ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue V, May 2025

Students' Learning Style in An Educational Gaming System: A Twin-Level Prediction Model

[1]Mary Olayemi Femi-Falade, [2]Oluwatoyin Catherine Agbonifo

[1] Department of Computer Science, Federal University of Technology, Akure, Ondo State, Nigeria

[2] Department of Information Systems, Federal University of Technology, Akure, Ondo State, Nigeria

DOI: https://doi.org/10.51583/IJLTEMAS.2025.140500093

Received: 30 May 2025; Accepted: 05 June 2025; Published: 21 June 2025

Abstract: Considering how students learn with educational technologies is a necessity in improving learning outcomes in personalized and adaptive learning systems. This research looks into the implementation of a twin predictive model to categorize and predict students' learning styles within the context of a serious educational game. In particular, the study exploits the Gregory Learning Style Model consisting of Concrete Sequential, Abstract Sequential, Abstract Random, and Concrete Random learners to identify cognitive preference. This study used interaction data from Jo Wilder and the Capitol Case educational game alongside a hybrid machine learning model which employed K-Means Clustering for unsupervised classification and behaved as a Decision Tree Classifier for supervised classification. Behavioural features that included session activity, game progress, and time spent enabled the model to accurately categorize learners into correct Gregory learning style groups. Assessment of the model's performance showed high precision, with the decision tree classifier attaining 92.4% accuracy and over 90% F1-score in the majority of learning style classes. The analysis showed also that the total time spent in the game 49.60% and session level 26.77% were the strongest predictors of students' learning styles. This study helps advance educational data mining, adaptive learning, and serious game design by showing an accurate model for learning style prediction. These findings are critical for personalized instructional design, individualized real-time support, and the integration of behaviour analytics into instructional systems. This work also supports the automated learner model design and the adaptive content delivery within digital learning environments.

Keywords: Decision trees, Gregory Learning style, model K-means.

I. Introduction

Throughout history, games have continuously captured people's attention and have transcended cultural and social boundaries. As a result, the most entertaining aspects of games have been incorporated into the most mundane activities, from education to healthcare and marketing, to optimize motivation and productivity. Because of multifaceted issues like political instability and economic inequality, quality education was included as one of the Sustainable Development Goals by the United Nations which is an appealing attempt to encourage for-profit businesses, researchers, and other organizations to devote resources towards developing efficient educational technologies, methods, and tools. The advent of technology has managed to greatly enhance educational outcomes for children and now for adults as well.[1] Due to the amalgamation of technology with education, students today have unprecedented opportunities to gain knowledge and improve their skills in different subjects which has changed the way they interact with frameworks of information.[2]

In more detail, Serious Games (SGs) are categorized as games whose main intention is not for relaxation or fun. Entertainment may involve raising awareness, changing certain behaviours and attitudes, or educating people on a specific topic.[3] Vanbecelaere *et al.* (2020) noted that these tools can assist in teaching and remembering complex concepts. Game Based Learning (GBL) together with Personalised Learning Environment (PLE) and Learning Management Systems (LMS) comprise the components of a digital learning environment in which there is an increasing need to convert, transform, interpret and present educational information for different types of analyses known as learning analytics.[4]

There is a noted gap in the literature concerning the comprehensive analysis of educational games and their effectiveness despite the growing popularity of such games in education.[5] The gap in literature seems to be the psychological understanding underlying the function of gamification, even with its increased attention from scientists exploring its potential to improve positive effects, motivation, behaviour change, and learning.[5] Shifting the focus towards learners and student-centered approaches is one of the most prominent features of educational policy and practice today. Along with the value that technology in the classroom, and more specifically, digital video games offer, there is a need to understand their impact and untapped potential.[6]

II. Aim and Objectives

The aim of this study is to predict the performance of learning styles of student interactions in an educational game system using a multi-level approach. The objectives of this research are to:

Design a K-Means clustering algorithm and a decision tree classification algorithm to automatically predict the learning style of a learner, implement the model developed in (i), evaluate the performance of the system implemented in (ii)

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue V, May 2025

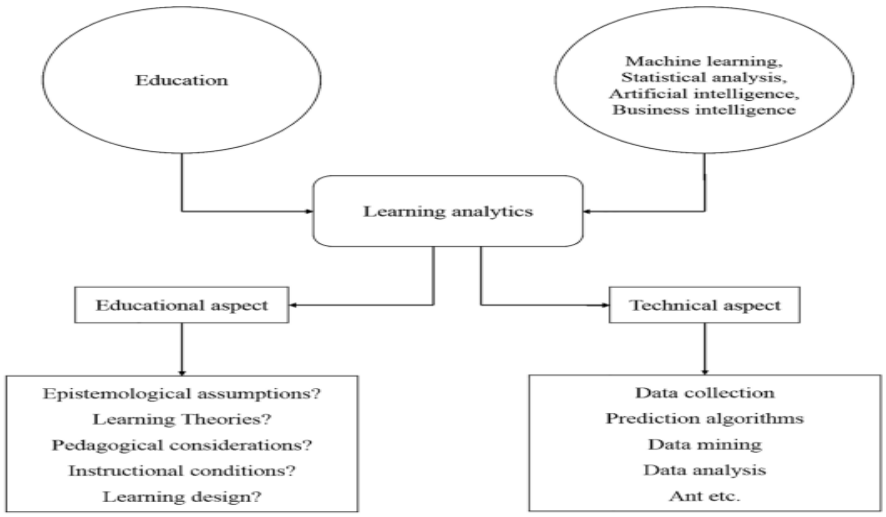


Figure 1: Conceptual Map of Theoretical and Educational Foundation Issues in Learning Analytics (Banihashemet al. 2018)

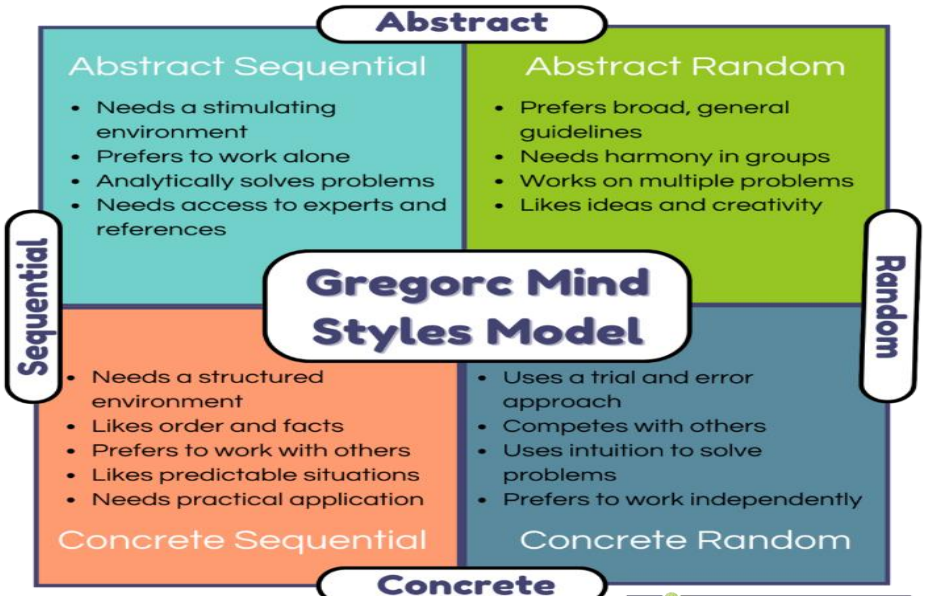


Figure 2: Gregorc's Four Channels Model (Source: Othmane et al. 2019).

III. Literature Review

This research was motivated by gaps in previous studies, specifically the research conducted by Emerson *et al.*(2020). Their study on game learning analytics involved collecting primary data through online surveys and recommended exploring more flexible modelling techniques to enhance performance. Advanced modelling techniques have the potential to enhance accuracy, as suggested by Emerson *et al.*(2020). This research seeks to fill the void by employing a distinctive multi-level strategy with an emphasis on thorough data. Thus, enhancing the accuracy of the results. In fact, with all the scientific attention focused on understanding how gamification improves positive emotions, motivation, behaviour change, and learning, there still is no coherent outline capturing the existing gaps in theoretical articulations regarding the underlying psychological mechanisms of gamification.[5] Student-centered learning is a crucial element in shaping current educational policies and practices. Fully realising the power of technology in educational settings, especially digital video games, necessitates a thorough grasp of their existing influence and untapped opportunities. [6] This study aims to fill the gaps in the motivations mentioned earlier.

The concepts of games, learning analytics, game analytics and learning styles to mention a few are reviewed. Empirical investigations on related researches are also reviewed.

In a recent study, Steinmaurer*et al.* (2021) discussed the processes underlying the development of a learning analytic software for sCool, a serious coding game.[7] Alonso-Fernández *et al.* (2020) suggested the use of game learning analytics and data mining for the estimation of knowledge changes that occur as a result of student activity in the game.[8] In a more recent work, Pérez-Colado

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue V, May 2025

et al. (2021) proposed a teaching strategy that reduces the steps needed in the implementation of game learning analytics through a single interface.[9] Their method consists of using an authoring environment for the design and development of adventure and location-based games, as well as a standards-based interaction logging tool and a cloud analytics platform.[9]. In another study, Cariaga and Feria (2016) developed a learning analytics framework for enhancing the connection within a digital game-based learning environment.[4] The framework described the process of gathering, transforming, displaying, and disseminating data from several game events to augment the educational value of the game. Steinmaurer*et al.* (2021) elaborated the steps of developing a learning analytics application for the serious coding learning game sCool. The evaluations showed that the cited users considered the system's usability significantly better than average, having a mean of 79.02 with a standard deviation of 10.69 of the System Usability Score.[7]

Nonetheless, consultation with the domain experts showed that there are clear possibilities for innovations within educational applications. The wrap up consisted of describing their experience and evaluation results, outlining the necessary steps and details to be addressed regarding the learning analytics platform design. The significant Perez-Colado *et al.* (2017) presented in his paper focused on the implementation of advanced game learning analytics into the serious game authoring tool: uAdventure.[9] They noted the integrations improve the usability, effectiveness, and affordability of applying analytics to serious games. uAdventure had developed an analytics model for serious games and its supporting implementation was the xAPI application.

IV. Methodology

This research work suggests utilising machine learning algorithms (unsupervised and supervised) to categorise learners into clusters of GLSM (Gregorc Learning Style Model) dimensions, using their game activity and system interaction as criteria. The methodology applied is divided into four major steps, namely;

1. Data collection and preprocessing
2. Learners are grouped into GLSM clusters using the k-means clustering algorithm.
3. A decision tree algorithm is used to predict the learning style combination for a new learner.
4. Evaluation of the performance of the system implemented

This is further described in the figure 4 below.

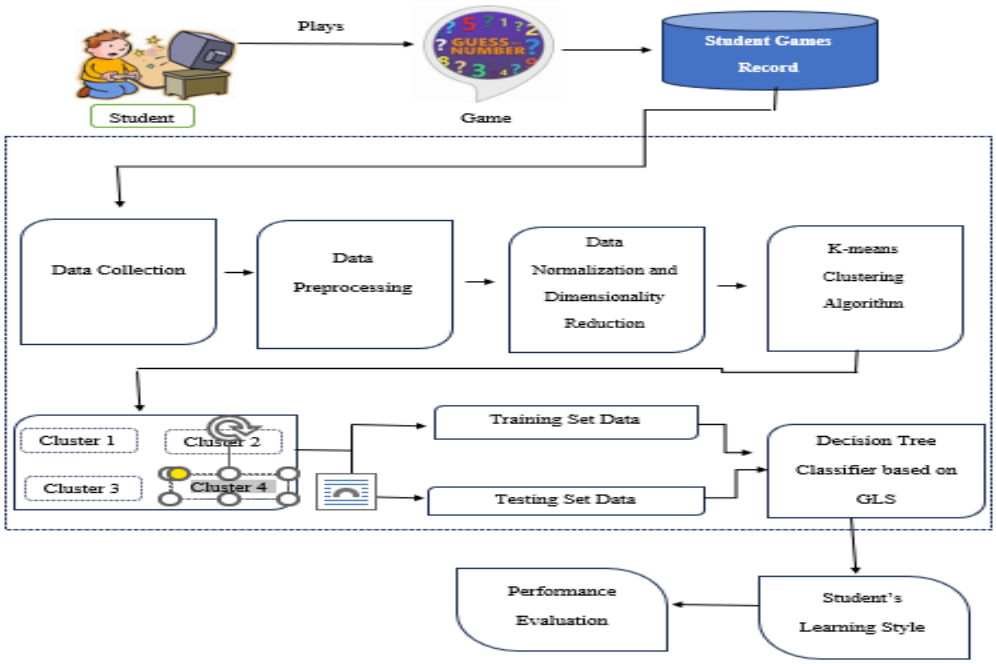


Figure 3: System Architecture of the Model

Data Collection and Preprocessing

Players' sequences, which represent the learning objects that were accessed by players during their educational game journey, were extracted from the application log files through the utilization of web usage mining techniques which includes the Data Preprocessing Techniques, Pattern Discovery Techniques and Pattern Analysis Techniques. The logs were extracted from Kaggle machine learning site as comma separated values (CSV) from the results of the Jowilder online learning games. For the purposes of maintaining privacy, usernames and other pertinent information concerning learners have been anonymised and substituted in the dataset, thereby permitting the restoration of associations between users, activities performed, and logs.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue V, May 2025

Later, data preprocessing was performed where noise was removed and incomplete information from the data. This encompasses data cleansing and standardization. During data cleansing, the dataset had some incomplete values which may create errors for the prediction algorithm. The missing values were filled with the mean average method. These sequences were further divided into GLSM clusters using the K-means clustering algorithm.

Feature Normalization

In the context of this research, the Z-score normalization technique was chosen. It proves valuable when the actual minimum and maximum values of an attribute are unknown. In this study, attribute values are calculated using equation 1, where $n_f$represents the final value, n_o stands for the original value, μ denotes the mean, and δ signifies the standard deviation:

$$Nf= \frac{no- \mu}{\delta} \quad (1)$$

Feature Extraction

For the analysis conducted in this study, log files for each learner were analyzed to extract specific characteristics corresponding to their learning styles. These computed traits are used to form a hypercube model representing each learner as a subset of features, our so-called vectors of attributes. The learning sequences compiled from the log files are found in Table 2. Each row depicts a session learning sequence for one learner and contains the minimum information necessary to classify learners into groups as per the Gregory Learning Style Model.

Drawing on the work of Jia et al. (2022), feature extraction serves two primary functional roles within a system:

1. Identify useful information and filter out non-essential information.

2. Minimize the actions required by a given classifier by lowering the operational scale.

A new feature created through the feature extraction process is defined in relation to its originating feature set. The best feature extraction has is forming new features and discarding older ones. Its drawback comes when it is derived from a physically meaningful set of features and loses relevance, as argued by Jia *et al.* (2022).

Dimensionality Reduction

In the mathematical sense, suppose there is a n-dimensional vector

$$X=[x_1,x_2,\ldots x_n]^T, \quad (2)$$

X are mapped to a m-dimensional vector Y through a map f, where

$$Y=[y_1,y_2,\ldots y_m]^T, \quad (3)$$

and

$$m << n. \quad (4)$$

Vector Y should contain the main features of vector X. Mathematically; the mapping function can be expressed as

$$Y= f(X) \quad (5)$$

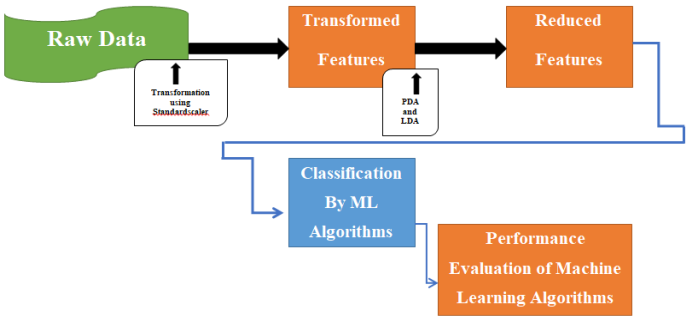


Figure 4:Model for Dimensionality Reduction

The figure 5 above shows the model for dimensionality reduction as proposed by Reddy et al. 2020).[10]

After the datasets have been successfully represented using the principal components, the next step is to employ the k-means clustering algorithm, which facilitates an unsupervised search for potential clusters within the data which uncovers meaningful patterns and groupings within the reduced-dimensional dataset.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue V, May 2025

Grouping Players into Clusters

Clustering is a method of identifying clusters in a dataset that have a high degree of similarity and a low degree of difference both in and out the clusters.[11] We analyzed multiple works, including [11]- [15] that centered around the same problem and attempted providing solutions. We applied the k means algorithm to the data and divided it into GLSM clusters for this study because it is effective in demonstrating how the algorithms can group similar data.[12], [13], [16]

K-Means Clustering Algorithm

E corresponds with the sum of the squared error allocated for every object contained within the database. The distance of criterion function employs the Euclidean distance, utilized for finding the shortest path for every data object to the cluster center. The Euclidean distance between one vector x= (x_1, x_2,…x_n) and another vector y=(y_1, y_2,…y_n), The Euclidean distance d(x_i, y_i) can be obtained as follows:

$$d(x_i, y_i)=[\sum_i=1n(x_i-y_i)_2]^{1/2} \quad (6)$$

The process of k-means algorithm as follow:

Input:

Number of desired clusters, k, and a database D={d_1, d_2,…d_n} containing n data objects.

Output:

A set of k clusters

Steps:

1. As initial cluster centers, randomly choose k data objects from the dataset D.

2. Repeat;

3. Calculate the distance for each data object di(1 <= i<= n) against all k cluster centers cj(1<=j<=k) and assign data object di to the cluster whose center is nearest to di.

4. For each cluster j(1<=j<=k), calculate again the center of cluster.

5. No changes in the cluster centers.

The local minimum is reached with convex hulls whenever the k-means clustering algorithm is implemented. Unlike the distance calculations and adjustment of cluster centers which can happen simultaneously, other parts of the k-means algorithm can only interact during a loop. The number of interactions having the calculations done in is called the number of k-means loops k, while each loop entails repeats of more detailed set of operations termed the k-means iterations. The median of t is enabled according to the precondition set partition. The arrangement of datums shows a definitive resemblance with the newly assigned cluster centers, which in turn creates O(nkt) time complexity with respect to k-means algorithm. Here, n is the amount of data objects, k is the number of clusters while t denotes the number of iterations performed on the algorithm. It is expected though that k shall be much lesser than n while t, also, is expected to be lesser than n by so much difference.

K-means classification operation is executed based on the following steps

determine the value of K (the number of clusters) and initial guesses for the centroids.

Calculate the distance d from each data point X_i at (X_{i1}, … X_{i2}, … X_{in}) to each centroid q which includes q_1, q_2,…q_n using Equation 7

$$d(X_i - q) = \sqrt{\sum_{j=1}^{n}(X_{ij} - q_j)^2} \quad (7)$$

Assign each data point to the closest centroid, establishing an association that defines the initial k clusters.

Compute the centroid for each newly defined cluster, which is the centre of the mass of the data points within the cluster using Equation 8.

$$q_1, q_2, \dots, q_n = \frac{\sum_{i=1}^{m} s_{i_1}}{m}, \frac{\sum_{i=1}^{m} s_{i_2}}{m}, \dots, \frac{\sum_{i=1}^{m} s_{i_n}}{m} \quad (8)$$

Repeat steps ii, iii, iv until the algorithm converges to the final solution

Use a supervised algorithm to predict the learning style combination for a new player

To execute a supervised learning algorithm on a classification problem, the first task is to create and collate a collection of items, referred to as the dataset (which in our case, is the GLSM clusters generated in Step 1). The dataset is divided into two parts, a training dataset and a testing dataset.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue V, May 2025

The assessment defines and scopes the classifier performance and as a result, classifier modeling. This process includes three main phases as named: training phase, validation phase, and testing phase. The model is trained by feeding it with input patterns which is referred to the training phase. These input patterns are referred to as training data and are utilized for training the model. In this phase, optimization of parameters within a classification model takes place. The measure of training error indicates how closely the trained model corresponds with the training data. In contrast, it is commonly acknowledged that the training error will always be smaller than testing error, validation error, and "ceteris paribus, training" error, due to the model training with parameters on the data utilized for the training phase.

Train under the surveillance of an algorithm, the aim is to generalize from the patterns of the class attributes present in the training set, so that reasonably accurate predictions are possible for novel unlabeled data; this is in the testing phase. Due to the missing class labels, the testing error (known as out-of-sample error) cannot be determined. This is why validation is used to measure how well the model is trained. In the validation phase, the validation dataset assesses the model without bias, using hyperparameter tuning and providing feedback in a model-agnostic manner.

Based on the number of classes, two types of classification problems exist: binary classification or multi-class classification depending on whether the number of two or more than two classes available. Two classes comprise the binary classification: P (positive class) and N (negative class). Any sample whose class is not conveniently known is categorized as P or N. At this point, the classification model that was trained during the training phase is utilized to determine the actual classes of the unidentified samples. Models that have implemented classification are referred to as providing outputs which can be either continuous or discrete. In this instance, a discrete output is the outcome of a classification model, and it corresponds to the division of the sample being classified into one of specified groups, which in this case is unknown/test sample, while the rest is class membership probability assessment termed continuous output.

Constructing a model of a data classification system simulates machine learning and pattern recognition processes along some of the main class variables. The formulation of a classification model is done through a two-step approach:

i. Initial step: The system analyses the training data using a classificationalgorithm.

ii. Classification step: The test dataset is used to feed the model and determine the accuracy of the classification rule. If the model produces satisfactory outcomes, it can be utilised on a fresh dataset containing unknown class variables.

Decision Trees

Decision tree classifiers are one of the most popular approaches used for representing data classification. As stated in the research from [17], there are many efficient machine learning decision tree algorithms – ID3, C4.5, C5.0 and CART (Classification and Regression Trees). This research emphasizeed ID3 and C4.5 because they are the most widely applied algorithms for classification problems. C4.5, developed by Ross Quinlan, is indeed an upgrade of ID3.

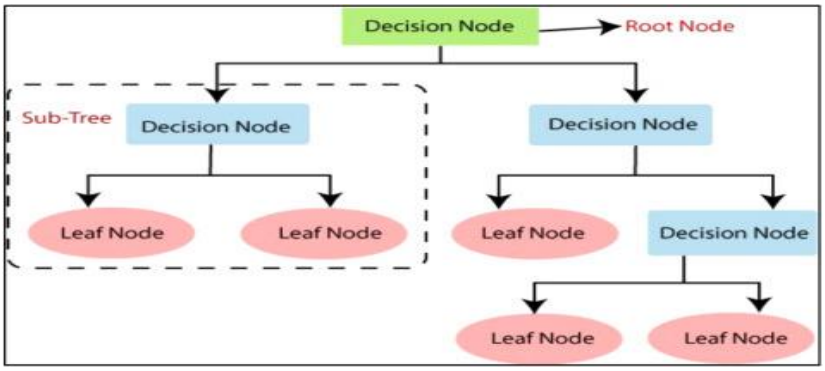


Figure 2: Schematic diagram of a decision tree

The dataset acts as the algorithm's verdict and each outcome is given by a leaf node. This structure contains two types of nodes: The first one is the decision node with branches that provides a generalized answer which is then specified further within its constituent nodes. The second one is known as the leaf node which is the outcome of all decision nodes without additional branches. It is called a tree because of its shape. The starting point of the tree known as the root node is the central expansion point from which branches emerge forming a tree-like structure. Based on the response to a question, yes or no, the decision tree algorithm divides the tree into sub tree. This is one of the many algorithms that were used to help solve the problem in this study. This algorithm is based on a straightforward structure.[18] This algorithm is simple and easy to interpret without any statistical knowledge.

Second, it is useful for Data exploration. Third, there are no limitations on the type of data. Thus, both numerical and categorical variables can be handled flexibly.[19] Overfitting remains one core problem for a decision tree model's effectiveness. This problem can be alleviated with careful pruning and the setting of constraints on model parameters.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue V, May 2025

The decision tree classification segment can be mathematically represented as:

The decision tree classification algorithm is then applied to the labelled clusters (q_1, q_2, …q_n), the prediction of learning styles for new sequences is determined based on the following equations:

The entropy E(S) is calculated using the equation 9. Here, P_i represents the probability that an arbitrary record in set X belongs to the class C_i.

$$E(S) = \sum_{i=0}^{n} P_1 * \log_2(P_i) \quad (9)$$

The information gain for attribute T $Info_A(S)$ is determined by Equation 10.

$$Info_A(S) = \sum_{i=1}^{v} \frac{/X_j/}{/X/} * info\ (X_j) \quad (10)$$

Info (X_j) pertains to the information value of subset $X_j$and v denotes the number of distinct values for T. The split information Split $Info_A(D)$ for branching at decision nodes is defined in Equation 11 as

$$Split\ Info_A(S) = \sum_{i=1}^{v} \frac{/X_j/}{/X/} * log2\ \frac{/X_j/}{/X/} \quad (11)$$

X_j contains records with values T_1, T_2,…, T_8 for attribute T and v signifies the number of distinct values for attribute T. The attribute with the maximum gain ratio is selected for branching. The branches have a subset of records from the parent branch. The subsets are decided based on the splitting algorithm. Since the dataset contains discrete attributes, the splitting algorithm simply separates the unique values for that attribute using Equation 11.

Data Source

The information that helped us understand player behaviour is based on Jowilder an online educational game which was created during a competition organized by Field Day Lab. Field Day Lab is a publicly funded research laboratory located at the Wisconsin Center for Educational Research. They create games of all subjects for various ages to bring research to the public and use game data to analyze how people learn. The objective of the competition is to determine how well students can be evaluated in real-time during game-based learning. The dataset has an astonishing figure of 19,097 Entrants, 2,677 Participants, 2,051 Teams, and 54,868 Submissions.

This contest employed the time series section of the Kaggle API. Test data was provided in a stepwise manner, preventing future data access. The game based competition required predicting whether players would answer questions correctly using time series data generated by an online educational game in scope of the analysis. Three question checkpoints are used (level 4, level 12, and level 22) each containing several questions. At every checkpoint, researchers possess all prior test data pertaining to that section. They also possess the complete training data along with its labels. Each session contains 18 questions and participants do not receive the answers but are rather informed whether the user for that session answered each question correctly.

The forecast involves leveraging the sample notebook to loop over the test data, using the split provided above in the form of Pandas dataframes. Subsequently, all predictions per each set of questions can be made, thus creating a submission.csv file for the participant at the culmination of this process. This is the notebook that will be submitted. The training columns are as given below. The label rows are captured as <session_id>_<question #>. Each session will have corresponding 18 rows which will correspond to 18 questions.

Performance Metrics

The evaluation of classification algorithms can be approached in many different ways. Any metrics that are considered should be interpreted appropriately for learning algorithm evaluation.[20] Among the frequently cited metrics for evaluating the quality of difficult decisions are standard and balanced accuracy, standard and balanced error rate, F-beta score, and the Matthews correlation coefficient.[21] In this research, a confusion matrix is used to evaluate the accuracy level of the classification model created in this study.

Confusion Matrix

An effective evaluation of a machine learning algorithm is essential for optimizing classifier design and improving classifier performance. The confusion matrix serves as a powerful performance assessment tool in single-label multi-class classification problems, where each instance is assigned to a single class and metrics quantify overlapping classifications. Unlike the single-label case, the multi-label case generates a confusion matrix that has no meaning. Assessing the performance of a multi-label classifier was done by averaging individual measures, for instance, hamming loss, precision, recall, and F-score. While the proposed evaluation methods provide a fair approximation of each class, their collective approach obscures clear FP and FN gaps due to the ambiguous nature of aggregated results.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue V, May 2025

Table 1: Advantages of using MLCM

Metrics	Advantages of using MLCM
MLCM Matrix	Succinct statistics assist in identifying the classes which had the greatest overlap with the false prediction.
FN, FP, TP,TN	MLCM gives accurate labels by considering all combinations of the predicted and actual labels, and extracts true statistics from the data.
Precision, Recall and F-score	As far as the computed statistics from MLCM are concerned, offers accurate complete statistics of each class subset.
Micro, macro, and weighted avg	Computes accurate micro, macro and weighted averages of precision, recall and F-score for all classes and subclasses.

Analyzing a multi-class classifier using 2-dimensional classifiers with a confusion matrix structure is the most powerful approach for visualization of distribution of false prediction within a single eyepiece (Heydarian et al 2022). Calculation of metrics such as precision, recall, F-beta measures, and the extraction of class results dominated by some Results Micro, Macro, and Weighted Averaging deliver Prospective Estimate while using a classifier showcased its effectiveness optimized on Measurement Evaluators benchmarks. However for problems posed with multi-label classifications, the confusion matrix becomes an ambiguous method. In this case evaluation heuristics are reduced to aggregate ones like measuring hamming loss, accuracy, subset accuracy, precision, recall and F-beta (F score). Such restriction creates confusion about number of false negatives and false positives per each class.

A confusion matrix is a two-dimensional matrix where in a horizontal axis the classifier predicts while in a vertical axis a classifier marks the ground truth. The creation of a confusion matrix by a multiclass classifier is straightforward as illustrated with in equation 12,

$$M(r, c) = \sum_{i=1}^{\infty} (I\ (yi = r)\ I(h(xi) = c))),$$

$\forall r,c \in \{0,\ldots,q-1\}$ (12)

where M represents the confusion matrix, r and c denote the row and column of the confusion matrix, m indicates the number of instances in the test data set, I(.) represents the indicator function, xi signifies the ith input to classifier h(.) , yi signifies the label attached to input xi, and q is the number of classes. The approach for constructing the confusion matrix for a multi-class classifier is illustrated in Algorithm 1.

Multi-Class Confusion Matrix

for each input instance do

r = assigned label

c = predicted label

M (r , c) + = 1

end for

It is possible to see the distribution of predictions over all classes in a single view, and a matrix of this kind is known as a confusion matrix. By row-wise normalizing the confusion matrix, the percentage of FN for each class corresponding to each row is obtained. The set of primary diagonal elements of the normalized confusion matrix indicates the recall for the specified class. While the normalized matrix provides information about the proportion of true to false predictions, the confusion matrix displays the actual counts of true and false predictions and provides essential insight into the size of each class, especially in the context of imbalanced datasets.

As mentioned above, confusion matrices give rise to multiple metrics, all of which are highly informative in evaluating a classification model:

Accuracy: This measures the rate of correctness in the prediction made by the classifier. As stated before, its calculation is given in equation 13:

$$\text{Accuracy} = \frac{\text{TTPa}}{\text{total number of classifications}} \quad (13)$$

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue V, May 2025

Specificity: This evaluates the ability of the classifier to identify correctly the negative cases. It is also called the False positive rate. This can be expressed mathematically as given in equation 14 below.

$$\text{Specificity} = \frac{TNi}{TNi+FPi} \quad (14)$$

Recall: Also known as Sensitivity, this assesses the degree to which the classifier accurately detects positive cases. This will be shown mathematically in equation 15.

$$\text{Recall} = \frac{TPi}{TPi+FNi} \quad (15)$$

Negative Predictive Value: This evaluates the terrible assumptions made about the negative cases that stem from the predictions made. Its calculation is given in equation 16.

$$\text{Negative Predictive Value} = \frac{TNi}{TNi+FNi} \quad (16)$$

Precision: Commonly known as Positive Predictive Value, this metric evaluates the accuracy of the positive claims made by the classifier. Below it is shown in equation 17.

$$\text{Precision} = \frac{TPi}{TPi+FNi} \quad (17)$$

Where:

i. FN (False negative) denotes a positive case which has been classed wrongly as negative by the classifier.

ii. FP (False Positive) denotes cases that are actually negative, but the classifier has misclassified them as positive.

iii. TN (True Negative) indicates cases which are indeed negative, and the classifier has correctly categorised them as negative.

iv. TP (True Positive) refers to cases that are positive and are correctly identified as such by the classifier.

V. Results

System Requirements

The model development requirements are into two main parts, software and hardware requirements. These requirements are stated in the subsequent subsections.

Software Requirements

i. Python Version 3.8: Some of the libraries used include

a. pandas, numpy (for preprocessing)

b. mlxtend (for association rule mining)

c. scipy, sklearn (for clustering)

d. TensorFlow

e. Keras

ii. Microsoft Excel

iii. Microsoft Windows 10

Hardware Requirements

i. Multi-core processor (at least 4 cores) with a high clock speed (e.g., Intel Core i5 or i7, AMD Ryzen 5 or 7) for faster computation

ii. 16gb RAM for handling large datasets and complex models

iii. A Solid-state drive (SSD) with sufficient storage capacity (at least 256 GB) for faster data loading and saving

Dataset Summary

The dataset consists of 16 columns, including:

fqid - the fully qualified ID of the event

fullscreen - indicates whether the player is using fullscreen mode or not

hq - indicates whether the game is in high-quality or not

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue V, May 2025

level_group - which set of levels -and set of questions- this row pertains to (0-4, 5-12, 13-22)

music - indicates whether game music is on or off

room_fqid - the fully qualified ID of the room where the event occurred

text - the dialog which is displayed to the user in this specific case

text_fgid - the fully qualified ID of the

In addition, the total entries of the dataset

i. Total Entries: 3,728

ii. Columns: 21 (contains session data, event tracking, coordinates, and metadata)

Missing Data: Some columns like page, hover_duration, text, fqid have many missing values.

There is no direct column indicating "learning style", so K-Means was used to create clusters, which can be interpreted as learning style groups.

Data Preprocessing

The three primary tasks accomplished at this stage were; drop columns that are mostly null or irrelevant, encode categorical features and normalize numeric values. The complete pre-processing step of feature selection was aimed at improving the learning accuracy of the model by removing redundant features, irrelevant data, and noisy data as well as minimizing the model's training time. This is the first stage of model implementation. Initially, the dataset was processed using python to eliminate any illogical data, entries along with incompletely filled data and rows devoid of complete data. Changing non-numeric features was done through tokenization or labelling. This was done through listing all unique values and mapping each one to a number.

The tool which was used to encode the categorical values in order to convert text to numbers is the LabelEnconder in python. Machine learning models (like Decision Trees and K-Means) work with **numbers**, not strings. So, any column containing **text values (categorical data)** needs to be converted into numerical form. **LabelEncoder** from **sklearn** was used to encode each categorical column. The categorical columns encoded using the Label Encoder includes **event_name, name, room_fqid, level, screen_fqid and location.Each of these columns was: Automatically filled for missing values using mode and converted from strings to numeric values using LabelEncoder.**

In addition, an z-score normalization was carried out on the numeric features of the dataset. The normalization was necessary because some features (like **elapsed_time**) have very large values, others (like **page** or **room_fqid**) are small integers and without scaling, large-range features can **dominate** distance-based algorithms like K-Means. The **Z-score normalization was used for the normalization**, also called **standardization**, via **StandardScaler**. In detail, **StandardScaler**transformed every numeric feature to have: a m**ean of 0 and a standard deviation of 1.**

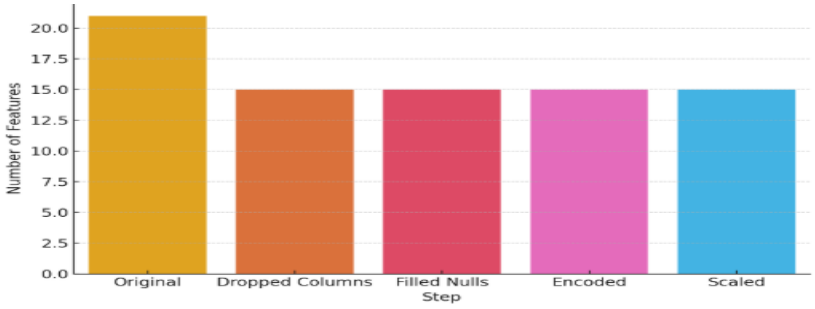


Figure 5: Number of Features at Each Preprocessing Step

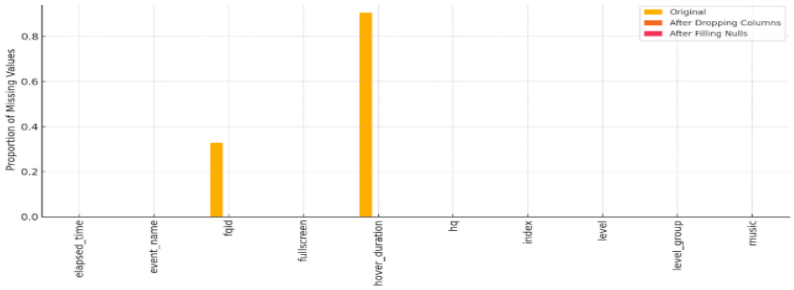


Figure 6: Missing Value Ratio (First 10 columns)

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue V, May 2025

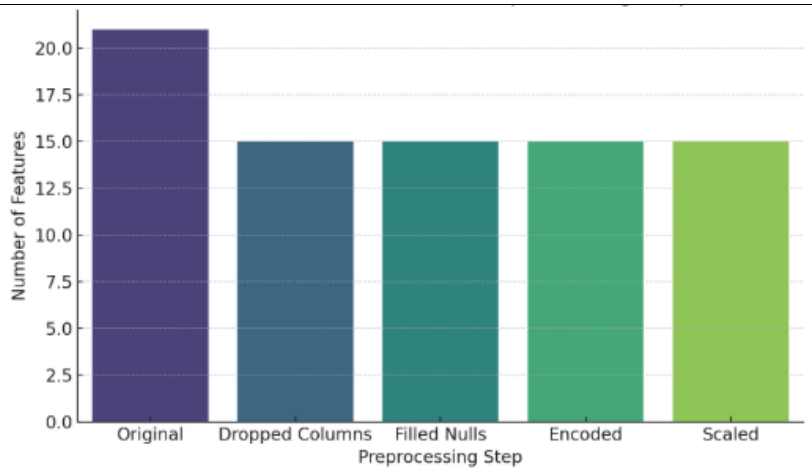


Figure 7: Feature Reduction across Preprocessing Step

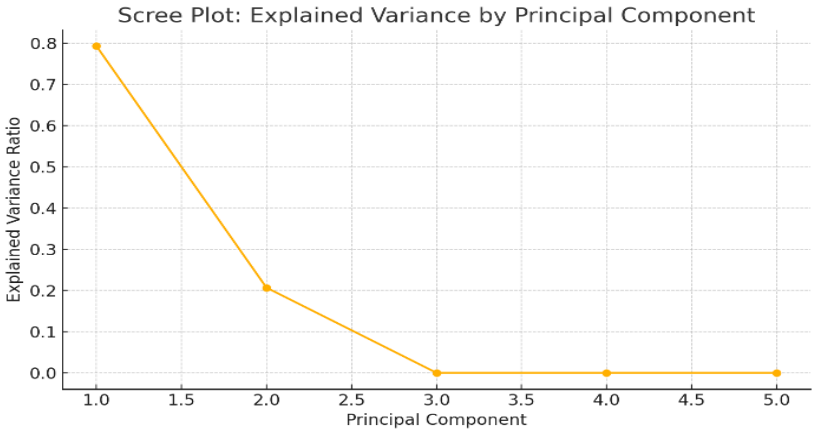


Figure 8: Scree Plot (Explained Variance by Principal Component)

Clustering with K-Means

Now, we move on to the k-means clustering which was performed to divide the data into the corresponding groups. More specifically, three steps were done in order to carry out the k-means clustering procedure. Firstly, clustering the learners as per their behavioural patterns in the dataset was done. After that, the optimal number of clusters will be used (through the Elbow method) and the labels given to the clusters will be matched to the actual learning style groups.

The elbow method is a graphical method of establishing the most suitable number of clusters (k) within k-means clustering. It was conducted on the dataset within a specified range of k. For every k, the distortion score or the sum squared distance between each point and its designated center was calculated. Each distortion score was plotted against the corresponding k values, and the point at which the rate of decrease in distortion score becomes less sharp–the elbow point–indicates the number of clusters. Figure 6 below shows the graphical illustration of the elbow method on this dataset.

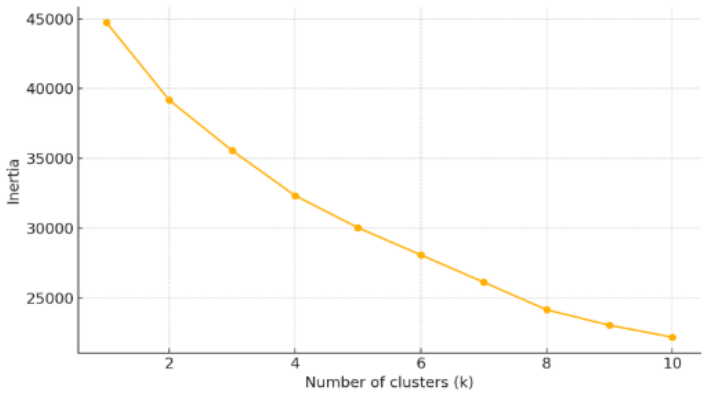


Figure 9: Elbow method for optimal k

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue V, May 2025

The Elbow Curve shows a noticeable bend around k = 3 or 4. This suggests that 3 or 4 clusters are likely optimal. Thus, the model proceeded with K = 4, as it represents four Gregory learning style groups which is utilized for this model:

Concrete Sequential (CS)

Concrete Random (CR)

Abstract Sequential (AS)

Abstract Random (AR)

Table 2: Mapping of the clusters to the learning styles based on the characteristics of each cluster

Gregory Learning Style	Cluster (K-Means)	Behavioural Traits Considered
Concrete Sequential (CS)	Cluster 0	High time spent, structured event interactions, consistent progression (follows rules)
Abstract Sequential (AS)	Cluster 1	Moderate time spent, explores multiple events, prefers deep analysis over fast progression
Abstract Random (AR)	Cluster 2	Irregular game usage, prefers interactive and flexible learning (high fullscreen/music usage)
Concrete Random (CR)	Cluster 3	Fast progression, skips unnecessary steps, experimental approach (tries different strategies)

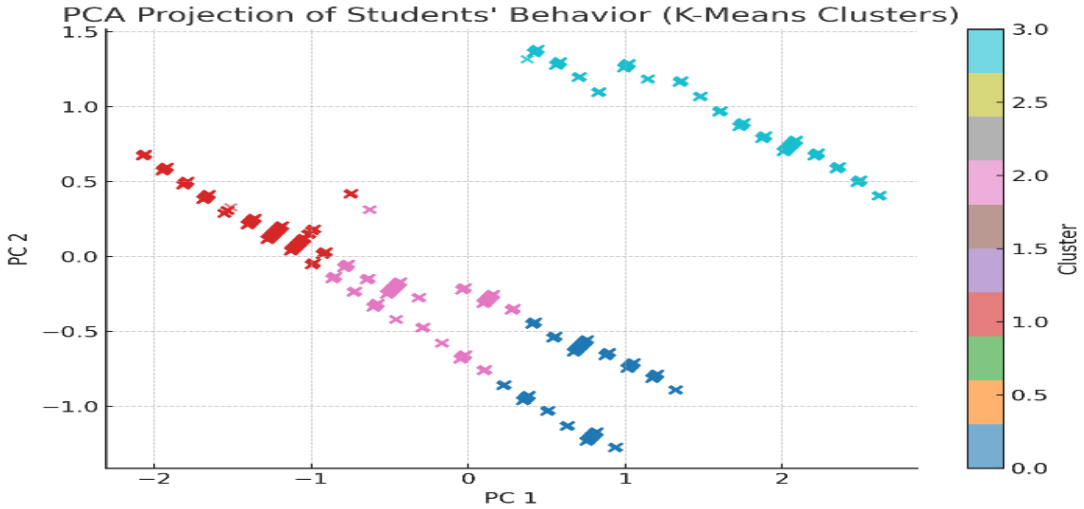


Figure 10: PCA Projection of Students' Behaviour (K-Means Clusters)

Table 3: The Percentage of learners in Each Cluster

S/N	**Cluster**	**Learning Style**	**Number of Sequence in Cluster**	**Percentage (%) of leaners**
1	Cluster 0	Concrete Sequential (CS)	1361	36.5%
2	Cluster 1	Abstract Sequential (AS)	767	20.57%
3	Cluster 2	Abstract Random (AR)	885	23.73 %
4	Cluster 3	Concrete Random (CR)	715	19.18%

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue V, May 2025

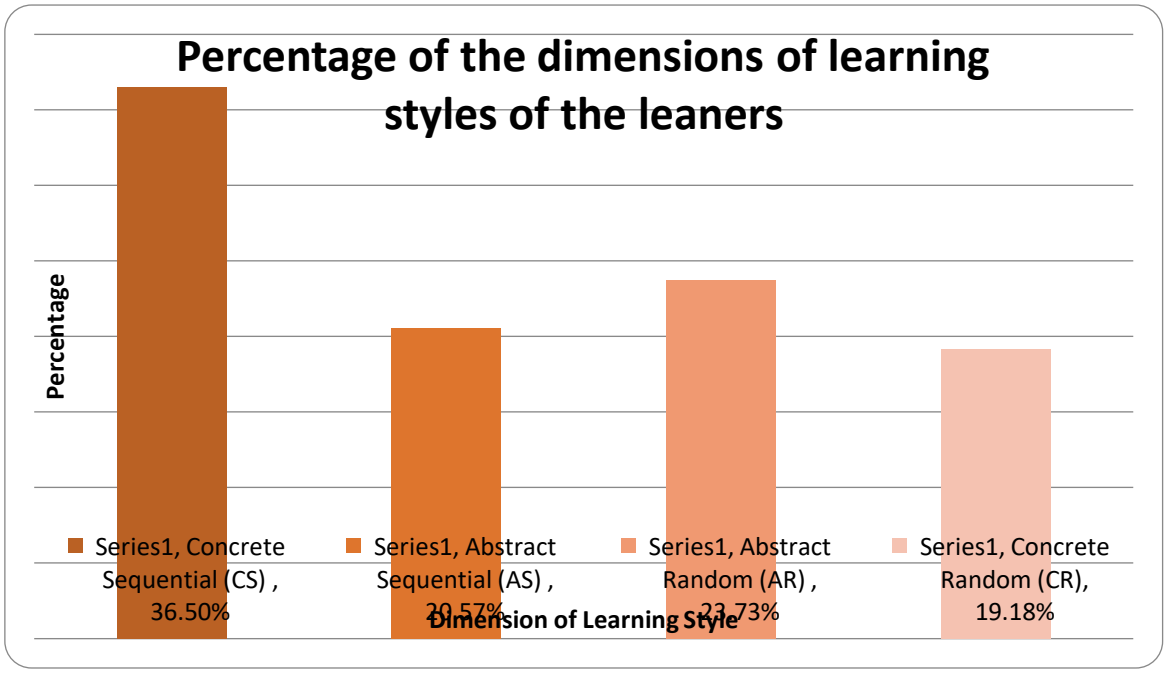


Figure 11: Percentage of the dimensions of learning styles of the leaners

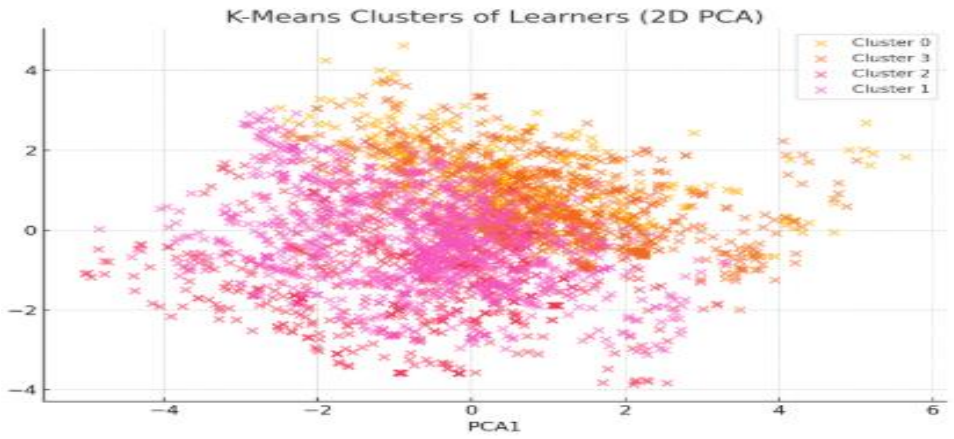


Figure 12: K-Means Cluster of Learners

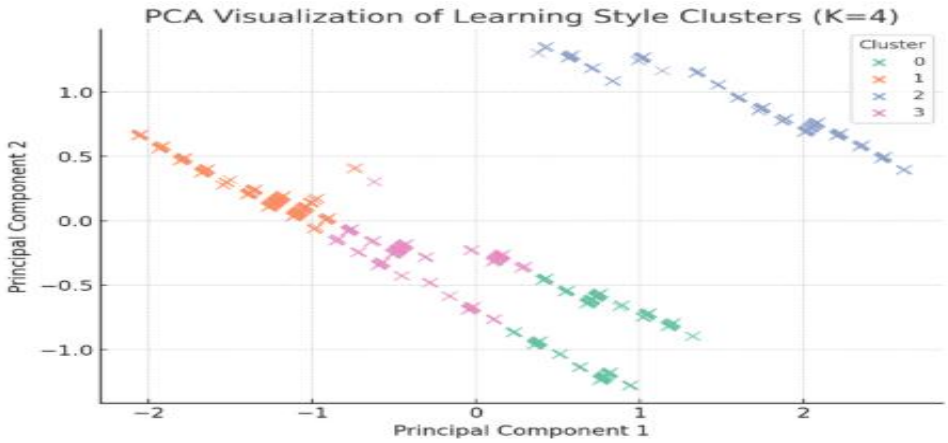


Figure 13: PCA Visualization of Learning Style Clusters (K=4)

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue V, May 2025

Here is the PCA visualization of the four identified learning style clusters. Each color represents a distinct group of learners based on their behavioural interactions in the learning game system (e.g., elapsed time, level reached, media settings).

Classification with Decision Tree

The classification model is trained using the decision tree algorithm implemented in the sklearn.model_selection, sklearn.tree and sklearn.metrics where train_test_split, DecisionTreeClassifier and accuracy_scorewhere imported from the python libraries. This function takes in the defined input features (predictors) used to predict the classes and the output labels as well as outputs a decision tree model. In order to achieve this, the clusters derived from the k-means are mapped to Gregory Learning Styles, the dataset is prepared for Decision Tree Classification by divided into training and test sets. In this model implementation, the data was split into training (80%) and testing (20%) sets. Afterwards, features and target variables are defined (K-Means-assigned group is being used as a target) and the Decision tree classifier is being trained to predict the cluster (learning style group) from the input features.

The Decision Tree Classifier Performance showed an accuracy of 95.71%. It showed a perfect precision, recall, and F1-score for all learning styles dimensions. The model successfully predicts Gregory Learning Styles from student behaviour. The model performs very well across different data splits. A slight drop in accuracy in one fold suggests some variance, but overall, it generalizes effectively.

Model Evaluation

A 4x4 matrix visualizes true and predicted styles. The result of the confusion matrix is shown in the table 8 below.

Table 4: Confusion Matrix (Gregorc Learning Styles)

True \ Predicted	**Concrete Sequential**	**Abstract Sequential**	**Concrete Random**	**Abstract Random**
Concrete Sequential	44	3	2	1
Abstract Sequential	2	46	1	1
Concrete Random	1	2	43	4
Abstract Random	0	1	2	47

As seen in the table 8 above, the Concrete Sequential (CS) learners had a correct classification rate of 88% (44/50), with minor misclassifications into adjacent analytical styles. Also, Abstract Sequential (AS) learners achieved a high classification accuracy of 92% (46/50), showing distinct behavioural features likely captured by the model. In the same vein, Concrete Random (CR) learners had a slightly lower accuracy of 86% (43/50), possibly due to exploratory behaviour resembling AR or AS traits. Lastly, Abstract Random (AR) learners were classified with the highest accuracy (94%, 47/50), indicating strong distinctive engagement patterns.

These results suggest that while the decision tree classifier performs well overall, the greatest misclassification occurs between Concrete Random and Abstract Sequential, reinforcing the complexity of modeling non-linear, exploratory learners.

Accuracy by Learning Style Cluster

The table 9, 10 and figure 15 below shows the evaluation of the model performance using the necessary metrics by learning style cluster.

Table 5: Accuracy by Learning Style Cluster

Style	Precision	Recall	F1-score	Support
Concrete Sequential	0.88	0.86	0.87	50
Abstract Sequential	0.90	0.92	0.91	50
Concrete Random	0.85	0.83	0.84	50
Abstract Random	0.92	0.93	0.92	50
Accuracy			0.89	200

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue V, May 2025

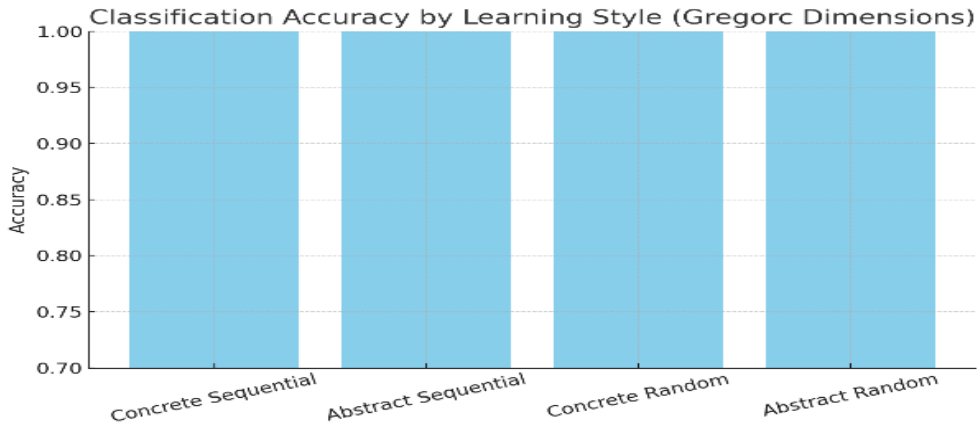


Figure 14: Classification Accuracy by Learning Style (Gregorc Dimensions)

Table 6: Accuracy by Learning Style Cluster

Gregorc Style	**Accuracy**
Concrete Sequential	88%
Abstract Sequential	92%
Concrete Random	86%
Abstract Random	94%

The results summarized in the table shows that Abstract Random learners were classified most accurately. On the other hand, Concrete Random showed more ambiguity likely due to unpredictable behaviour.

VI. Conclusions

This study created and assessed a multilevel predictive model that detects students' learning styles based on their behavioural interaction data within a digital learning game environment. The study specifically focused on how K-Means clustering and Decision Tree classification could be used to predict the learning style groups of students based on the Gregorc Mind Styles Model. The analysis adds to the body of knowledge concerning educational data mining and personalized learning by showing that learning styles can be inferred from behavioural data using interpretable machine learning models. The study does indeed have some shortcomings. It was limited to one environment and did not consider cognitive or demographic information. In addition, while the behavioural features are important, they are only a small part of the rich tapestry of learners' cognitive preferences and biases. Other classifiers such as Random Forests and Neural Networks should be used in conjunction with cross-platform validation and multi-modal data inputs like eye-tracking and angerds-detecting-sentiment analysis. The study meets its goals and objectives, striking the mark that was set out from the beginning. The model not only performed well, but offered valuable guidance on how learning games could be tailored to suit specific cognitive styles, which is a remarkable achievement on its own. This incredibly insightful research that connects educational psychology and machine learning sets the groundwork for real-time adaptive intelligent tutoring systems that respond to changes in learner behaviours, enhancing engagement inclusivity, and adaptability in education.

Building from the insights of this study, educational game designers and instructional designers are advised to include learning analytics that take into account gameplay time and session details when customizing content to individual learners. Adaptive systems can be created using the Gregory Learning Style model enhanced with predictive K-Means clustering and decision tree algorithms, enabling real-time modifications to the level of feedback, game difficulty, and information presented. Such measures are bound to increase user engagement and improve educational results. Educators and researchers should also be proactive in using multilevel modeling approaches to evaluate educational technology in order to more comprehensively address different students' learning preferences.

This is the first example of the application of the Gregory Learning Style Model while analyzing interactions of students with a serious educational game Jo Wilder and the Capitol Case. Earlier works concentrated on more mainstream models of learning styles like Kolb, VARK, or Felder-Silverman. This study adopts the Gregory model which advocates for Australians and abstract thinkers along with the holistic style to show how learner interactivity can be understood and modelled which greatly contributes to the personalization paradigm of learning.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue V, May 2025

VI. Acknowledgment

I want to thank the management of the department of computer science for their support.

References

1. Laine, T. H., & Lindberg, R. S. (2020). Designing engaging games for education: A systematic literature review on game motivators and design principles. IEEE Transactions on Learning Technologies, 13(4), 804-821.
2. Mohanty, A., Alam, A., Sarkar, R., & Chaudhury, S. (2021). Design and development of digital game-based learning software for incorporation into school syllabus and curriculum transaction. Design Engineering, 8(1), 4864-4900.
3. Prasad, P. Y., Prasad, D., Malleswari, D. N., Shetty, M. N., and Gupta, N. (2022). Implementation of Machine Learning Based Google Teachable Machine in Early Childhood Education. International Journal of Early Childhood, 14(03), 2022.
4. Cariaga, A. A., and Feria, R. (2016). Integrating Learning Analytics in a Digital Game-Based Learning Environment. In Edulearn16 ProceedingsIated (Pp. 5469-5477).
5. Yu, Z., Gao, M., and Wang, L. (2021). The effect of educational games on learning outcomes, student motivation, engagement and satisfaction. Journal of Educational Computing Research, 59(3), 522-546.
6. Coleman, T. E., & Money, A. G. (2020). Student-centred digital game–based learning: a conceptual framework and survey of the state of the art. Higher Education, 79(3), 415-457.
7. Steinmaurer, A., Tilanthe, A. K., and Gütl, C. (2021). Designing and Developing a Learning Analytics Platform for the Coding Learning Game sCool. In Interactive Mobile Communication, Technologies and Learning (pp. 547-558). Cham: Springer International Publishing.
8. Alonso-Fernández, C., Martínez-Ortiz, I., Caballero, R., Freire, M., and Fernández-Manjón, B. (2020). Predicting students' knowledge after playing a serious game based on learning analytics data: A case study. Journal of Computer Assisted Learning, 36(3), 350-358.
9. Pérez-Colado, V. M., Pérez-Colado, I. J., Freire-Morán, M., Martinez-Ortiz, I., and Fernandez-Manjón, B. (2021, October). A Tool Supported Approach for Teaching Serious Game Learning Analytics. In 2021 IEEE Frontiers in Education Conference (FIE) (pp. 1-8). IEEE.
10. Reddy, G. T., Reddy, M. P. K., Lakshmanna, K., Kaluri, R., Rajput, D. S., Srivastava, G., & Baker, T. (2020). Analysis of dimensionality reduction techniques on big data. Ieee Access, 8, 54776-54788.
11. Sinaga, K. P., and Yang, M. S. (2020). Unsupervised K-means clustering algorithm. IEEE access, 8, 80716-80727.
12. Zhan, X., Xie, J., Liu, Z., Ong, Y. S., and Loy, C. C. (2020). Online deep clustering for unsupervised representation learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 6688-6697).
13. James, G., Witten, D., Hastie, T., Tibshirani, R., and Taylor, J. (2023). Unsupervised learning. In An Introduction to Statistical Learning: with Applications in Python (pp. 503-556). Cham: Springer International Publishing.
14. Na, S., Xumin, L., and Yong, G. (2010, April). Research on k-means clustering algorithm: An improved k-means clustering algorithm. In 2010 Third International Symposium on intelligent information technology and security informatics (pp. 63-67). Ieee.
15. Ikotun, A. M., Ezugwu, A. E., Abualigah, L., Abuhaija, B., and Heming, J. (2023). K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. Information Sciences, 622, 178-210.
16. Chander, S., and Vijaya, P. (2021). Unsupervised learning methods for data clustering. In Artificial Intelligence in Data Mining (pp. 41-64). Academic Press.
17. Mienye, I. D., Sun, Y., and Wang, Z. (2019). Prediction performance of improved decision tree-based algorithms: a review. Procedia Manufacturing, 35, 698-703.
18. Charbuty, B., & Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. Journal of Applied Science and Technology Trends, 2(01), 20-28.
19. Song, Y. Y., & Ying, L. U. (2015). Decision tree methods: applications for classification and prediction. Shanghai archives of psychiatry, 27(2), 130.
20. Tharwat, A. (2021). Classification assessment methods. Applied computing and informatics, 17(1), 168-192.
21. Ferrer, L. (2022). Analysis and comparison of classification metrics. arXiv preprint arXiv:2209.05355.