techniques to classify question papers into different levels of Bloom’s taxonomy. They collected a dataset of 1024 questions from three universities and developed a web app to evaluate their approach. The research approach included the use of nine different ML techniques, such as Logistic Regression, Linear Discriminant Analysis, K-Nearest Neighbors, Random Forest, Decision Trees, Support Vector Machine (SVM), Neural Network, and others, to classify the question papers. The implementation details included feature extraction, ML techniques used, and experimental setup. The authors also discussed related work, the work flow, coding approach, and the experimental setup in the paper. The results showed that the best accuracy of 83.3% was achieved with Logistic Regression and Linear Discriminant Analysis ML techniques. Additionally, the authors suggested using Nature-Inspired algorithms to further improve accuracies.

[2] Paper Named “Question classification based on Bloom's taxonomy cognitive domain using modified TF-IDF and word2vec” Published in year 2020. The authors named Manal Mohammed and Nazlia Omar. Manal Mohammed interpret their findings as demonstrating the effectiveness of their proposed method for classifying exam questions based on Bloom’s taxonomy cognitive domain. They emphasize that their model, which utilizes modified TF-IDF and word2vec features, achieved significant results in classifying questions from multiple domains based on Bloom’s taxonomy. The study used two datasets, one with 141 questions and the other with 600 questions, and evaluated the classification results using three different classifiers: K-Nearest Neighbour, Logistic Regression, and Support Vector Machine. The findings showed that the proposed method outperformed traditional TF-IDF, achieving weighted F1-measures ranging from 71.1% to 89.7% for the different classifiers and datasets. Additionally, the authors highlighted the potential applications of their model in educational settings, such as aiding educators and lecturers in analyzing exam questions to fulfill the requirements for different levels of education. They also suggested potential applications in automatic test generation systems, intelligent tutoring systems, and serious games. Furthermore, the authors noted that further study could improve, enhance, or extend their work by expanding the

dataset to include different types of questions and providing a public benchmark dataset for question classification based on Bloom's taxonomy.

[3] Paper Named "Analysis of Bloom Taxonomy-Based Examination Data Using Data Mining" published in year 2023. The authors named Dr Amit Kumar, Dr. Dinesh Singh, Dr. M. S. Dhankhar interpret their findings by presenting a comprehensive analysis of the research conducted on the application of data mining techniques to the assessment of student test scores, particularly in the context of utilizing Bloom's Taxonomy for question classification. They utilize various machine learning algorithms, such as Bi LSTM and CNN, for feature extraction and model training and testing. The authors compare the performance of their proposed model with other cutting-edge models, emphasizing metrics such as recall, precision, and F1-score. They also provide a detailed evaluation of the outcomes, including the accuracy, precision, recall, and confusion matrix of the proposed model. Additionally, the authors present the results of their research in a tabular format, outlining the comparison criteria applied to the outcomes and the mathematical equations used for evaluation. They also discuss the implications of their findings, highlighting the effectiveness of the proposed model in predicting question difficulty levels and its potential for enhancing the assessment of students' comprehension and learning progress. Overall, the authors interpret their findings by demonstrating the superiority of their proposed model and its contribution to the field of automated assessment systems.

[4] Paper Named "Exam Questions Classification Based on Bloom's Taxonomy: Approaches and Techniques" published in year 2020. The authors named Monika Yogendra Gaikwad, Vaishnavi Ravindra Kute, Gayatri Subhash Pawar, Mrunal Manikchand Magare interpret their findings by proposing an autonomous question paper-generation system that utilizes Bloom's Taxonomy and machine learning to assign marks to questions and generate question papers. They emphasize the significance of leveraging Bloom's Taxonomy for question categorization and mark assignment, providing a structured approach to streamline the process of question paper generation. The proposed system architecture, algorithm, and technology for question paper generation are outlined, with a focus on Bloom's Taxonomy for categorizing questions, text pre-processing, rule development, and randomization algorithms. The authors conclude by highlighting the potential of their proposed system to accurately assign marks to questions and generate multiple question papers using Bloom's Taxonomy and machine learning. Additionally, the document provides a thorough review of existing literature on exam question classification based on Bloom's Taxonomy and related techniques, offering insights into the research landscape and the evolution of automated question paper generation systems.

[5] Paper Named 'Classifying Exam Questions Based On Bloom's Taxonomy Using Machine Learning Approach" published in year 2019. The authors named Nidaa Ghalib and Dhiyaa Salih Hammad of the document utilized a machine learning approach to classify exam questions based on Bloom's Taxonomy. The methodology involved several phases, including data set planning and compiling, tokenizing and stop words elimination, feature selection, classification, and evaluation. They employed supervised machine learning approaches, specifically Naïve Bayes (NB) and K-Nearest Neighbour (KNN) classifiers, to classify exam questions. The data set used for training and testing was obtained from a programming questions bank for the Bachelor of Information Technology Year 1 to 3, as well as questions from various websites related to Bloom's taxonomy studies in programming subjects. Additionally, the authors experimented with different feature selection methods, such as Mutual Information, Odd Ratio, and Chi-Square, to improve the classification performance of the machine learning models. The results indicated that feature selection methods positively contributed to the performance of the classifiers. Overall, the authors' methodology involved the application of machine learning techniques and feature selection methods to automate the classification of exam questions based on Bloom's Taxonomy.

[6] The paper "Exam questions classification based on Bloom's taxonomy cognitive level using classifiers combination" published in year 2015 by authors Dhuha Abdulhadi Abduljabbar and Nazlia Omar contributes new knowledge by proposing a combination framework that efficiently addresses the problem of cognitive category determination for programming questions and achieves satisfactory results. The study introduces a novel method that utilizes a combination strategy based on a voting algorithm, integrating three machine learning classifiers: Support Vector Machine (SVM), Naïve Bayes (NB), and k-Nearest Neighbour (k-NN). This approach aims to classify exam questions automatically according to the cognitive levels of Bloom's taxonomy, addressing the challenge of classifying short text questions. The paper also explores the use of feature selection methods, such as Mutual Information, Odd Ratio, and Chi-Square, to enhance the classification process. Additionally, the study evaluates the performance of the proposed combination model and discusses the potential for further experimentation with other techniques to obtain enhanced results. Overall, the paper contributes to the advancement of automated question classification systems and provides insights into the effective integration of different feature selection methods and classification algorithms to achieve more accurate question classification.

Table No. 1: - Literature Surveys

Sr	**Ref No**	**Survey**
1	**1**	In this research, a range of algorithms was examined, including Logistic Regression, Linear Discriminant Analysis, K-Nearest Neighbors, Random Forest, Decision Trees, Support Vector Machine (SVM), Neural Network, and others. Notably, both Linear Discriminant Analysis (LDA) and Logistic Regression (LR) demonstrated an accuracy of 83.3%. These algorithms were evaluated for their ability to classify exam questions according to Bloom's Taxonomy levels. The

		results indicate that LDA and LR show potential for precise classification in educational assessment scenarios.
2	**2**	In this paper the authors found their method effective in classifying exam questions based on Bloom's taxonomy. Using modified TF-IDF and word2vec features, they tested their model on two datasets of 141 and 600 questions, respectively. Comparing with traditional methods, their model achieved better results, with F1-measures ranging from 71.1% to 89.7% using classifiers like K-Nearest Neighbor, Logistic Regression, and Support Vector Machine. They highlighted its potential in aiding educators analyze exam questions and suggested applications in test generation, tutoring systems, and serious games. They also suggested expanding the dataset for further improvement and providing a public benchmark dataset.
3	**3**	In this paper the study examines how data mining techniques, like BiLSTM and CNN, assess student test scores using Bloom's Taxonomy. They compare their model's performance with others, focusing on metrics like recall and precision. Results show their model, especially BiLSTM, achieves 82% accuracy in predicting question difficulty levels, promising improved assessment of student learning. The authors provide detailed analyses, including confusion matrices, and present findings in tables. This research highlights the potential of their model in automated assessment systems, affirming its superiority..
4	**4**	In this paper the authors review how they used Bloom's Taxonomy and machine learning to create a system that makes exam papers on its own. They explain how important it is to use Bloom's Taxonomy to organize questions and assign grades, making it easier to make exam papers. They describe the system they made, including how it works and the technology it uses. They end by saying their system can accurately grade questions and make lots of different exam papers. They also talk about other research on making exam papers using Bloom's Taxonomy and similar methods.
5	**5**	In this paper the authors used machine learning to sort exam questions by Bloom's Taxonomy. They planned, gathered, and processed the data, then trained classifiers like Naïve Bayes and K-Nearest Neighbour. They got their questions from IT-related sources and tested various methods to pick the best features for classification. Results showed that choosing the right features improved classification accuracy, with KNN performing the best. In summary, the authors used machine learning and feature selection to automate sorting exam questions based on Bloom's Taxonomy, finding KNN to be the most effective classifier.
6	**6**	In this paper the paper introduces a new method to automatically classify programming exam questions based on their difficulty levels. It combines three types of machine learning tools: Support Vector Machine (SVM), Naïve Bayes (NB), and k-Nearest Neighbour (k-NN), using a voting system. This method helps solve the challenge of classifying short text questions. The study also tests different ways to choose the most important information for classification. Overall, it improves how we classify exam questions and suggests more experiments to make it even better.

III.PROPOSED METHODOLOGY

The overall approach is illustrated in the following flow chart:

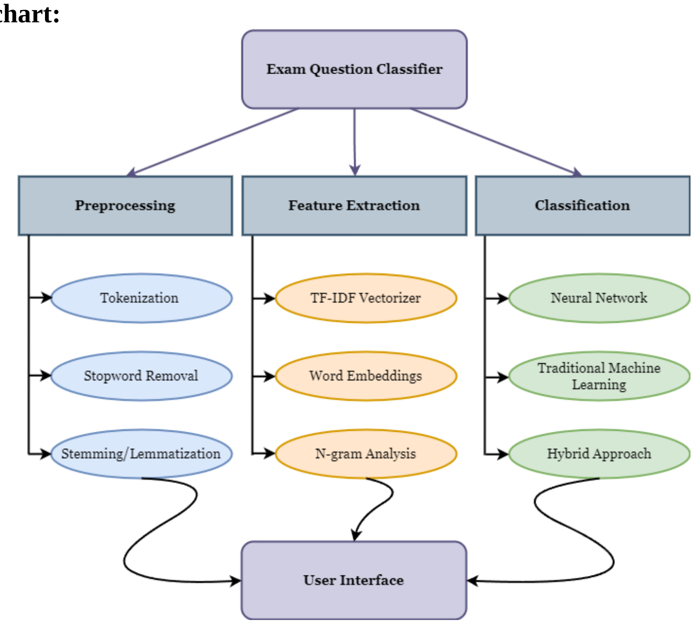


Fig.1 Methodology Flowchart

3.1 Data gathering:

To conduct this research, a dataset comprising student information from previous years was obtained from Google. The dataset included over 4000+ Questions.

3.2 Pre-processing:

This is the stage where raw data is prepared for use in a machine learning model. In the context of exam questions, this might involve breaking the questions down into separate words or phrases (tokenization) and removing common words (stop words) that don't provide much meaning.

3.2.1 Tokenization:

This breaks text down into individual words or phrases, called tokens..

3.2.2 Stopword Removal:

Stopword removal cleans text by filtering out common words like "the" or "and" that carry little meaning. This focuses analysis on important words for tasks like summarizing text or building search engines..

3.2.3 Stemming/Lemmatization:

Stemming and lemmatization are text tricks for finding word roots. Stemming chops off ends (playing -> play) but might create nonsense (loved -> lov). Lemmatization uses dictionaries to find the actual base word (running -> run).

3.3 Feature Extraction:

This involves identifying and extracting the most important features from the data. In the case of exam questions, this might involve things like the keywords used in the question, the length of the question, or the part of speech used in the question. Here we have used the following 3 feature selection methods.

3.3.1 TF-IDF Vectorizer:

The Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer is a popular technique for converting text data into a numerical representation suitable for machine learning algorithms.

Functionality:

TF-IDF vectorizer transforms text data into a sparse matrix, where each row represents a document and each column represents a unique word in the corpus (collection of documents).
The value in each cell reflects the weight assigned to a particular word within a specific document.

Core Components:
Term Frequency (TF):

TF calculates the frequency of a word appearing within a single document. It's often a raw count normalized by the total number of words in the document.
Formula:

TF(t, d) = frequency of term t in document d / total number of words in document d
Inverse Document Frequency (IDF):

IDF captures how important a word is to a specific document by considering its frequency across the entire corpus. Words that appear frequently across all documents have a lower IDF weight, while words that are specific to a few documents have a higher weight.

Formula (common logarithm):

IDF(t) = log (N / df(t))
where:

N is the total number of documents in the corpus
df(t) is the number of documents containing term t
TF-IDF Weight:

The TF-IDF weight for a word (t) in a document (d) is the product of TF and IDF:
Formula:

TF-IDF(t, d) = TF(t, d) * IDF(t)

3.3.2 Word Embeddings:

This is a technique used to represent words as vectors. These vectors can then be used to capture the relationships between words.

3.3.3 N-gram Analysis:

This involves analyzing sequences of n words. For example, bigrams would be sequences of two words, while trigrams would be sequences of three words.

3.4 Classification:

3.4.1 Neural Networks:

3.4.1.1 BERT:-

BERT: A Powerful Approach to Natural Language Processing

Standing for Bidirectional Encoder Representations from Transformers, BERT is a game-changer in the field of NLP. It capitalizes on the Transformer architecture, particularly its encoder component, to generate contextual embeddings for words within text. Unlike traditional models that process text sequentially, BERT analyzes the entire sentence at once, taking into account both preceding and following words (bidirectional analysis). This empowers BERT to capture intricate relationships between words and grasp the subtleties of language.

Key Functionalities of BERT

BERT's effectiveness stems from two primary pre-training tasks:

Masked Language Modeling (MLM): Words in a sentence are randomly masked, and BERT predicts the most likely original word based on the surrounding context. This strengthens BERT's ability to understand how words interact with each other.
Next Sentence Prediction (NSP): Given two sentences, BERT predicts if the second sentence logically follows the first. This helps BERT grasp the

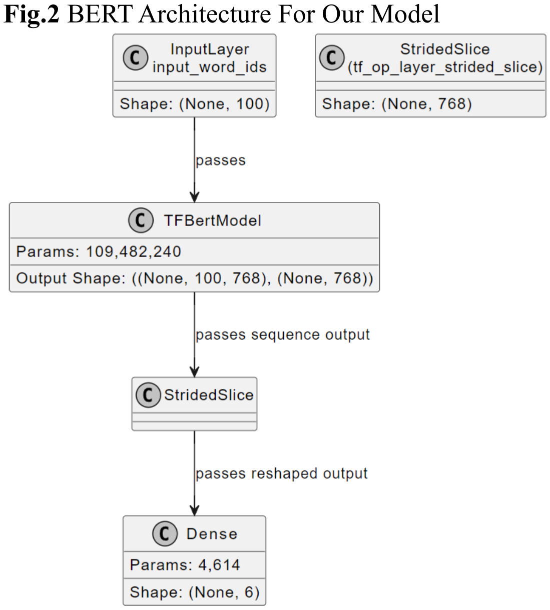
coherence between sentences and how ideas are connected.

BERT's Advantages

BERT's proficiency in understanding context makes it excel in various NLP tasks, including:

Question Answering: BERT can accurately locate answers to questions posed about a given passage.
Text Summarization: It can effectively condense lengthy text while retaining crucial information.
Sentiment Analysis: BERT can determine the sentiment (positive, negative, or neutral) expressed within text.
A key strength of BERT is its versatility. The pre-trained model can be fine-tuned for specific NLP tasks, making it adaptable and efficient.

Fig.2 BERT Architecture For Our Model



3.4.2 Traditional Machine Learning

3.4.2.1 KNN Algorithm:

Functionality:
KNN classifies new data points based on the similarity to their closest neighbors in the training data.

Core Steps:
Data Representation: Each data point is represented as a feature vector, containing numerical attributes relevant to the task. Imagine disease diagnosis, where features might be symptom frequencies.
Distance Metric: A distance metric, like Euclidean distance, calculates the "closeness" between two data points based on their feature vectors.
K-Nearest Neighbors: Given a new, unlabeled data point, KNN identifies the K closest data points (neighbors) from the training data using the chosen distance metric.
Prediction:
Classification: For classification tasks (like diagnosing a disease), KNN predicts the new point's class by looking at the most frequent class among its K nearest neighbors (majority vote).
Regression: For regression tasks (like predicting house prices), KNN predicts a value for the new point by averaging the values of its K nearest neighbors.

Mathematical Considerations :
While there's no single formula for KNN, a common distance metric is the Euclidean distance:

d(x1, x2) = sqrt(sum((xi1 - xi2)^2))

where:
d(x1, x2) is the distance between data points x1 and x2
xi1 and xi2 are corresponding features in the feature vectors of x1 and x2

3.4.2.2 SVM Algorithm

Functionality:
SVM aims to identify an optimal hyperplane in a high-dimensional space that effectively separates data points belonging to different classes.

Core Principles:
Data Representation: Each data point is represented as a feature vector, containing numerical attributes relevant to the classification task. Imagine classifying emails as spam or not-spam based on features like word frequencies.

Hyperplane: An SVM seeks a hyperplane that maximizes the margin, the distance between the hyperplane and the closest data points from each class (known as support vectors).

Kernel Functions: For non-linearly separable data, SVMs employ kernel functions to implicitly map the data into a higher-dimensional space where a linear separation becomes possible.

Classification: New data points are classified by predicting on which side of the hyperplane they fall in the high-dimensional space (as determined by the kernel function).

Mathematical Considerations (Optional):

While the specific formulation of SVM involves optimization techniques, the concept of maximizing the margin can be intuitively understood.

3.4.2.3 NB Algorithm

Functionality:

NB classifies data points based on applying Bayes' theorem and assuming conditional independence between features.

Core Assumptions:

Conditional Independence: NB assumes that the features used for classification are conditionally independent of one another given the class label. While this assumption might not always hold true in reality, NB often performs well in practice.

Classification Process:

Data Representation: Data points are represented as feature vectors, often containing word counts or other relevant attributes for text classification tasks.
Bayes' Theorem: NB utilizes Bayes' theorem to calculate the posterior probability (probability of a class given a data point) for each possible class.

Key Equation:

P(Class C | Data D) = (P(C) * P(D | C)) / P(D))

where:
P(Class C | Data D): Posterior probability (probability of class C given data D)
P(C): Prior probability of class C (independent of data)
P(D | C): Probability of data D given class C
P(D): Total probability of data D (often a constant for a given data point)
Class Prediction: The class with the highest posterior probability is assigned to the new data point.

3.4.2.4 RF Algorithm

Functionality:

RF builds a collection of decision trees at training time, with each tree predicting a class or value for a new data point.
The final prediction is based on the majority vote (classification) or average (regression) of the individual tree predictions.

Core Steps:

Bootstrapping: RF creates multiple bootstrapped samples (random samples with replacement) from the training data.
Decision Tree Induction: For each bootstrap sample, a decision tree is grown. Each tree node splits the data based on a random subset of features (instead of considering all features at each split).
Prediction: When presented with a new data point, each tree in the forest makes a prediction.
Classification: The final class prediction is the most frequent class among the individual tree predictions (majority vote).
Regression: The final predicted value is the average of the individual tree predictions.

3.4.3 Hybrid Approach

Hybrid fusion in the realm of machine learning (ML) involves the strategic amalgamation of traditional techniques with modern ML approaches. This blending aims to capitalize on the unique advantages of both methods to create ML models that are more resilient, interpretable, and proficient. Essentially, hybrid fusion recognizes that while traditional methods offer interpretability and often rely on expert domain knowledge, ML techniques excel at discerning intricate patterns within data and making precise predictions.

This integration of traditional methodologies and ML approaches can manifest in diverse ways, each contributing to the overall efficacy of the hybrid model:

1. Incorporation of Expertise: Traditional methods frequently draw upon expert knowledge or domain-specific rules to address problems. In hybrid fusion, this expertise seamlessly integrates into the ML process. Expert insights may influence feature selection, model architecture, or decision-making criteria, thereby enriching the model's accuracy and relevance.

2. Feature Engineering: Traditional methodologies often entail manual feature engineering, guided by domain knowledge to select and craft pertinent features. In hybrid fusion, these engineered features complement those acquired automatically by ML algorithms. This synergy enables the model to capture both explicit domain knowledge and subtle data patterns, thereby enhancing performance.

3. Interpretability: Many traditional methods, such as decision trees or linear models, inherently offer interpretability. In hybrid fusion, interpretable models coalesce with more complex ML models, such as deep neural networks. By doing so, researchers uphold transparency and comprehensibility while harnessing the predictive power of advanced ML techniques.

4. Ensemble Techniques: Hybrid fusion frequently involves ensemble methods, amalgamating traditional methods with ML models to bolster overall performance. Ensembling amalgamates predictions

from multiple models to yield a more accurate and robust final prediction. For instance, merging a decision tree classifier with a gradient boosting algorithm often yields superior predictive accuracy compared to standalone methods.

5. Data Augmentation and Preprocessing: Traditional methodologies may grapple with unstructured or intricate data. In contrast, ML approaches thrive in learning from vast datasets. In hybrid fusion, ML techniques can augment data, preprocess it, or clean it, enhancing the quality of input data for traditional methods and fortifying overall model performance.

6. Adaptive Learning: Traditional methods may confront challenges when confronted with evolving data distributions or dynamic environments. Hybrid fusion addresses this by integrating ML components that continually learn from incoming data. This adaptive learning capability ensures that the model remains pertinent and effective over time, even amidst changing data dynamics.

By integrating traditional methodologies with ML approaches through hybrid fusion, researchers can craft ML models that not only harness the strengths of each approach but also mitigate their respective limitations. This approach paves the way for innovative applications across various domains, from healthcare and finance to natural language processing and beyond.

IV. RESULTS AND DISCUSSION

Here the algorithms used by us for classification of prediction are KNN, SVM, NB and RF. The classification and comparison of algorithms is done on the basis of Accuracy, Recall, F1, Precision and Support. From which we concluded that the Accuracy of Random Forest is highest in traditional machine learning.

1. Precision:
Precision measures the proportion of correctly predicted positive cases out of all the cases predicted as positive.

Formula: Precision = True Positives / (True Positives + False Positives)

A high precision indicates that the model is good at not making false positive predictions. For example, in a spam email classification task, high precision means the model identifies most emails flagged as spam as actual spam.

2. Recall:
Recall measures the proportion of correctly predicted positive cases out of all the actual positive cases.

Recall = True Positives / (True Positives + False Negatives)

A high recall indicates that the model is good at identifying most of the actual positive cases. Continuing the spam email example, high recall means the model catches most actual spam emails.

.

3. F1-Score:
F1-Score is a harmonic mean between precision and recall, combining their influence into a single metric.

Formula: F1-Score = 2 * (Precision * Recall) / (Precision + Recall)

A high F1-Score indicates a good balance between precision and recall. It's a useful metric when both precision and recall are important.

.

4. Support:
Support refers to the total number of actual cases in a particular class (positive or negative) in the data.

Formula: No specific formula, it's the raw count.

Support provides context for precision and recall by indicating the number of data points involved in each class.

5. Accuracy:
Accuracy is the overall proportion of correctly classified cases (both positive and negative).

Formula: Accuracy = (True Positives + True Negatives) / Total Cases

Accuracy is a simple metric, but it can be misleading if the data has imbalanced classes.

Table.2 Performance Analysis On the basis of F1, Recall, Support, precision and Accuracy. By using TF-IDF Feature Extraction Technique

Sr. no	**Ref no**	**Blooms Taxonomy Level**	**Precision**	**Recall**	**F1 Score**	**Support**
1	**SVM**	1	1.00	1.00	1.00	35
		2	0.91	0.80	0.85	25
		3	0.94	0.97	0.96	33
		4	0.88	0.93	0.90	30
		5	1.00	1.00	1.00	34
		6	1.00	1.00	1.00	28
		Accuracy			**0.96**	**185**
		Macro Avg	**0.95**	**0.95**	**0.95**	**185**
		Weighted Avg	**0.96**	**0.96**	**0.96**	**185**
2	**NB**	1	0.97	0.97	0.97	35
		2	0.67	0.72	0.69	25

		3	0.88	0.64	0.74	33
		4	0.72	0.87	0.79	30
		5	0.93	0.79	0.86	34
		6	0.82	1.00	0.90	28
		Accuracy			**0.83**	**185**
		Macro Avg	**0.83**	**0.83**	**0.82**	**185**
		Weighted Avg	**0.84**	**0.83**	**0.83**	**185**
3	**KNN**	1	0.20	1.00	0.34	35
		2	0.00	0.00	0.00	25
		3	0.91	0.30	0.45	33
		4	0.00	0.00	0.00	30
		5	1.00	0.09	0.16	34
		6	0.00	0.00	0.00	28
		Accuracy			**0.26**	**185**
		Macro Avg	**0.35**	**0.23**	**0.16**	**185**
		Weighted Avg	**0.38**	**0.26**	**0.18**	**185**
4	**RF**	1	1.00	1.00	1.00	35
		2	1.00	0.88	0.94	25
		3	0.92	1.00	0.96	33
		4	0.97	0.93	0.95	30
		5	1.00	1.00	1.00	34
		6	0.97	1.00	0.98	28
		Accuracy			**0.97**	**185**
		Macro Avg	**0.97**	**0.97**	**0.97**	**185**
		Weighted Avg	**0.97**	**0.97**	**0.97**	**185**

In Traditional Method by using TF-IDF Feature Extraction Technique We Got **97% Accuracy** of **Random Forest** Algorithm.

Fig.3 Accuracy Comparison Of Models Using TF-IDF

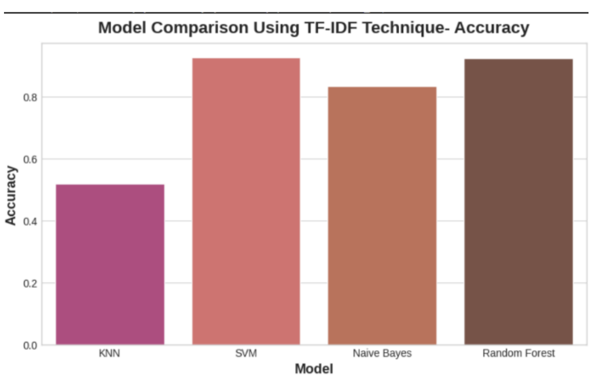


Fig.4 Comparison between model on the basis of evaluation metrices using TF-IDF

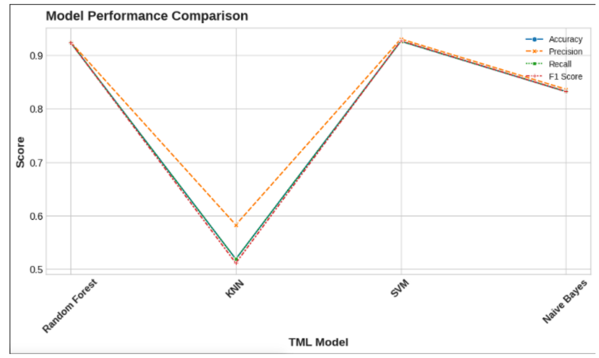


Table.3 Performance Analysis On the basis of F1, Recall, Support, precision and Accuracy. By using Word-Embeddings and N-gram Feature Extraction Technique

Sr. no	**Ref no**	**Blooms Taxonomy Level**	**Precision**	**Recall**	**F1 Score**	**Support**
1	**SVM**	1	0.00	0.00	0.00	53
		2	0.00	0.00	0.00	74
		3	0.00	0.00	0.00	53
		4	0.16	1.00	0.27	58
		5	0.00	0.00	0.00	59
		6	0.00	0.00	0.00	72
		Accuracy			**0.16**	**369**
		Macro Avg	**0.03**	**0.17**	**0.05**	**369**
		Weighted Avg	**0.02**	**0.16**	**0.04**	**369**
2	**NB**	1	0.97	0.68	0.80	53
		2	0.87	0.55	0.68	74
		3	0.73	0.68	0.71	53
		4	0.49	0.90	0.63	58
		5	0.85	0.59	0.70	59
		6	0.69	0.85	0.76	72
		Accuracy			**0.71**	**369**
		Macro Avg	**0.77**	**0.71**	**0.71**	**369**
		Weighted Avg	**0.77**	**0.71**	**0.71**	**369**
3	**KNN**	1	0.95	0.74	0.83	53
		2	0.80	0.53	0.63	74
		3	0.62	0.55	0.58	53
		4	0.42	0.53	0.47	58
		5	0.44	0.71	0.55	59
		6	0.67	0.58	0.62	72
		Accuracy			**0.60**	**369**
		Macro Avg	**0.65**	**0.61**	**0.61**	**369**
		Weighted Avg	**0.65**	**0.60**	**0.61**	**369**
4	**RF**	1	0.84	0.79	0.82	53
		2	0.57	0.61	0.59	74
		3	0.63	0.58	0.61	53

		4	0.45	0.62	0.52	58
		5	0.72	0.47	0.57	59
		6	0.51	0.51	0.51	72
		Accuracy			**0.59**	**369**
		Macro Avg	**0.62**	**0.60**	**0.60**	**369**
		Weighted Avg	**0.61**	**0.59**	**0.60**	**369**

In Traditional Method by using Word-Embeddings and N-gram Feature Extraction Technique We Got **71% Accuracy** of **Naïve Bayes** Algorithm

Fig.5 Accuracy Comparison Of Models Using Word Embeddings and N-gram.

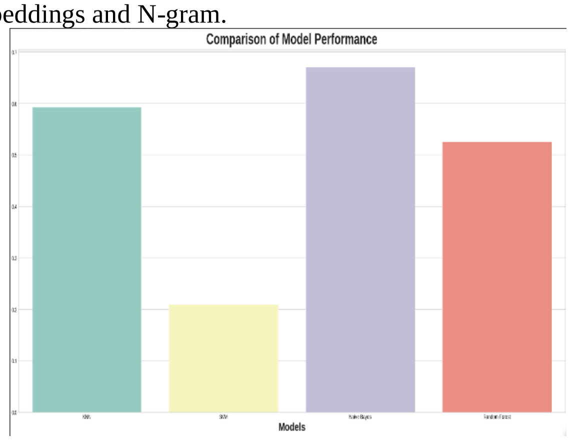
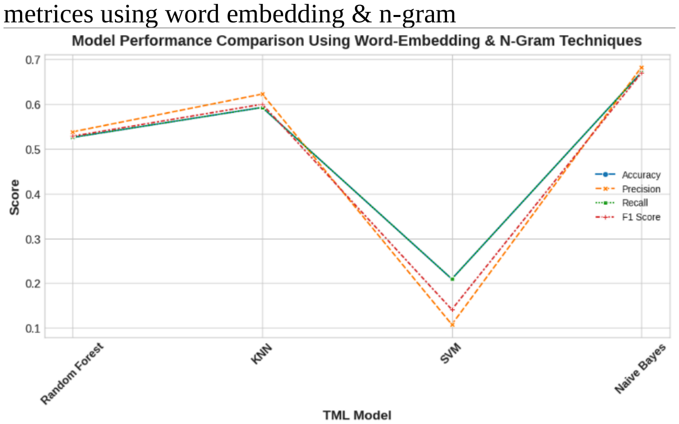


Fig.6 Comparison between model on the basis of evaluation metrices using word embedding & n-gram



BERT Model Results

Fig.7 Bert Model Results

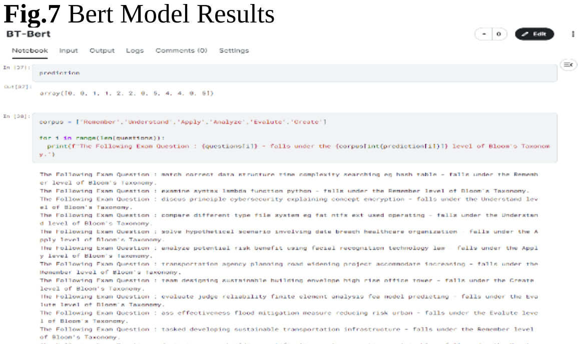


Fig.8 Bert Model Results

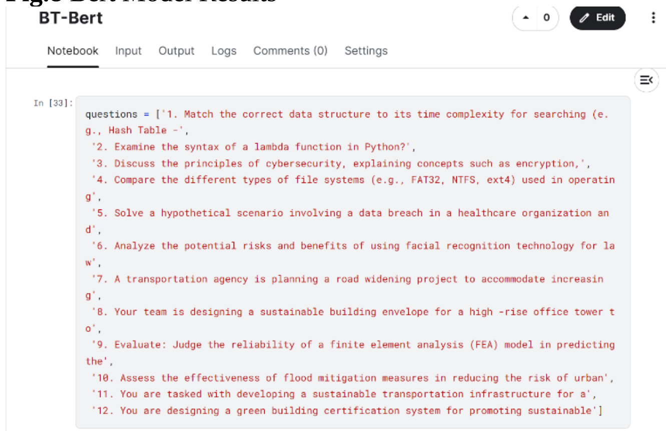


Fig.9 Extracted Questions From Input PDF

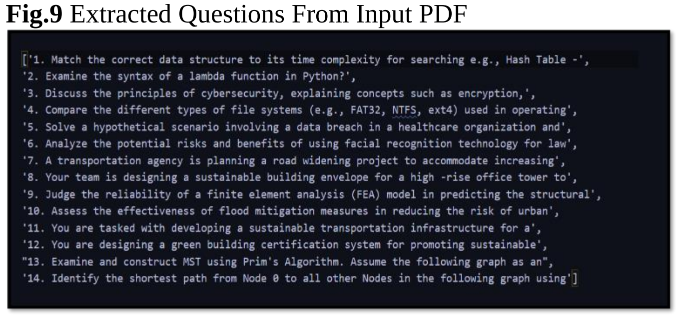
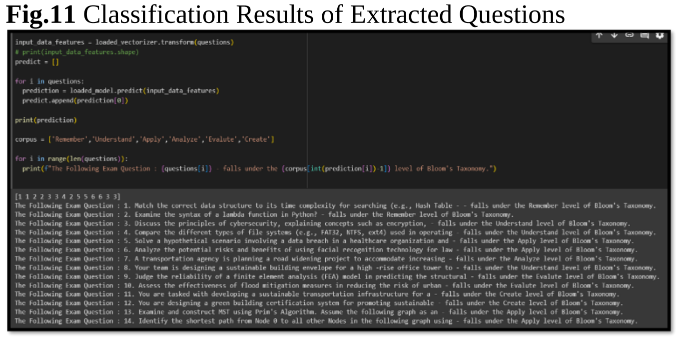


Fig.10 Data Before and After Preprocess

Fig.11 Classification Results of Extracted Questions



V. CONCLUSION

This study investigated the efficacy of various machine learning algorithms for classifying exam papers according to specific criteria. Our findings demonstrate that the Random Forest (RF) algorithm using TF-IDF Feature Extraction Technique (In Traditional Approach) achieved the highest accuracy in this task (Bloom's Evaluate). This suggests that RF's ability to handle complex data structures and reduce overfitting makes it well-suited for the often nuanced characteristics of exam paper content (Bloom's Analyze)..

VI. FUTURE SCOPE

The use of Bloom's Taxonomy for exam paper classification offers a promising foundation for future advancements. Here are some exciting possibilities to explore:

1. Automatic Feedback and Learning Path Recommendations:

Classifying exam questions based on Bloom's Taxonomy allows for automatic generation of targeted feedback for students. Imagine a system that identifies students struggling with "application" level questions and recommends relevant practice problems or learning resources.

This personalized feedback could highlight areas for improvement and suggest targeted learning paths to strengthen a student's understanding.

2. Adaptive Learning Systems:

By analyzing past exam papers and student performance, the system could predict the appropriate cognitive level for future assessments.

This would enable the creation of adaptive learning systems that adjust the difficulty level of questions presented to students based on their individual strengths and weaknesses as identified through Bloom's Taxonomy classification.

3. Educational Data Analytics and Insights:

Analyzing exam papers classified by Bloom's Taxonomy can provide valuable insights into learning outcomes and curriculum effectiveness.

Educators could identify areas where students struggle with specific cognitive levels and adjust their teaching strategies accordingly.

This data-driven approach can inform curriculum development and ensure it caters to the development of various cognitive skills outlined in Bloom's Taxonomy.

4. Multi-label Classification and Skill Identification:

Current approaches might focus on classifying a question based on its dominant cognitive level. Future advancements could involve multi-label classification, where a question can be assigned multiple Bloom's levels if it targets various cognitive skills.

This would provide a more nuanced understanding of the skills assessed by each question.

5. Integration with Learning Management Systems:

Integrating Bloom's Taxonomy classification with Learning Management Systems (LMS) could create a seamless learning environment.

The LMS could automatically categorize learning materials and assessments based on Bloom's Taxonomy, allowing students to target specific skill development through relevant resources.

Challenges and Considerations:

Developing robust algorithms for accurate classification remains a challenge, especially for open-ended or complex questions.

Defining clear boundaries between different Bloom's levels can be subjective.

Integrating these advancements with existing educational infrastructure requires careful planning and implementation.

Overall, leveraging Bloom's Taxonomy for exam paper classification holds immense potential to personalize learning, improve assessment practices, and ultimately enhance educational outcomes.

REFERENCES

[1]https://link.springer.com/chapter/10.1007/978-981-13-9942-8_38

[2]https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7081997/

[3]https://ijisae.org/index.php/IJISAE/article/view/3860

[4]https://www.researchgate.net/publication/347254153_Exam_Questions_Classification_Based_on_Bloom%27s_Taxonomy_Approaches_and_Techniques

[5]https://www.researchgate.net/publication/339775553_CLASSIFYING_EXAM_QUESTIONS_BASED_ON_BLOOM%27S_TAXONOMY_USING_MACHINE_LEARNING_APPROACH

[6]https://www.jatit.org/volumes/Vol78No3/15Vol78No3.pdf