Learning Style Detection and Content Modifier with Artificial Intelligence

Md. Kabin Hasan Kanchon
Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
kabin.kanchon@northsouth.edu

Kaniz Fatema Nabila
Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
1912688@northsouth.edu

Mahir Sadman
Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
mahir.sadman@northsouth.edu

Ramisa Tarannum
Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
ramisa.tarannum@northsouth.edu

Riasat Khan
Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
riasat.khan@northsouth.edu

***Abstract*—Customized learning material can boost individuals' learning capacity in this fast-forwarding EdTech era. This project aims to develop a suitable way to identify a learner's preferred learning style and modify the learning content that fits the style using Artificial Intelligence. We found that activity classification through learners' web tracking logs and feedback classification from an individual's response are the two most efficient techniques for determining a learner's learning style. We analyzed different clustering methods for activity classification where K-means clustering gives our desired number of clusters to categorize by user's actions. For feedback classification, we applied a few text classification models to classify learners into one of three primary learning styles: Visual, Auditory, and Kinesthetic, and DistilBERT achieved the highest accuracy of 100%. We used these two classification techniques on our specially created data set of about 14506 samples to categorize students into their respective learning styles. Decision tree, Random Forest, and KNN provided the maximum accuracy of 100% and F1 score of 100% for the custom dataset. Classification methods, text summarization, knowledge graphs, and question-answer techniques have been used for content modification. To accommodate the learning styles, we modify the content in various ways, including color-code, audio content, mind-maps, flashcards, etc.**

***Index Terms*—learning style, DistilBERT, k-means clustering, text summarization, knowledge graph**

I. Introduction

Learning is a complex and multifaceted process, and understanding individual differences in how people learn is critical for effective education. Learning style refers to the preferred manner of processing information and acquiring knowledge of an individual's ability [1]. It involves understanding how a person learns best, which could be through visual, auditory, read/write, or kinesthetic (hands-on) methods. Learning style theory suggests that people have different learning preferences, and understanding these preferences can help enhance the effectiveness of teaching and learning [29]. Visual learners prefer visual aids such as images, charts, and videos to learn. Auditory learners benefit by listening to lectures and discussions. Hands-on activities and movement, such as experiments and role-playing, help kinesthetic learners to know the best. Different studies say the ratio of visual, auditory, and kinesthetic learners is 65:30:5 [6]. We are motivated to perform this work because we are interested in how each student's adaptation capability and learning outcome to various content and study material differs. Therefore, we wish to detect an individual's learning style and align the learning materials with their learning style. It will make learning enjoyable, improve their engagement with the knowledgeable content, and motivate them to gather knowledge in their preferred learning style. Hence, we aim to detect one's learning style and help them optimize their learning experiences and perform better. This project aims to create a learning style detection and content modification system with the following contribution:

- A novel dataset of Learning Style Detection has been introduced, which contains 14506 samples with 22 attributes. The dataset has been collected with the help of North South University Students, which required them to take part in a course through MOODLE (a learning management platform) where we track and analyze various factors of the learners' interaction, their duration of completion of the course, assessment of multiple types of contents, survey, and feedback to detect their learning styles. We have used the six machine learning algorithms: decision tree, random forest, Support Vector Machine (SVM), Logistic regression, Gaussian Naive Bayes, and K-nearest neighbor (KNN).
- For the content modification, we worked on multiple modification processes on the learning materials, which include color-code, audio content, mind maps, flashcards, etc. The methods we used for content modification are classification method, text summarization, knowledge graph, and question-answer techniques with the help of multiple models and libraries such as SpaCy, Rapid Au-

tomatic Keyword Extraction(RAKE), Relation Extraction By End-to-end Language generation (REBEL), Text-to-Text Transfer Transformer (T5), Bidirectional Encoder Representations from Transformers (Bert), Text-to-Text Transfer Transformer base (T5 base) and Generative Pretrained Transformer 3 (GPT -3).

- Our work aims to identify learning styles and modify contents based on particular learning styles. Through this, we aim to enhance personalized learning experiences by accurately detecting learning styles and modifying content to cater to individual learners' needs.

This work combines personalized and public datasets for precise learning style detection and customizes learning materials to fit the particular learning style. While learner-type detection and content modification exist separately, we brought them into an integrated system with specialized features like mind maps and flashcards, enhancing personalized learning experiences significantly.

After introducing our topic, we present the related works in Section II and discuss the proposed system in Section III, where we talk through the datasets and models used for the project. Section IV is dedicated to the significant results of our project, and section V has the limitation of the project. Finally, Section VI concludes the paper with some directions for the future improvement of this work.

II. Related Works

Learning style detection has long been a topic of interest in education research as educators strive to create personalized and compelling learning experiences for their students. Detecting learning styles and modifying contents according to their preference is not easy as there are various methods to determine how individuals learn. We're checking out past studies to see how the experts have used to detect learning styles. By doing this, we can better understand where our project fits in, what's already out there, and where we can make our project stand out. Several researchers have contributed to this field. Some of them are worth discussing.

Fareeha and Wahid [21] detected learning styles through machine learning approaches. The authors employed the Felder Silverman Learning Style Model (FSLSM) for their experiment and used 16 combinations of learning style dimensions. Firstly, they used a methodology to check the consistency of their proposed model by equating their predicted value with a questionnaire model result. After reviewing the consistency of the model, they cleaned all the data preprocessing and started training the model with a clean dataset consisting of 498 samples. They used various ML algorithms, where the Support Vector Machine (SVM) algorithm gave the most accuracy, 75.55%, with the closest precision and recall values, 73% and 76%, respectively.

Aissaoui et al. [5] proposed an approach for detecting learning styles using machine learning algorithms to implement adaptive e-learning systems (AeS). There are two aspects to the suggested process. Using an unsupervised method (Kmeans) based on FSLSM, the first stage involves collecting information about the learners from the log file and clustering them. Using a supervised algorithm, the Naive Bayes tree (NBtree), the second part estimates a new learner's preferred learning method. The authors used a real dataset to perform this research and evaluated the performance using the confusion matrix. The overall outcome of the strategy produced an accuracy rate of 89.06%. The outcomes coming from the approach showed that the process has outstanding results.

Ananthu and his teammates [12] applied the Visual Auditory Kinesthetic (VAK) model and focused on identifying multiple learning styles parallelly. They chose the VAK model and focused on identifying various learning styles parallelly. A survey consisting of fifteen multiple-choice questions was conducted to gather data from 1123 students. They used the k-means algorithm to cluster the dataset, concluding with 51 clusters. With 92.87% accuracy, they employed the SVM algorithm to predict learning styles. They then combined the predictions using the decision tree algorithm with inputs from cluster values, individual learning styles, and combinations using the Rand index approach.

Wibirama et al. [28] researched the strengths and limitations of three main approaches to distinguishing learning styles: the conventional approach, the artificial intelligence-based approach, and the sensor-based approach. They used web-based learning programs, object-oriented modeling lectures for Bayesian network and NBTree algorithms, an accelerometer, and an eye-tracker to measure different approaches to learning style detection. Furthermore, they introduced a theoretical way to detect learning styles utilizing SVM classification algorithms.

Vaishnav [26] studied three popular learning styles, visual, auditory, and kinesthetic, and their impact on students' academic performances. The author implemented the Howard Gardner VAK learning style brainbox and the VAK Learning Style Inventory by Victoria Chislett and Alan Chapman processes to identify the learning style. A sample of 200 students was used to conduct the study. Analysis of the sample shows that the percentage of visual, auditory, and kinesthetic learners is 33.50%, 28.50%, and 38%, respectively. The study also suggests different learning methods for VAK learners. For visual learners, the author recommends watching videos, drawing maps, taking notes, and using highlights, for auditory learners, it is suggested that they take part in group discussions, record lectures, and listen to videos; and for kinesthetic learners, they are recommended to study in brief sessions and attend lab classes.

Thushara et al. [25] conducted a comparative study on keyword and keyphrase extraction algorithms without using the corpus. The authors implemented TextRank, RAKE, and PositionRank algorithms for the study. Whereas the RANK algorithm includes candidate and keyword selection, TextRank and PositionRank include Graph-building, sorting, and forming keyphrases. Based on unsupervised techniques, the study did not require a specific training dataset. The authors experimented with the algorithms on research documents. The result analysis shows that PositionRank gives the most

relevant keyphrases within less time than TextRank and Rake. TextRank, which takes more time for large documents, the keyphrases extracted from RAKE are quite long and less relevant.

Naseer et al. [17] discussed different NER techniques and the advantages and disadvantages of various libraries and compared them based on their performances. The NER techniques they discussed include rule-based, learning-based, and hybrid approaches. Among the libraries and NER tools, SpaCy, StanfordNLP, TensorFlow, and Apache OpenNLP are the subject of the discussion. The dataset for training and testing all these tools is the same. The NER tools show that SpaCy has the highest F1 score, followed by TensorFlow, OpenNLP, and StanfordNLP. SpaCy provides the best result with 100% training accuracy, F1 score, and prediction probability and with the least training loss compared to all the tools.

Ranganathan and Abuka [20] proposed a text summarization technique using the Text-to-Text Transfer Transformer (T 5) model. To perform the theoretical text summarization, they fine-tuned the T5 model. The authors used the University of California, Irvine's (UCI) drug reviews dataset and BBC news dataset, a standard dataset previously used for the same purpose. The authors measured the efficacy of the model using the ROUGE metrics. The model received an average ROUGE1, ROUGE2, and ROUGEL scores of 45.62, 25.58, and 36.53, respectively, for the UCI dataset, which upon using the BBC News dataset, obtained an average ROUGE1, ROUGE2, and ROUGEL scores of 69.05, 59.70, and 52.97, respectively.

Andrei Popescu-Belis [19] nd his team proposed a novel solution to generate synthetic data based on a rhyming algorithm that helps learn to rhyme. For this, they used the GPT2 model and successfully obtained 60% of rhyming lines of (AABB) consecutive rhymes. However, it failed to work with alternative verses (ABAB) as it has longer-range dependencies.

Xiaojun Chen [3] and his team represented the knowledge graph, which works according to knowledge reasoning. Knowledge reasoning is divided into three categories: rule-based reasoning, distributed representation, and neural knowledge-based reasoning. There are many applications of knowledge graphs, and knowledge reasoning based on recurrent neural networks was one of them.

The features to identify learning styles in the existing popular solutions are expensive (eye movement) and time-consuming (Survey). Our solution, on the other hand, can be used with all the regular standard LMSs. No extra device or dedicated time will be required for the detection. Though some content converters exist independently, central systems exist for only some models or conversions. In our approach, we get all helpful content for each style under a hood.

III. Proposed System

A. Methodology

Our work is divided into two major parts, as shown in Fig.1. The description is given below :

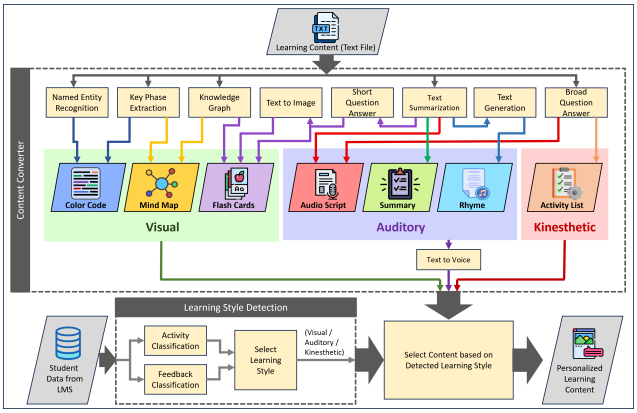

Fig. 1. Methodology

1) Learning Style Detection: To identify a student's learning style, we input the student's data from the Learning Management System and pass those to the identifying methods. We have two identifying ways.

- Activity Classification: In this method, we try to identify the student's style by his/her activities in the Learning Management System such as average video-watching time, audio listing time, quiz scores, lesson completion time, etc. For activity classification, we did not find any suitable dataset. So, we decided to create a custom dataset. We analyzed some clustering methods with lookalike public datasets to evaluate the dataset's usability and selected the most appropriate clustering method. Using that clustering method, we got three clusters in our custom dataset. It indicates that we can work with the custom dataset to identify the style with classification methods. Finally, we analyzed some classification methods and selected the one that gave us the best result.
- Feedback Classification: In this method, we try to identify the student's style by their preferred words in conversation. We take the student's feedback in text form as input. We built a text classification model using a public dataset. This model can identify a student's style based on their comment or feedback.

2) Content Modifier: We take a PDF of text as learning content and convert it into various learning contents for three different learning styles.

- Visual: for visual learners, we are creating three contents
  - Color Code: Here, combining the Name Entity Recognition technique and Key Phase Extraction technique, we are identifying the words, which are persons' names, locations, year, time, etc. These words will be colored according to the type so that a virtual learner can easily notice the necessary information.
  - Mind Map: Mind maps are helpful to visual learners to understand a topic better. We use Key Phase Extraction and Knowledge Graph techniques to extract the key points and the relations between those.

  - Flash Cards: Using Knowledge Graph, Text to Image, and Question Answering techniques, we enlist the most essential information and represent that. It helps visual learners to memorize better.
- Auditory: For Auditory learners, we are creating three contents
  - Audio Script: Auditory learners learn best by listening, so we are creating a script for the audio lesson. The lesson is designed with some questions and mentions what to expect in this lesson at the beginning. After that, the lesson will continue following the primary learning content. We take advantage of text summarization and question-generation techniques.
  - Summary: After an audio lesson, an overview of the lesson is best for this type of learner to take out the key points. We use the Text Summarization technique to get an abstract summary of the learning content.
  - Rhyme: Rhymes are the best ways for auditory learners to memorize something. We generate short rhymes containing the learning content's key points using the Text Generation technique and appropriate prompts.

  We convert these text contents into audio content using the text-to-speech technique, as audio is the best way for auditory learners to learn.
- Kinesthetic: For auditory learners, we are creating one content.
  - Activity List: As kinesthetic learners learn best by doing activities, we create a list of questions from the learning content. The students will try to get the answers by themselves using different sources like books or the internet. After that, the lesson will begin. For this, we are again utilizing the question-generation technique.

  Finally, we pass the selected converted contents to the student according to the detected learning style.

B. Dataset

We have used two public datasets and a customized dataset in our project. The first dataset, named Mooc Dataset, was collected from Kaggle. It is an unlabeled dataset with 9 features and 29165540 samples.

The other dataset we have used is the Learning style (VAK) dataset, which was also collected from Kaggle. This dataset consists of 15451 samples with two columns: sentence and type. The dataset has three classes: Visual, Auditory, and Kinesthetic. It is a relatively balanced dataset. I.

TABLE I
NUMBER OF SAMPLES IN EACH CLASS OF VAK DATASET

Class	Number of Sentences
Visual	5,827
Kinesthetic	4,819
Auditory	4,804

We gathered about 14506 samples for our customized dataset by introducing a course to some students through Moodle. This course included videos, audio, and text, and students engaged with the content by taking quizzes, participating in a survey, and providing feedback. Moodle allowed us to track time, assess marks, and feedback. Treating this information as data, we discern the learner types. The students' survey responses are the ground truth for identifying their learning styles. From the dataset, we found that 65% of the students who took the course were visual learners, 30% were auditory learners, and 5% were kinesthetic learners, as shown in Fig.2.

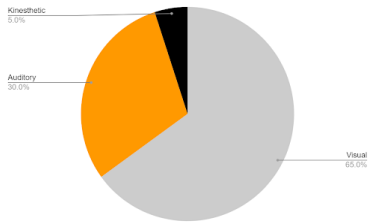

Fig. 2. Percentage of classes in Custom Dataset

C. Dataset Preprocessing

1) Activity Classification: Some preprocessing has to be done before fitting a dataset into a model. For our MOOC dataset, we initially selected 50,000 data for two models and 5,000 randomly for the other two models. We ensured that the visualization of our dataset was similar to the previous one. The PC crashed because our dataset was too huge, so we had to take a small portion of the samples to run the models.

We used some models that need to be revised with categorical values. So, we converted all the categorical values into numerical ones with one-hot encoding and label encoding. The 'action' feature was converted using one-hot encoding, and the other features were converted using label encoding. We converted one of our features, "Datetime" in date format, to the "time diff" of float type format column using the 'Pandas' operation.

The dataset has some null values; every sample was significant for our model. So, we filled the null values to 0. This dataset has no duplicate samples. For feature selection, we generated a heatmap with a correlation matrix. The features that have values of more than 0.7 were dropped. Only one feature, 'enroll id,' had a correlation value of 1.0. There were some outliers in our dataset. For this, we did boxplotting. Also, we need to scale all our features to do Principal Component Analysis (PCA). We mounted our dataset with a min-max scalar technique. Min-max scalar is a prevalent scalar function nowadays, which can be expressed as:

$$X_{new} = \frac{X_i - min(X)}{max(X) - min(X)} \tag{1}$$

where in (1), X_{new} is the scaled value. X_i is the current value, $min(X)$ is the minimum value, and $max(X)$ indicates

the maximum value of the feature. This scalar gives a value in the range (0,1).

The dataset had a huge dimension with 28 features. So, we used PCA to convert the higher dimension into a lower dimension with two columns.

2) Feedback Classification: For our VAK dataset, we first checked the number of samples and if any null value was present. Then, we went through some preprocessing of the datasets, starting with label encoding. We converted our categorical column "Types" into numerical values. Three classes, Auditory, Kinesthetic, and Visual, were thus transformed into numeric values 0, 1, and 2, respectively.

The samples of sentences in our dataset contain many characters alongside English words. These characters can be any punctuation mark or unique character but are not impactful. So, to cleanse our dataset, we removed all the unimportant characters.

For contextual and sentimental analysis of the words, we further did lemmatization. Lemmatization was done to organize each word's inflected forms into a single unit.

We split our dataset into train, validation, and test data to train our model. We did stratification to select the samples in training, validation, and test data proportionately. After stratification, the ratio in train, validation, and test data for Auditory, Kinesthesis, and Visual are 24:24:29, 6:6:7, and 24:25:30, respectively.

We also did tokenization to transform each sentence and phrase into understandable chunks of data. With tokenization, our Natural Language Processing (NLP) is initiated.

3) Customized dataset preprocessing: There were four sections of our data. Each section has an Excel file of grades, participation, activity logs, event completion, and types, which is the ground truth. We took the activity logs file and mapped each file according to the "User full name" column of activity logs. Then, finally, we concatenated all four sections in a final data frame. Our final data frame has a shape of 14506 samples and 22 columns. We checked for null values and duplicated rows. We dropped all the rows that had any null values or were duplicated. A " Time " column had time values for each event a student finished or started. First, we defined the "Time" column as the "Datetime" data type. Then, we sorted our data frame by "User full name" and "Time." Then, we created another column, "Time difference," for the time differences between two consecutive events done by a student. For this, we used the "diff()" method on the "Time" column and grouped the data frame by "User full name". We put 0 for the first difference in column "Time difference." Then, we converted the categorical columns into numerical columns using the Label Encoder. We checked if the dataset was balanced or not and also checked if there were any outliers. We found some outliers, and the dataset was quite imbalanced. So, we scaled the columns that had outliers by using the "MinMaxScaler()" and used the "SMOTE" to balance the data. Before these, we checked correlations of the columns with threshold 0.7 and dropped unnecessary features; we also split the data frame into train and test parts using the ratio of 0.2. Now, our data is ready to go through machine learning models.

4) Content preprocessing: Before converting the raw text file input, we did the necessary preprocessing. The preprocessing included tokenization, where we split the texts into smaller units. We also did lemmatization to determine the base root form to categorize words with similar meanings and lower the text to convert the text into the same case. Finally, we performed stopped word removal as these words contain less than zero values. We also removed the unnecessary symbols and replaced extra spaces with single spaces using the regular expression library of Python.

D. Machine Learning Models

1) Activity Classification: For the MOOC dataset, we used K-means clustering, Density-Based Spatial Clustering of Application with Noise (DBSCAN), Mean-shift, and Affinity Propagation algorithm for clustering. The equations we used while clustering the MOOC dataset were Min-max Scalar and Silhouette Score. We used Decision tree, Random Forest, SVM, Logistic regression, and KNN for the customized dataset. We used Min-max Scalar, Accuracy, and F1 Score for clustering our customized dataset.

- K-means clustering: It is a prevalent and most known clustering method [24]. It divides the data into several clusters and then refers the points to the nearest clusters. Then, it plots the centers of each cluster and calculates the Euclidean distance from each end to the centers of the clusters. Then, it again plots the centers after calculating the Euclidean distances and continues until it makes optimized clusters [14].
- Density-Based Spatial Clustering of Application with Noise (DBSCAN): Density-Based Spatial Clustering of Application with Noise (DBSCAN) clusters dataset based on density. It divides the data into several regions based on the density of the data. DBSCAN needs neighborhood distance named 'eps' as a hyperparameter. This distance decides the range of that cluster. That's how DBSCAN divides the dataset into several clusters based on the density of that region [4].
- Mean-Shift: Mean-shift is a kernel-based algorithm that does clustering by shifting the means. It finds out the region with a high density of data. An essential hyperparameter is the bandwidth of the dataset. Firstly, it randomly selects a part and then shifts that region's mean in the direction of high density. It is a very popular algorithm with robustness [2].
- Affinity Propagation: Affinity Propagation is a high-quality clustering model that clusters datasets by sending messages. In this algorithm, all the data sends a message to the other data. That's how they choose their representative of the cluster. Then, the closest members from the representative make the region of a cluster. In this way, they divide the dataset into several clusters [11].
- Decision Tree (DT): The DT uses a flowchart-like tree structure to learn simple decision rules and predict out-

comes. It begins with a root node, splits into branches based on the decision rules, travels through child nodes, and ends with the leaf nodes [16] that denote the result of the algorithm. The uncertainty in the dataset decides whether a feature should be the root node, child node, or leaf node. Although DT is a supervised algorithm, it can also solve regression problems.

- Random Forest: A Random Forest classifier combines multiple decision trees generated randomly. Then, each tree generates a prediction, and the prediction that receives the most votes is the outcome [18]. Using a bootstrap method, it builds a classifier by selecting features at random. When compared to other classifiers, it gives more accurate predictions.
- SVM: Binary classification is carried out by SVM on the mapped data points. It plots a hyperplane by separating the two distinct classes of data points [10]. It maximizes the distance between the hyperplane and the closest data points to the hyperplane. Solving multi-classification problems with SVM requires splitting the problem into minor binary classification problems.
- Logistic Regression: Logistic regression uses a statistical method to predict dichotomous outcomes but can be extended to solve multi-classification problems. It does not face the shift in threshold value issue as it uses the sigmoid function, an S-shaped curve. To represent two discrete classes, it predicts probabilities to maintain the prediction range between 0 and 1 [23].
- KNN: A classification model that uses distance as its primary factor is called a K-Nearest Neighbors Classifier. There is no prerequisite for any training stage. The process starts by measuring the distance between the test samples and the training samples, then selects the samples that are geographically closest to them, and finally classifies the data [8].K, the most essential hyperparameter, indicates the number of neighbors utilized in the calculation. The hyperparameter K tuning has a significant impact on the model's performance.
- Synthetic Minority Oversampling Technique (SMOTE): Synthetic Minority Oversampling Technique or SMOTE is a popular machine learning preprocessing approach for unbalanced data. To adjust imbalances in datasets, it creates synthetic samples focusing on the minority class. Other supervised learning paradigms, including multilabel classification, incremental learning, semi-supervised learning, and multi-instance learning, have been impacted by SMOTE[7].

2) Feedback Classification: We used different Text Classification Models for Feedback Classification. We fine-tuned these pre-trained models with the same data and parameters and finally compared the results. The models we used are mentioned below.

- BERT: Bidirectional Encoder Representations (BERT) is a machine learning framework specially designed for natural language processing (NLP). It is based on transformers [27] and pre-trained with text from Wikipedia. This model got famous because of its simple concept and analytical power. We can use BERT by fine-tuning our labeled dataset.
- DistilBERT: DistilBERT is a transformer model trained by refining the BERT base [22]. It is light, fast, cheap, and trim. As it followed the Knowledge Distillation [9] approach, it is smaller than BERT.
- RoBERTa: RoBERTa model has been developed like BERT but comprises the key parameters [15]. It also removed the following sentence pretraining mechanism and increased the mini-batch size and learning rates. These simple embedding tweaks made this model better at pretraining tricks.
- XLNet: The XLNet model is an add-on to the TransformerXL model [30]. With its autoregressive formulation, it overcomes the limitations of BERT and enables learning bidirectional contexts. One of its most significant benefits is it has no restriction on sequence length.
- XLM: The XLM model proved effective by extracting the BERT approach to multiple languages [13]. Causal language modeling (CLM), masked language modeling (MLM), and translation language modeling (TLM) are the objectives XLM followed. We selected the ones associated with CLM among various checkpoints as it supports our needs.

3) Content Modification: We used multiple methods for different types of modification of the contents, which again required different NLP models. A brief discussion of the methods and models is stated below.

- Name Entity Recognition: Name Entity Recognition(NER), an element of natural language processing (NLP), also known as entity chunking or entity extraction, recognizes preset categories of objects inside a text.
  - SpaCy: SpaCy is a library for NLP built for the extraction of information. The text is initially tokenized using spaCy, which then undergoes a processing pipeline that includes an entity recognizer, lemmatizer, tagger, and parser.
- Key Phase Extraction: In natural language processing (NLP), keyphrase or keyword extraction is a text analysis approach that pulls keywords and phrases out of the input text. Numerous activities can benefit from using these key terms, including content categorization, document summarization, and information retrieval.
  - Rapid Automatic Keyword Extraction(RAKE): RAKE is a keyword extraction algorithm that looks for important phrases in a text by analyzing the frequency of a word and how often it occurs with other terms.
- Knowledge Graph: A Knowledge Graph is a knowledge base that employs a graph-structured data model. It's a particular type of network graph that shows qualitative connections between facts, concepts, and events of real-world entities.

  - Relation Extraction By End-to-end Language Generation (REBEL): REBEL is a text2text model trained to convert an unprocessed phrase with entities and implicit relations into a set of triplets that specifically mention those relations.
- Text-to-Text Transfer Transformer (T5): T5 restructures NLP actions into a single text-to-text format, with text strings as both the input and the output. It takes the model text as input and trains it to generate some target text. It uses a text-to-text approach to generate translation, question answers, summarization, and classification. This transformer is used for the following actions:
  - Question-Answer Generation: Question Answer Generation in Natural Language Processing (NLP) is a task that involves creating models capable of automatically generating questions and answers from a text or PDF posed in natural language. The goal is to develop algorithms that can understand a given context or set of documents and then generate relevant and accurate questions and answers. T5 and Bert were used for this technique.
  - Text Summarization: Text summarization is an essential aspect of NLP. It is a technique of creating a summary by reducing the textual size without changing the semantic structure.
- Generative Pre-trained Transformer 3 (GPT-3): To carry out a range of natural language activities, GPT-3 processes text input. It combines natural language processing with natural language generation to comprehend and produce text written in real human language. For generating automated text, we used a GPT-3 transformer.

E. Evaluation metrics

First, we constructed the confusion matrix to evaluate the text classifier models. The shape of the matrix was 3:3, as we had three labels. Then, we calculated the accuracy and used silhouette score as an evaluation metric while clustering in activity classification.

- Accuracy: This metric is used to evaluate classification models. Accuracy is the percentage of correct predictions over the total number of predictions. We scaled the accuracy rate between 0 to 1.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

In (2), TP denotes True Positives, FP = False Positives, TN = True Negatives, and FN = False Negatives.

- F1 Score: It is an evaluation metric for binary classification, which represents the harmonic mean of precision and recall. It assesses a model's accuracy by balancing the trade-off between correctly identified positive instances and overall correct predictions across the entire data set.

$$F1Score = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{3}$$

In (3), TP denotes True Positives, FP = False Positives, and FN = False Negatives.

- Silhouette Score: The silhouette score is an evaluation metric for measuring clustering. In (4), a_i is the average distance between other points and the current point. b_i is the distance between the current point and the nearest centroid. Silhouette score gives a score from -1 to 1. The closest value to 1 is a more accurate cluster.

$$S_i = \frac{b_i - a_i}{max(a_i, b_i)} \tag{4}$$

- Confusion Score: A confusion matrix is a table that summarizes the performance of a classification model by displaying the counts of true positive, true negative, false positive, and false negative predictions. It provides a detailed snapshot of a model's effectiveness in binary or multiclass classification.

IV. Results and Discussion

A. Learning Style Detection

1) Activity Classification:

- MOOC Dataset: After evaluating the models on the MOOC dataset, we find the silhouette scores and execution times. We used 50,000 samples for the K-means clustering and DBSCAN models and 5,000 samples for the Mean-shift and Affinity Propagation models for limited resources. Table II depicts the essential hyperparameters, silhouette score, and execution time for each model.

TABLE II
MOOC Evaluation Table

Model	Hyperparameter	Silhouette Score	Execution Time
K-means clustering	K=3	0.169	0.345 sec
K-means clustering	K=2	0.162	0.346 sec
Mean Shift	Bandwidth=1.78	None	6.43 min
DBSCAN	eps=1.0	0.38	28.42 sec
Affinity Propagation	None	0.199	44.68 sec

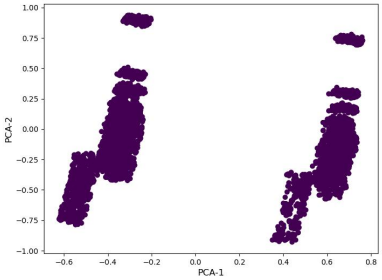

Fig. 3. Mean Shift clustering

As depicted in Table II, DBSCAN scored the best silhouette score and that is 0.38. K-means clustering with $K = 2$ scored 0.162, and $K = 3$ scored 0.169. The affinity propagation model scored 0.199, which is better

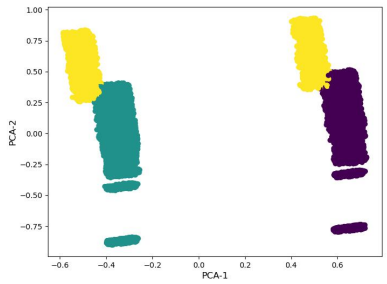

Fig. 4. K-means clustering (K=3)

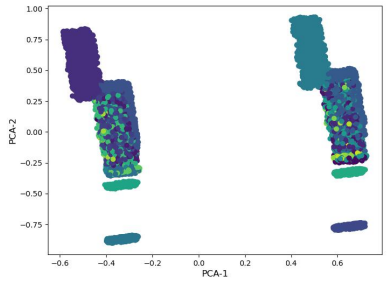

Fig. 5. DBSCAN clustering

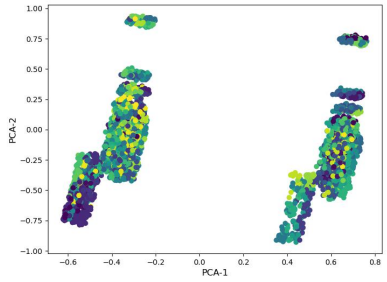

Fig. 6. Affinity Propagation clustering

than K-means clustering. On the other hand, the Mean shift divided the dataset into only one cluster as shown in Fig. 3. But if we see the cluster figures, we can see affinity propagation messed up with the whole dataset, illustrated in Fig. 4. DBSCAN did well, but it messed up in the middle part, shown in Fig. 5. Only K-means clustering did a solid clustering, displayed in Fig. 4.

- Custom Dataset: After evaluating the classification models on the Custom dataset, we find accuracies and f1 scores. No hyperparameter tuning was needed, as it did very well with some models. Fig. 7 shows the imbalanced graph and balanced graph together concerning the classes of our labels.

Table III describes the accuracies and f1 scores of the used classification models.

Fig. 8, Fig. 9 and Fig. 10 show the confusion matrix for

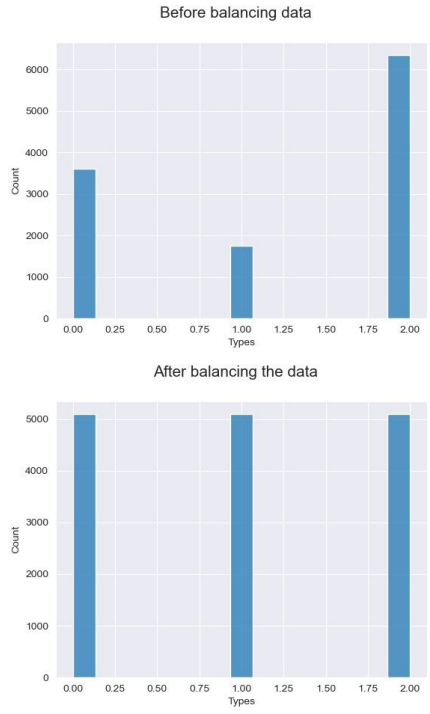

Fig. 7. Balancing labels using SMOTE

TABLE III
CUSTOM DATASET EVALUATION TABLE

Model	Accuracy	F1 Score
Decision tree	100%	100%
Random Forest	100%	100
SVM	61%	59%
Logistic Regression	49%	46%
KNN	100%	100%

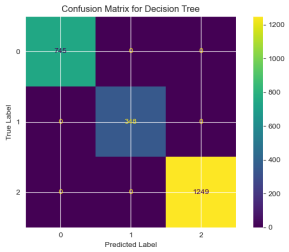

Fig. 8. Confusion Matrix for Decision Tree

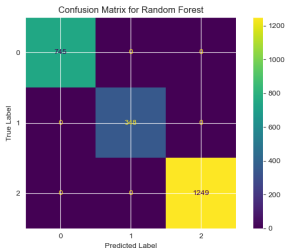

Fig. 9. Confusion Matrix for Random Forest

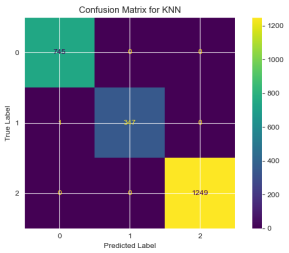

Fig. 10. Confusion Matrix for KNN

the models respectively Decision Tree, Random Forest and KNN.

We tried to classify the students into three categories: Visual, Auditory and Kinesthetic. But in the MOOC dataset, if we choose the clustering number 3, we do not find a well-divided graph. But our supervised custom dataset is doing fantastic with the machine learning models: Decision Tree, Random Forest and KNN.

2) Feedback Classification: We did many experiments on the VAK dataset, which included experimenting with and without preprocessing the data and experimenting with multiple methods and models Fig. 7 shows the evaluation loss for each model.

We used the same hyperparameter values for each model, as shown in Table V. We selected these hyperparameter values randomly.

TABLE IV
HYPERPARAMETERS FOR FEEDBACK CLASSIFICATION MODELS

Hyperparameter	Values
Max Square Length	128
Train Batch Size	8
Gradient Accumulation Steps	1
Evaluation Batch Size	8
Number of train epochs	2
Weight Decay	0
Learning Rate	4e-5
Adam Epsilon	1e-8
Warmup Ratio	0.06
Warmup Steps	0
Max Gradient Norm	1.0

From Fig. 11, we can see that the evaluation loss for Bert is 0.098 before data preprocessing. As we proceed through the preprocessing, the evaluation loss increases gradually. This is because Bert is built up in a way that does not require data preprocessing. The evaluation loss for RoBERTa evaluation loss is 0.096.

We also evaluate the models in terms of their total training time.

We also evaluate the models in terms of their total training time. From Fig. 12, we see that the total training time of the dataset with BERT before preprocessing is 11 minutes. For DistilBert, it is even more, which is 14 minutes. ROBERTa, XLNet, and XLM took comparatively less time, 6, 4, and 6 seconds, respectively.

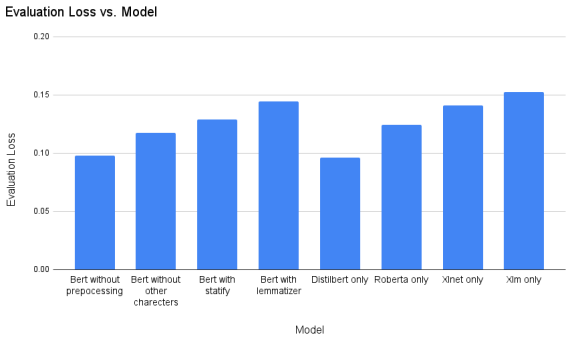

Fig. 11. Evaluation loss of the applied models on the VAK dataset.

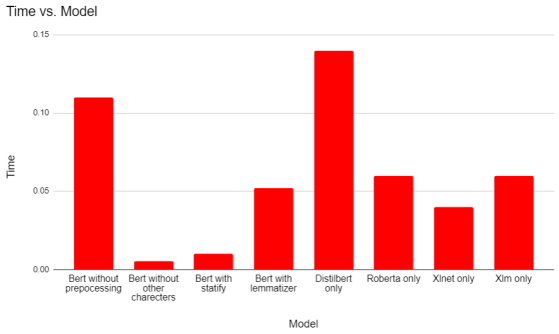

Fig. 12. Total training time vs. Model graph.

We can demonstrate the performance of each model in terms of evaluation loss and total training time in Fig. 13.

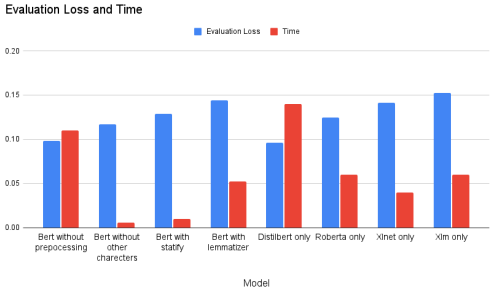

Fig. 13. Evaluation Loss and Total training time vs. Model graph

According to Fig. 13, we can say that although BERT and DistilBERT have more total training time, their evaluation loss is much lower than the other experimenting models.

TABLE **??** shows the accuracy rate for each model.

Fig. 14 shows the accuracy rate of the models. The lowest accuracy we get is from the XLM model, which is 85%. XLNet and RoBERTa have 87% and 92% accuracy rates, respectively. BERT has a better accuracy rate of 96%. Meanwhile, for our dataset, DistilBERT performs the best with an accuracy rate of 100% as shown in Table **??**. The visual representation of the model's accuracy is shown in Fig. 14.

TABLE V
ACCURACY RATE TABLE

Model	Accuracy
Bert without preprocessing	0.96
Distilbert only	1
Roberta only	0.92
Xlnet only	0.87
Xlm only	0.85

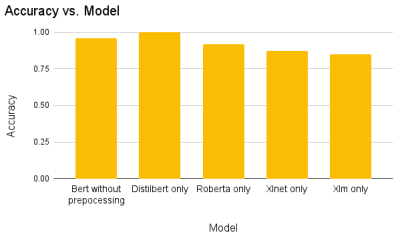

Fig. 14. Accuracy vs. Model graph

B. Content Modifier

For modifying content shown in Fig. 15, we used some pre-trained models with high performances and generated our desired output. Especially for summarization and question generation, while generating the sequences, we set the "output_scores" as "True" and checked for the sequence with the best value.

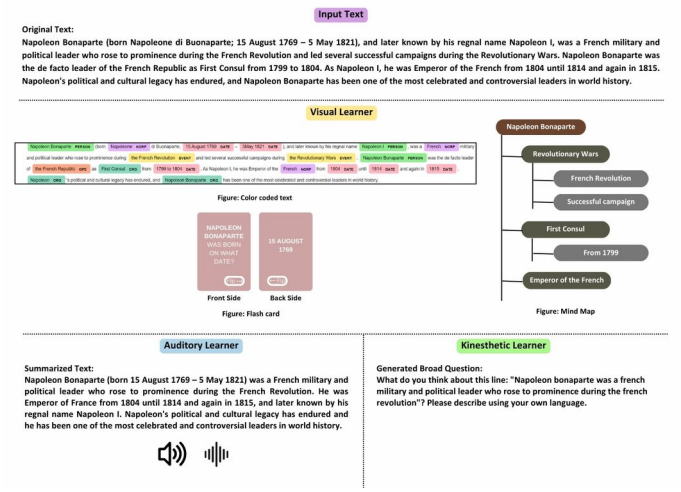

Fig. 15. Converted Content

V. LIMITATION

Followings are the limitations of our work:

- One of the biggest limitations was that we couldn't collect a satisfactory number of student information for the custom dataset.
- Using one uniform tokenizer would have significantly increased the models' efficiency and reduced the conversion time and number of operations, but we had to use different tokenizers for different conversions.
- We couldn't find any model with the most appropriate result in some cases, so we had to collect a dataset and train our model with it.

VI. CONCLUSION

The learning style detection and content modification project has provided valuable insights into understanding how individuals learn best and how we can improve their learning experiences. Using a KNN, decision tree, and random forest for the clustering model for students' activity detection and a DistilBERT model for students' feedback detection, we have successfully determined each student's learning style, allowing us to personalize academic content accordingly.

Our future work will focus on collecting information from more students. We wish to choose a uniform tokenizer to build and improve a custom model with it. This will not only increase efficiency by reducing the conversion time but will also reduce the repetition of work. In the future, we also plan to extend our work by including video materials from both the input and output end. These developments are meant to provide individualized learning opportunities and enhance overall academic results.

REFERENCES

[1] Junaid Ahmed, K Shah, and N Shenoy. "How different are students and their learning styles". In: *International Journal of Research in Medical Sciences* 1.3 (2013), pp. 212–215.

[2] Miguel A Carreira-Perpinán. "A review of mean-shift algorithms for clustering". In: *arXiv preprint arXiv:1503.00687* (2015).

[3] Xiaojun Chen, Shengbin Jia, and Yang Xiang. "A review: Knowledge reasoning over knowledge graph". In: *Expert Systems with Applications* 141 (2020), p. 112948.

[4] Dingsheng Deng. "DBSCAN clustering algorithm based on density". In: *International Forum on Electrical Engineering and Automation*. IEEE. 2020, pp. 949–953.

[5] Ouafae El Aissaoui et al. "A hybrid machine learning approach to predict learning styles in adaptive E-learning system". In: *International Conference on Advanced Intelligent Systems for Sustainable Development*. Springer. 2018, pp. 772–786.

[6] Richard M Felder, Barbara A Soloman, et al. *Learning styles and strategies*. 2000.

[7] Alberto Fernández et al. "SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary". In: *Journal of artificial intelligence research* 61 (2018), pp. 863–905.

[8] Gongde Guo et al. “KNN model-based approach in classification”. In: *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings*. Springer. 2003, pp. 986–996.

[9] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. “Distilling the knowledge in a neural network”. In: *arXiv preprint arXiv:1503.02531* (2015).

[10] Shujun Huang et al. “Applications of support vector machine (SVM) learning in cancer genomics”. In: *Cancer genomics & proteomics* 15.1 (2018), pp. 41–51.

[11] Hongjie Jia et al. “A density-adaptive affinity propagation clustering algorithm based on spectral dimension reduction”. In: *Neural Computing and Applications* 25 (2014), pp. 1557–1567.

[12] Ananthu S Kuttattu et al. “Analysing the learning style of an individual and suggesting field of study using Machine Learning techniques”. In: *International Conference on Communication and Electronics Systems*. IEEE. 2019, pp. 1671–1675.

[13] Guillaume Lample and Alexis Conneau. “Cross-lingual language model pretraining”. In: *arXiv preprint arXiv:1901.07291* (2019).

[14] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. “The global k-means clustering algorithm”. In: *Pattern recognition* 36.2 (2003), pp. 451–461.

[15] Yinhan Liu et al. “Roberta: A robustly optimized bert pretraining approach”. In: *arXiv preprint arXiv:1907.11692* (2019).

[16] Anthony J Myles et al. “An introduction to decision tree modeling”. In: *Journal of Chemometrics: A Journal of the Chemometrics Society* 18.6 (2004), pp. 275–285.

[17] Salman Naseer et al. “Named Entity Recognition (NER) in NLP Techniques, Tools Accuracy and Performance.” In: *Pakistan Journal of Multidisciplinary Research* 2.2 (2021), pp. 293–308.

[18] Mahesh Pal. “Random forest classifier for remote sensing classification”. In: *International journal of remote sensing* 26.1 (2005), pp. 217–222.

[19] Andrei Popescu-Belis et al. “GPoeT: a language model trained for rhyme generation on synthetic data”. In: *Proceedings of the 7th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*. Association for Computational Linguistics. 2023.

[20] Jaishree Ranganathan and Gloria Abuka. “Text summarization using transformer model”. In: *2022 Ninth International Conference on Social Networks Analysis, Management and Security (SNAMS)*. IEEE. 2022, pp. 1–5.

[21] Fareeha Rasheed and Abdul Wahid. “Learning style detection in E-learning systems using machine learning techniques”. In: *Expert Systems with Applications* 174 (2021).

[22] Victor Sanh et al. “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter”. In: *arXiv preprint arXiv:1910.01108* (2019).

[23] Kanish Shah et al. “A comparative analysis of logistic regression, random forest and KNN models for the text classification”. In: *Augmented Human Research* 5 (2020), pp. 1–16.

[24] Kristina P Sinaga and Miin-Shen Yang. “Unsupervised K-means clustering algorithm”. In: *IEEE access* 8 (2020), pp. 80716–80727.

[25] MG Thushara, Tadi Mownika, and Ritika Mangamuru. “A comparative study on different keyword extraction algorithms”. In: *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE. 2019, pp. 969–973.

[26] Rajshree S Vaishnav and KC Chirayu. “Learning style and academic achievement of secondary school students”. In: *Voice of research* 1.4 (2013), pp. 1–4.

[27] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).

[28] Sunu Wibirama et al. “A survey of learning style detection method using eye-tracking and machine learning in multimedia learning”. In: *International Symposium on Community-centric Systems*. IEEE. 2020, pp. 1–6.

[29] Daniel T Willingham, Elizabeth M Hughes, and David G Dobolyi. “The scientific status of learning styles theories”. In: *Teaching of Psychology* 42.3 (2015), pp. 266–271.

[30] Zhilin Yang et al. “XLNet: Generalized autoregressive pretraining for language understanding”. In: *Advances in neural information processing systems* 32 (2019).