BIRMINGHAM CITY University

ANALYZING STUDENT'S ADAPTABILITY LEVEL IN ONLINE EDUCATION USING DATA MINING TECHNIQUES.

Name and Student IDs

SAMUEL CHUKWUMA EZEMBU: 22217206:

MUHAMMAD NIZAR HAZIM BIN ZARUL ANUAR: 22138147

COURSE CODE: CMP7206

COURSE TITTLE: DATA MINING

MODULE LEADER: DR MUHAMMAD AFZAL

TABLE OF CONTENTS

LIST OF TABLES

TABLE OF FIGURES

1.0 DOMAIN DESCRIPTION

Data mining involves the process of discovering patterns, relationships, and insights from large datasets using various statistical and machine learning techniques. It is important in education to gain insights into student performance, identify learning trends, and develop personalized interventions. Additionally, data mining can aid in curriculum development, resource allocation, and policy-making.

S. A. Kumar & M Vijayalakshmi (2011) defined educational data mining as a burgeoning area dedicated to devising techniques for examining the distinctive data forms produced within educational contexts. The objective is to employ these techniques to acquire enhanced understandings of students and the learning environments they engage in.

The use of data mining in education offers several benefits, such as early identification of struggling students, the identification of factors influencing student success, and the evaluation of educational interventions. Examples of successful data mining applications in education include predictive modelling, association rule mining, and clustering techniques. These examples demonstrate the potential of data mining to enhance educational practices and student success.

Traditional methods cannot process and analyse the complex and high-volume educational data. Therefore, data mining provides the methodology and technology to transform the massive amounts of raw educational data into useful information and knowledge.

2.0 PROBLEM DEFINITION

In recent times, there has been a substantial surge in the utilization of data mining methods in the analysis of educational data. This growth has been partly propelled by the significant increase in the availability of educational data. The implementation of information systems within educational institutions has facilitated the collection and retention of large volumes of data. Additionally, the rise of contemporary distance learning has contributed to a greater abundance and variety of educational data categories. Consequently, the essential prerequisites for applying data mining techniques to the field of education have materialized. This has led to the emergence of educational data mining as a distinct interdisciplinary domain, a subject explored in numerous studies such as the findings of Romero & Ventura (2007), Papamitsiou & Economides (2014), Peña-Ayala (2014), Thakar et al. (2015), Sukhija et al. (2015), and Del Río & Pineda Insuasti (2016).

Assessing the adaptability level of learners is crucial before implementing any learning environment, whether online, in-person, or hybrid. Understanding how well learners can adapt to different teaching methods and styles allows educators to deliver instruction in the most effective way. It also enables efficient allocation of educational resources by matching teaching styles to students' adaptability. Evaluating learners' adaptability is thus a key step in developing optimal learning environments tailored to the students.

3.0 LITERATURE REVIEW

The inception of **educational data mining** (EDM) can be dated to 1982, when the initial workshops and conventions centred around EDM took place. Notably, the "*International Conference on Artificial Intelligence in Education*" stands as the first conference in the EDM realm. Furthermore, the "*Workshop on Applying Machine Learning to ITS Knowledge Design/Construction*," hosted in Canada, marked the earliest workshop focused on EDM (Romero and Ventura, 2013). These early occurrences laid the groundwork for the emergence of EDM as a discipline and marked the commencement of its evolution as a separate domain of research.

The **educational data mining (EDM)** literature can be categorized into several chronological developmental phases:

- Between 1995-2005, EDM focused on relationship mining.
- From 2005-2009, prediction methods gained popularity.
- Since 2009, techniques like Item Response Theory, Bayesian Networks, and Markov Chains have been applied in psychometric and learner modelling frameworks (Baker and Yacef, 2009).

Baker (2010) identifies **four core EDM** application domains:

- Student Modelling - Modelling student characteristics like prior knowledge, attitudes, motivation, metacognition.
- Domain Modelling – Discovering and improving models of the knowledge structure within a domain.
- Pedagogical Support - Testing and enhancing applications that provide pedagogical assistance, determining optimal support strategies.
- Investigating factors affecting learning to design improved environments and provide empirical evidence.

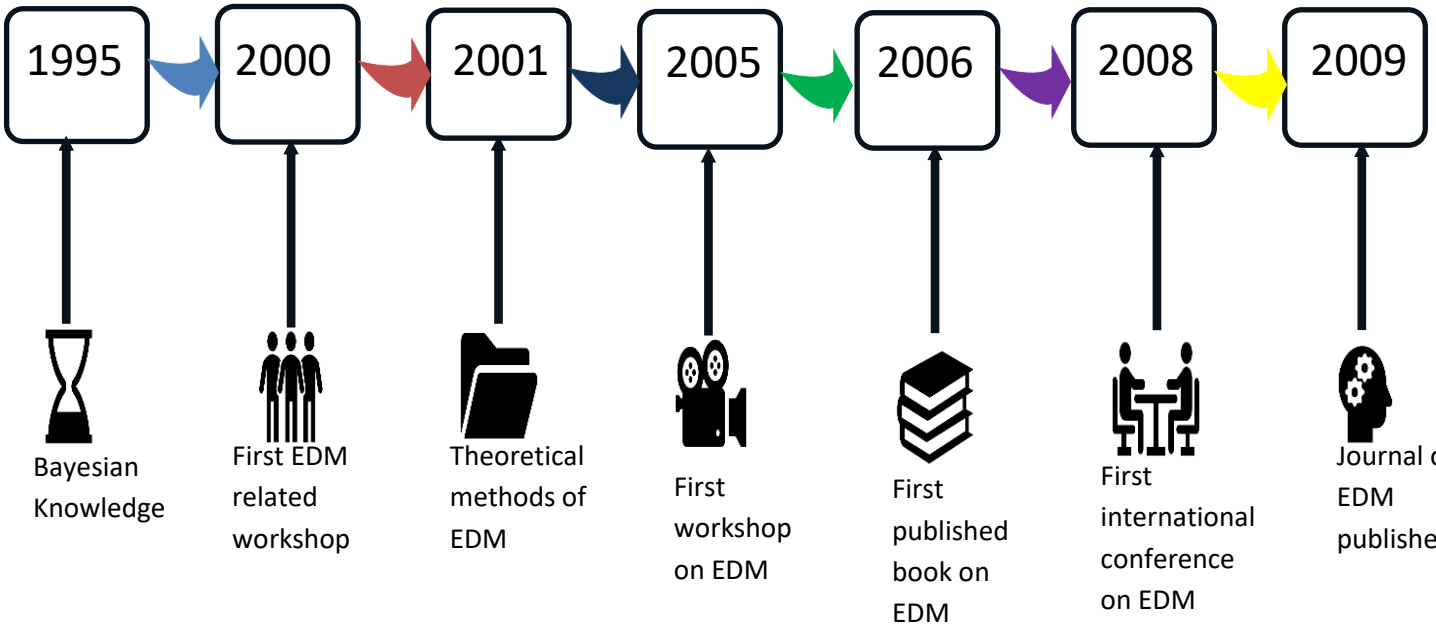

Figure 1 Historical development of educational data mining

E-leaning/Online learning as defined opined by Abou El-Seoud et al (2014) denotes a technology-centred method of education and learning. It encompasses the electronic distribution of educational materials and lectures to students. Functioning as a type of remote learning, e-learning enables students to electronically access educational content and resources via the internet, regardless of their location or the time. It relies on internet platforms to digitally present lectures and supplementary materials.

This literature review of existing literature follows the **PICO** framework (Pai et al., 2004) commonly used in medical research. The review focuses on articles that examine academic performance in secondary schools, higher education institutions, and related online tools using data mining techniques. The methods, algorithms, and features used in these studies are reviewed, Comparisons are made between different evaluation measures used. The review summarizes the most commonly used data mining methods, algorithms, and features as well as applications of the findings in education.

To identify relevant articles, several databases were searched, including Springer Link, IEEE Xplore, ERIC, ACM, ScienceDirect, the Journal of Educational Data Mining (JEDM), and Google Scholar. The search process began by using Google Scholar to broadly locate articles across numerous scientific journals. Additional targeted searches were then conducted using the search tools within each database. To ensure a comprehensive search, keywords from pertinent articles were analyzed to identify alternative spellings and terms related to the main

search keywords. The key search terms used were: **(educational data mining**) AND (**tertiary education OR secondary education**) AND (**online/e-learning**). The time period searched was 2015-2019 with the addition of early 2020 publications.

Following Kitchenham's (2010) methodology, inclusion and exclusion criteria were developed as follows:

Inclusion criteria:

- Works predicting student academic performance or dropout in secondary, tertiary, and related online education using data mining techniques.
- Papers published in scientific journals or conference proceedings.
- Works providing detailed methodology.

Exclusion criteria:

- Works not using data mining to predict academic performance or dropout.
- Works not detailing the attributes used.
- Works not detailing the algorithms used.
- Works not providing evaluation details of the algorithms.

The paper identified **125** research findings after applying the inclusion and exclusion techniques on **520** papers. For the purpose of the study, **30** papers were unique to the problem domain.

Table 1 summarizes the frequencies of the measurement tools used. The accuracy was the most used evaluation tool in the papers collated for the purpose of the course work.

Table 1 Summary of literature review on techniques in educational data mining

Method	Count	Accuracy	Papers
Neural Networks	10	**79**	Adekitan and Salau (2019), 7, Altai et al. (2019), Amirhajlou et al. (2019), Asif et al. (2017), Bergin et al. (2015), Burgos et al. (2018), Chanlekha and Niramitranon (2018), Costa et al. (2017), Francis and Babu (2019), Guo et al. (2016), Hasheminejad and Sarvmili (2018), Hassan et al. (2019), Huang et al. (2019), Hussain et al. (2018a),

Association Rule	2	**89**	Hussain et al. (2018), S. Abu-Oda and M. El-Halees (2015) Saravanan and Jyothi (2019),
Support Vector Machine	12	**79**	Al-Shehri et al. (2017), Amirhajlou et al. (2019), Angiani et al. (2019), Bergin et al. (2015), Bydzovska (2016), Chanlekha and Niramitranon (2018), Corrigan et al. (2015), Costa et al. (2017), Daud et al. (2017), Francis and Babu (2019), GonzalezMarcos et al. (2019), Guo et al. (2016), Hasheminejad and Sarvmili (2018), Hassan et al. (2019), Huang et al. (2019), Kamal and Ahuja (2019), Kumar et al. (2019), Livieris et al. (2016),
Clustering (K-means)	7	**80**	Mahboob et al. (2017), Migueis et al. (2018), Moscoso-Zea et al. (2019), Pandey and Taruna (2016), Polyzou and Karypis (2018), Satyanarayana and Nuckowski (2016), Shrivas and Tiwari (2017), Zhang et al. (2018)
Clustering (DBSCAN)	5	**78**	Livieris et al. (2019b), Livieris et al. (2019c) Livieris et al. (2019a), Mahboob et al. (2017), Mhetre and Nagar (2017), Miguelis et al. (2018), MoscosoZea et al. (2019), Namomsa and Sharma (2018), Pandey and Taruna (2016), Santoso and Yulia (2019), Saravanan and Jyothi (2019), Satyanarayana and Nuckowski (2016), Shrivas and Tiwari (2017), Singh and Kaur (2017), Strecht et al. (2015), Yehuala (2015), Zhou et al. (2015).
Logistic Regression	8	**86**	Adekitan and Salau (2019), 7, Bergin et al. (2015), Burgos et al. (2018), Canagareddy et al. (2019), Daud et al. (2017).
Decision Tree	10	**88**	Mhetre and Nagar (2017), Miguelis et al. (2018), Moscoso-Zea et al. (2019), Namomsa and Sharma (2018), Nunez et al. (2018), Pandey and Taruna (2016), Parmar et al. (2015), Polyzou and Karypis (2018), Rodrigues et al... (2019), Ruby and David (2015), Santoso and Yulia (2019), Sara et al. (2015), Saravanan and Jyothi (2019).

From the table we found out that decision trees were tested the most and had the highest accuracy, while association rule mining was tested the least but had the second highest accuracy. The other methods were tested a similar number of times and achieved accuracies between 78% and 86%.

4.0 DATASET DESCRIPTION

The dataset was obtained from Kaggle as provided in the link. The data set is a student adaptability data set. https://www.kaggle.com/datasets/mdmahmudulhasansuzan/students-adaptability-level-in-online-education it contains 1508 rows and 14 columns. The authors of the data set wereNishat Ahmed Samrin and Md. Aktaruzzaman Pramanik. The data set was collected to predict the outcome of Students' Adaptability Level Prediction in Online Education using Machine Learning Approaches with adaptability level as the target feature.

Table 2 Summary of dataset

Features	**Description**	**Attribute Type**
Gender	Male or Female.	Nominal
Age	Age of the students.	Interval
Education Level	The educational level of the students.	Nominal
Institution Type	Whether institution belong to Government or not.	Nominal
IT Student	Whether the student is studying in the field of Information Technology (IT) or not.	Nominal
Location	Whether the student is in a town or not.	Nominal
Load-shedding	The level of load shedding students experience.	Nominal
Financial Condition	The financial condition of the students.	Nominal
Internet Type	The type of internet connection the student has access to.	Nominal
Network Type	The type of network the students use for online education.	Nominal
Class Duration	The duration of online classes attended by the students.	Ordinal
Self LMS	Whether the students have Learning Management System (LMS).	Nominal
Device	The device used by the students for online education.	Nominal
Adaptivity Level	To detect the adaptivity level for online learning.	Nominal

4.1 DATASET BEFORE PRE-PROCESSING

R studio was used to examine the dataset for any missing values, revealing the presence of no null values. The dataset encompassed both nominal and ordinal values.

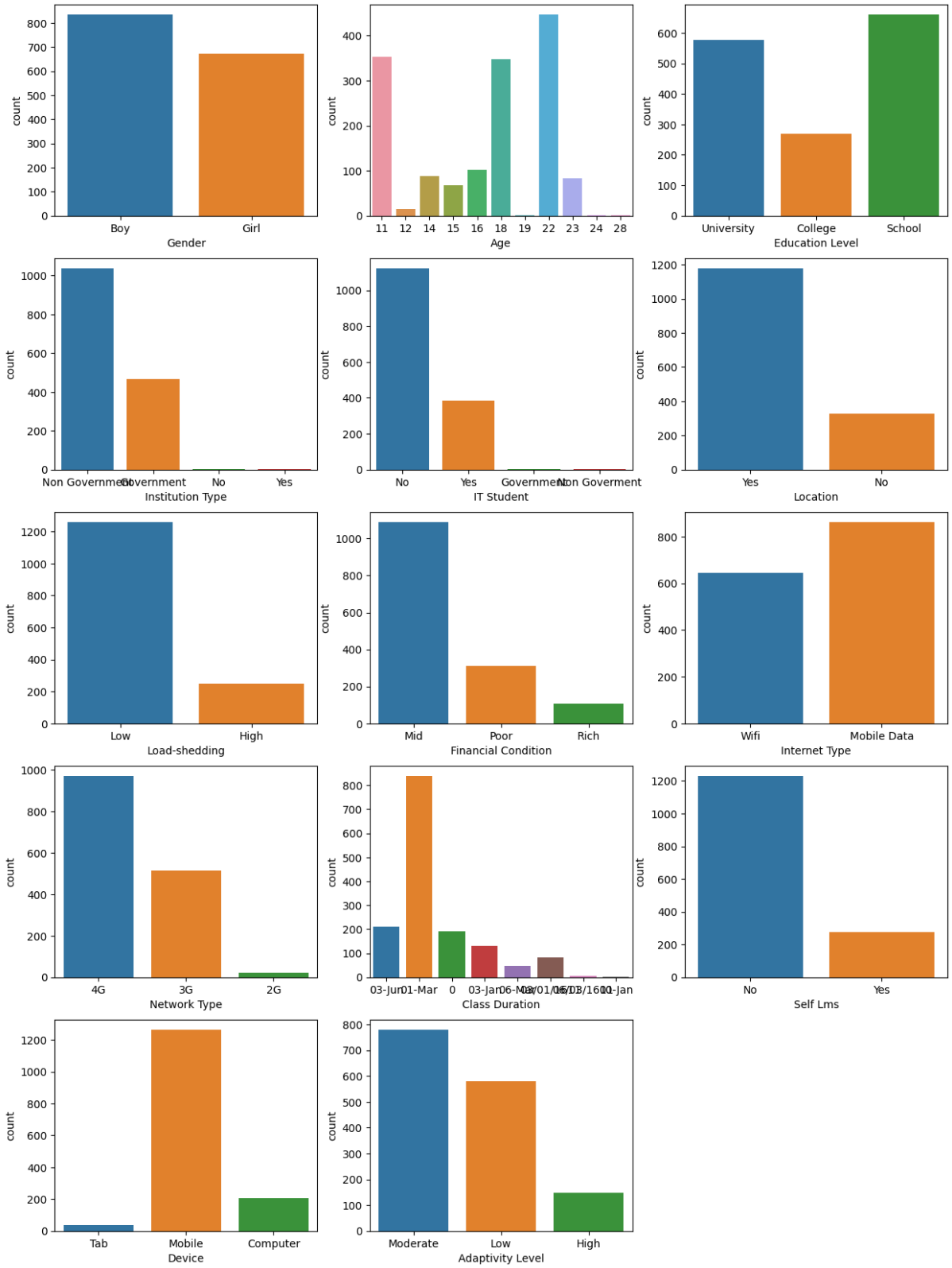

Figure 2 Exploratory data analysis before preprocessing

4.2 DATASET AFTER PRE-PROCESSING

The following steps were accrued out for preprocessing:

- Class Duration column was dropped as the feature provides no relationship to the analysis
- The data set was standardized
- The data was scaled.
- The data type was converted to numerical variables for analysis.

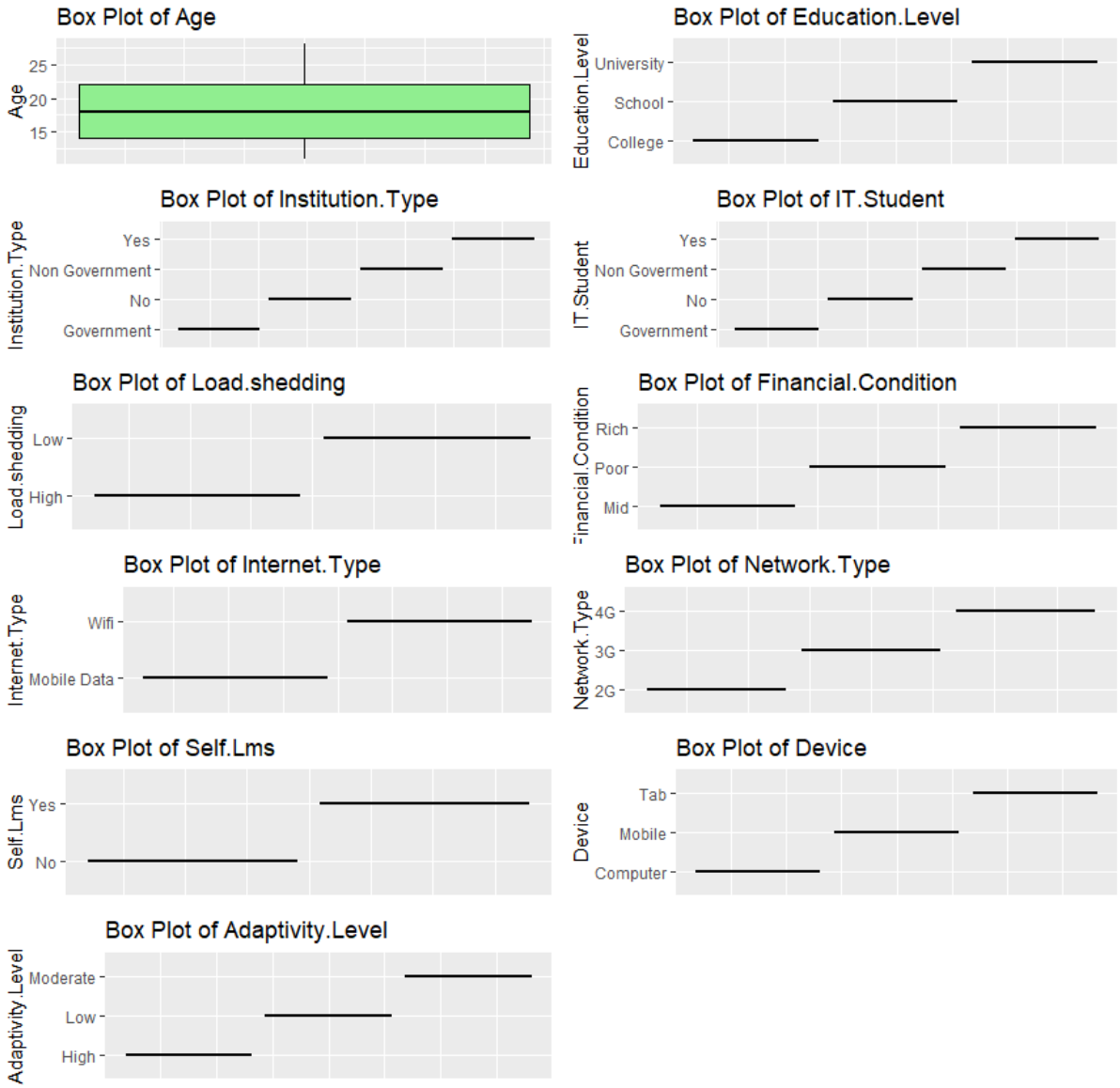

Figure 3 Box summary after pre-processing

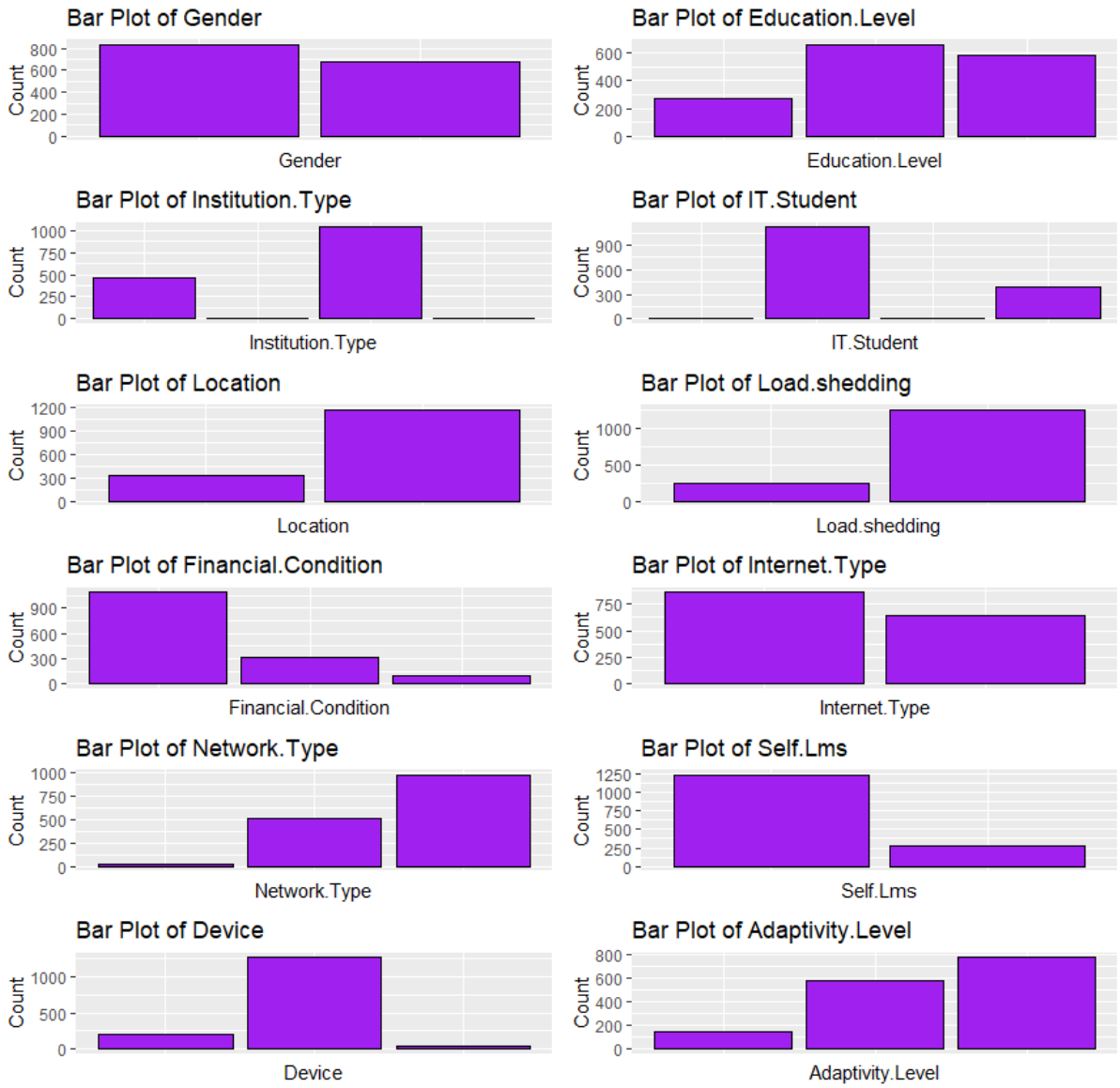

Figure 4 Exploratory Data Analysis after pre-processing

5.0 METHODOLOGY

In this paper three algorithms were employed to discover hidden patterns within the dataset.

1. **Clustering**
   - Association rule mining
2. **Classification**
   - Naïve bayes
   - Random forest
3. **Clustering**
   - Hierarchical clustering
   - Gausian mixture model
4. Deep Learning

5.1 WORK FLOW DIAGRAM

The work flow diagram was to analyse of the dataset.

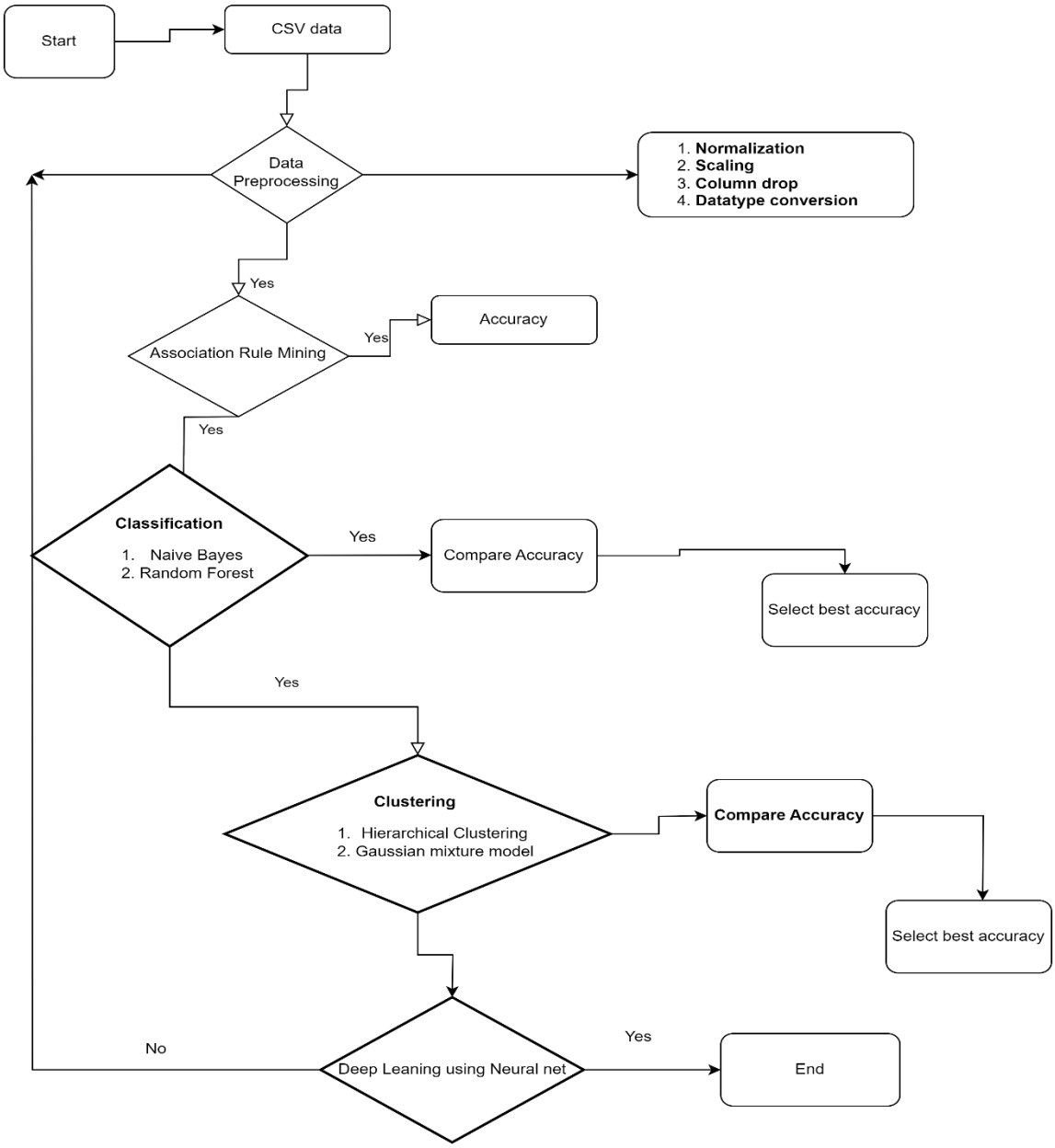

Figure 5 Work Flow Diagram

5.1 Association Rule Mining

Agrawal, R., Imieliński, T., & Swami, A. (1993) defined association rule mining is an exploratory data mining technique used to uncover relationships between variables in large datasets. It is commonly used in market basket analysis to identify combinations of products that customers frequently purchase together. The goal is to find "associations" between items where one item implies another, like "customers who buy item A also tend to buy item B".

These relationships are generated in the form of if-then rules, with the if-part called the antecedent (left-hand side of the rule) and the then-part called the consequent (right-hand side). An example rule could be "If a customer purchases diapers, they are likely to also purchase baby wipes". Each rule contains two key metrics - support and confidence. Support indicates how frequently the items appear together, while confidence indicates the strength of the rule.

By surfacing the hidden associations in large transactional datasets, retailers can improve sales through marketing campaigns, product placements, and recommendations. Overall, association rule mining is a key technique for uncovering co-occurrence relationships between variables in a dataset.

5.2 Naive Bayes

Naive Bayesian classifiers are a type of fast, supervised machine learning algorithm well-suited for large-scale classification and prediction problems. They rely on two key assumptions: first, that numeric attribute values follow a normal distribution within each class, and second, that each attribute value is conditionally independent of other attributes for a given class value. Based on these assumptions, the classifier calculates the posterior probability for each class and predicts that a new sample belongs to the class with the highest probability. A major advantage of naive Bayes is its efficiency in learning and prediction, enabling it to handle datasets with many attributes and large numbers of samples. However, the independence assumption is often an oversimplification, which can limit classification accuracy for some complex real-world datasets.

In summary, naive Bayes classifiers make predictions using Bayes theorem and assume independence between input attributes. Their speed and scalability make them useful for large classification tasks, but their simplicity can lead to limitations in predictive accuracy.

The educational field has extensively employed the Naïve Bayes classifier for prediction and classification purposes, as demonstrated by various studies Pandey and Taruna (2016), Huang et al. (2019).

5.3 Random Forest

Random forest is an ensemble supervised learning method made up of multiple decision trees. It operates by constructing a large number of decision trees during training and outputting a prediction that is the mode of the predictions from individual trees.

Decision trees are prone to high variance, meaning a small change in the training data can result in a very different tree structure. To overcome this, random forest introduces randomness in two ways during the training phase:

- Bootstrap aggregating (bagging): random sampling with replacement is used to generate multiple different training datasets for each tree.
- Random subsets of features are selected at each split when building the decision trees.

The randomization and ensemble approach helps reduce overfitting and improves the overall predictive performance compared to a single decision tree model. The aggregation of predictions averages out the noise and variance of individual trees. Overall, random forests are very effective for a variety of regression and classification problems.

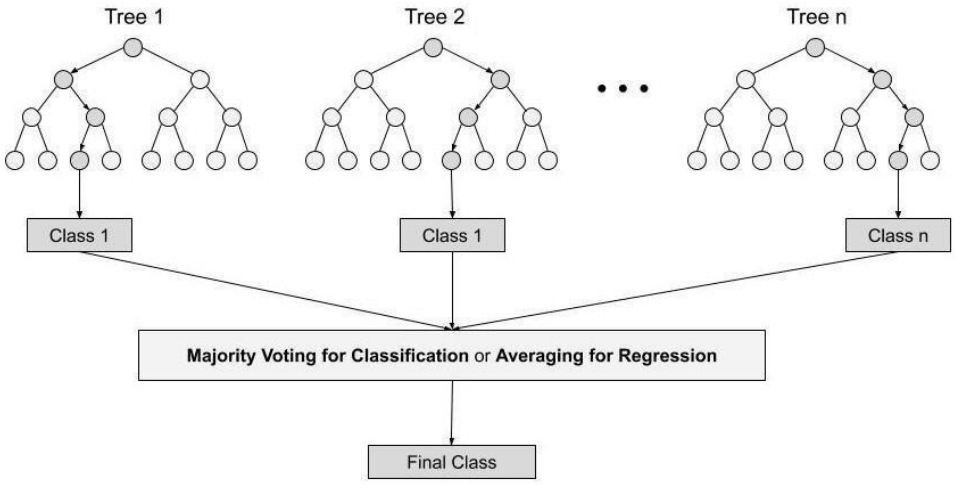

Figure 6 Random Forest Architecture (Yiu, 2019).

5.4 Hierarchical Clustering

Hierarchical clustering is an unsupervised machine learning technique that builds a hierarchy of clusters by either agglomerative (bottom-up) or divisive (top-down) approaches. It does not require the number of clusters to be pre-specified. The agglomerative approach starts with each data point in its own cluster and merges the most similar pairs of clusters successively to build a hierarchy. The divisive approach starts with all data points in one cluster and splits the clusters recursively. Distance metrics like Euclidean distance are used to determine similarity.

The result is represented in a dendrogram. Hierarchical clustering has applications in data mining and bioinformatics (Johnson, 1967; Jain & Dubes, 1988; Xu & Wunsch, 2005).

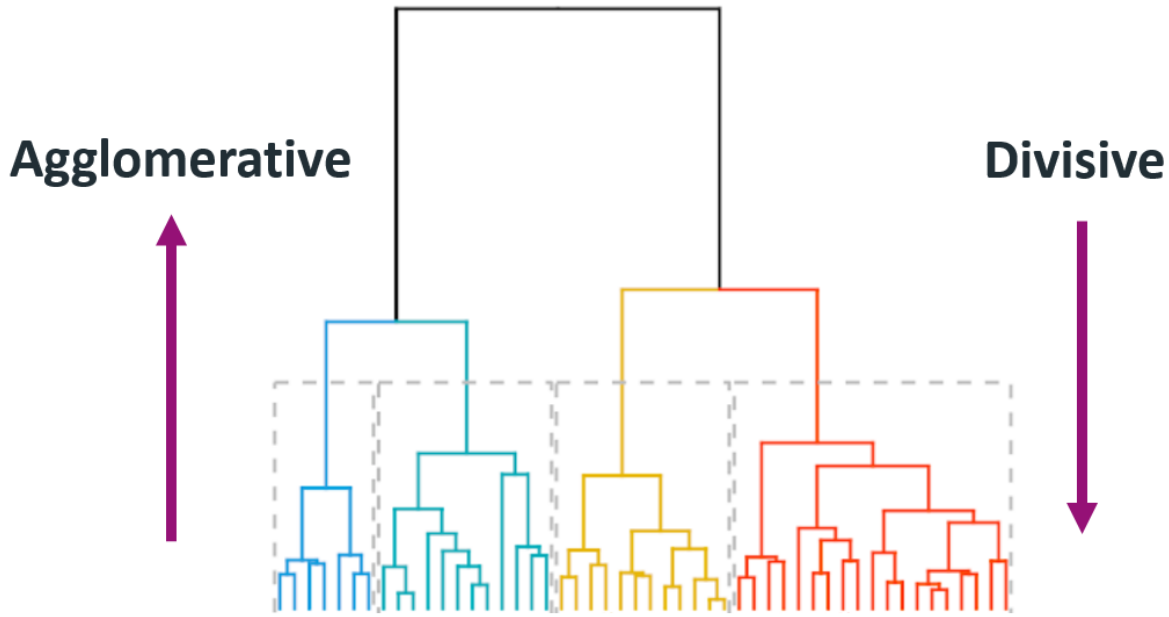

Figure 7 hierarchical clustering (Himashu Sharma, 2021)

5.5 Gaussian mixture model (GMM)

A Gaussian mixture model (GMM) is a probabilistic model that assumes data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. GMMs are used for clustering and density estimation. The Expectation-Maximization (EM) algorithm is commonly used to iteratively estimate the model parameters like the mixing coefficients, means, and standard deviations of each component Gaussian distribution. GMMs can model complex distributions and have applications in machine learning for statistical modelling, classification, and pattern recognition. They are a generalization of k-means clustering. GMMs are flexible but can be prone to overfitting (Reynolds, 2009; Bishop, 2006; McLachlan & Peel, 2000).

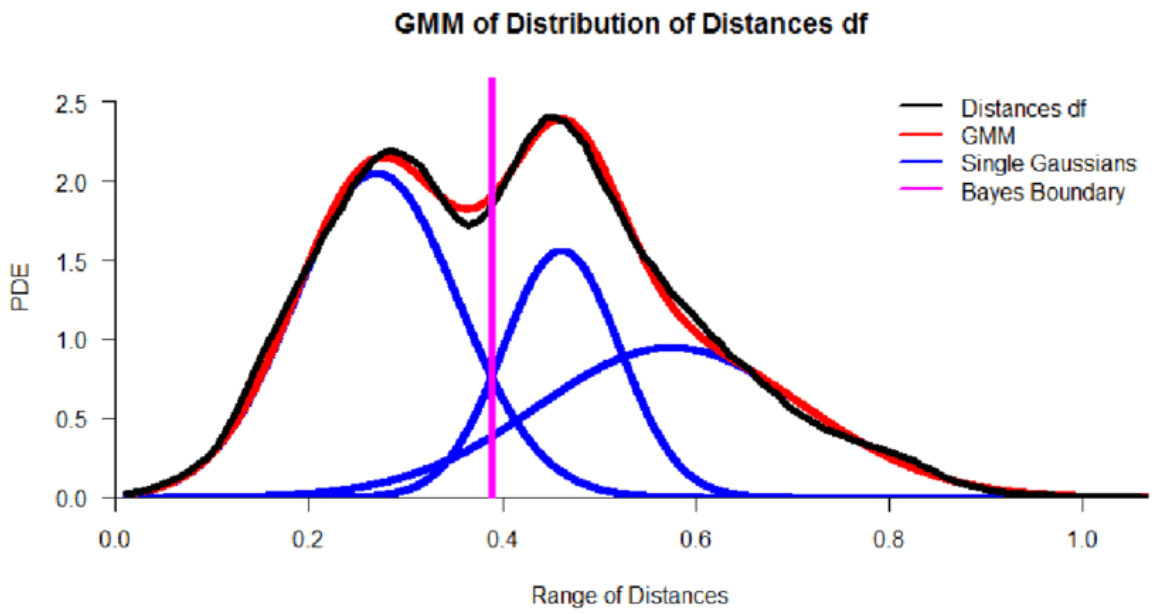

Figure 8 Gausian Mixture Model

5.6 Deep Learning

Deep learning is a class of machine learning algorithms that use multiple layers of non-linear processing units called neural networks for feature extraction and transformation. Each layer transforms the representation at the previous layer into a more abstract and composite representation. The layers between the input and output layers are called hidden layers. Deep neural networks can model complex non-linear relationships and have outperformed other machine learning techniques in applications like computer vision, speech recognition, and natural language processing. Key to their success is the availability of big data and GPU computing power for training (LeCun et al., 2015; Goodfellow et al., 2016; Schmid Huber, 2015).

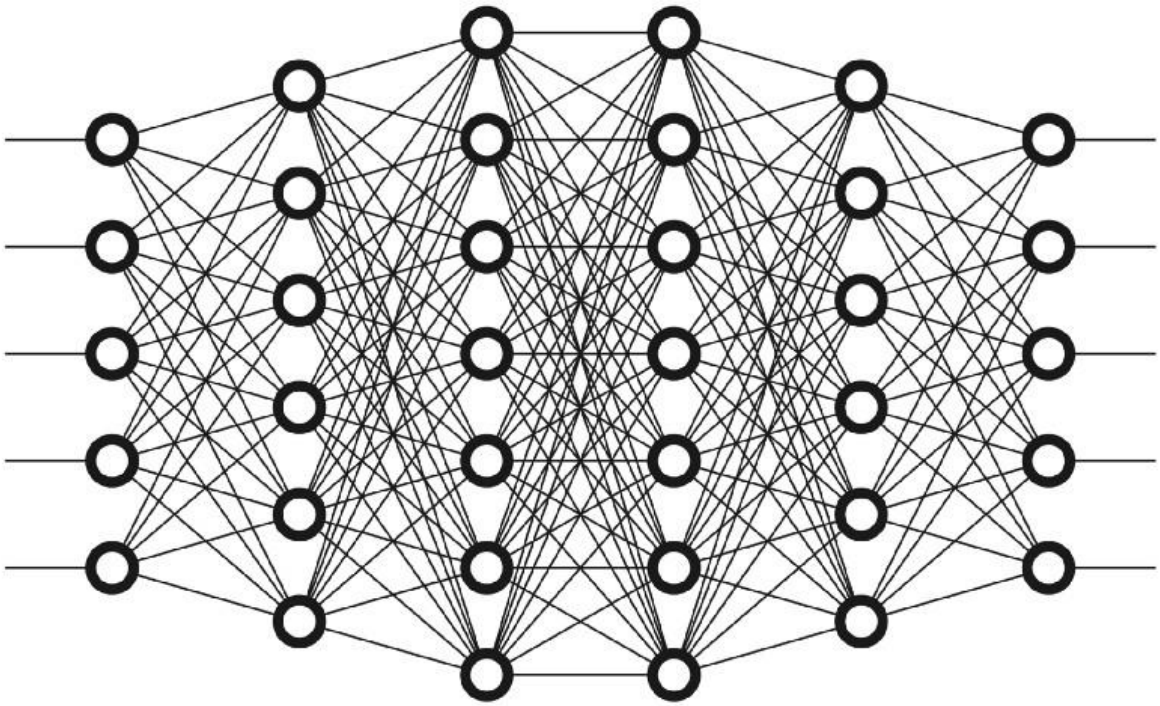

Figure 9 Deep Learning

6.0 ANALYSIS & RESULTS

6.1 Association rule mining

The categorical variables were encoded as factors. Age was discretized into bins. Transaction data was created with each row as a transaction.

Apriori algorithm was used to generate association rules between student attributes and adaptability level with minimum support of 65% and confidence of 85%.

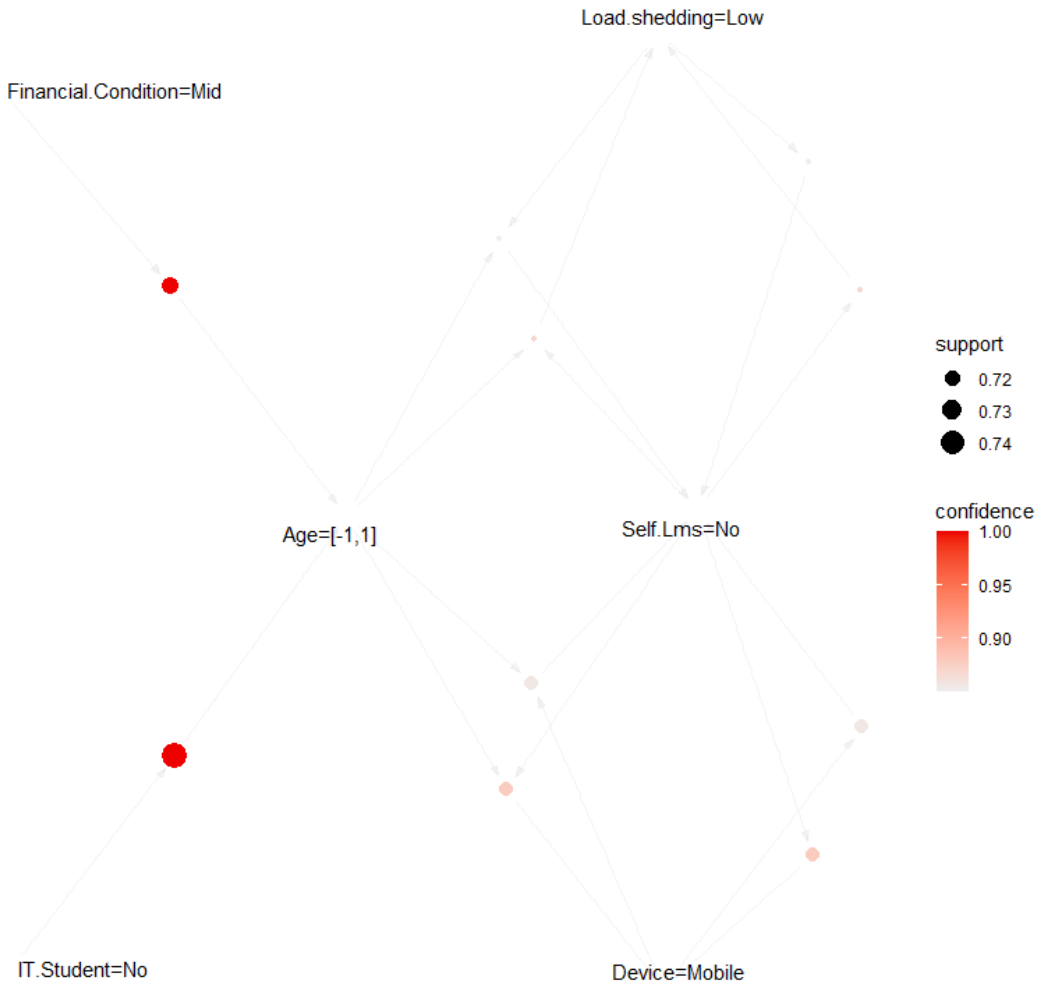

Figure 10 Association rule mining report

The apriori algorithm generated 17 rules and produced a confidence of 95% with a support of 6%. Overall, this association rule highlights that better financial condition, institution, device has a positive correlation with higher adaptability level for students in online education.

6.2 Naïve Bayes

With the dataset split into 70 % training and 30% test data, the algorithm produced a prediction accuracy of 42.3%.

Figure 10 below shows the confusion matrix of the naïve bayesian algorithm on the data set.

```
Confusion Matrix and Statistics

          y_test
preds      High Low Moderate
  High       19   1        6
  Low        25 144      186
  Moderate    8  24       39

Overall Statistics

               Accuracy : 0.4469
                 95% CI : (0.4004, 0.4941)
    No Information Rate : 0.5111
    P-Value [Acc > NIR] : 0.9973

                  Kappa : 0.1071

 Mcnemar's Test P-Value : <2e-16

Statistics by Class:

                     Class: High Class: Low Class: Moderate
Sensitivity              0.36538     0.8521          0.16883
Specificity              0.98250     0.2544          0.85520
Pos Pred Value           0.73077     0.4056          0.54930
Neg Pred Value           0.92254     0.7423          0.49606
Prevalence               0.11504     0.3739          0.51106
Detection Rate           0.04204     0.3186          0.08628
Detection Prevalence     0.05752     0.7854          0.15708
Balanced Accuracy        0.67394     0.5532          0.51202
```

Figure 11 Confusion matrix and statistics of the naive bayes algorithm

6.3 Random Forest

With the dataset divided into a 70:30 train-test dataset, Thus, a K-fold cross validation model of random forest is applied to the whole dataset to obtain the best value, use it to train the training dataset, and predict using the test dataset. Two n-tree values was used to test for the error rates to determine the best number of trees on the dataset. **Table 3** and **Table 4** gives the summary statistics of the confusion matrix.

Type of random forest: classification
Number of trees: 300
No. of variables tried at each split: 3
OOB estimate of error rate: **17.57%**

Table 3 Confusion matrix for n300

Confusion matrix	1	2	3	Class error
1	88	10	51	0.4093960
2	0	482	97	0.1675302
3	15	92	673	0.1371795

Type of random forest: classification
Number of trees: 500
No. of variables tried at each split: 3
OOB estimate of error rate: **17.31%**

Table 4 Confusion matrix for n300

Confusion matrix	1	2	3	Class error
1	84	11	54	0.4362416
2	3	479	97	0.1727116
3	14	82	684	0.1230769

The accuracy of 83.7% suggests a relatively high-performing model.

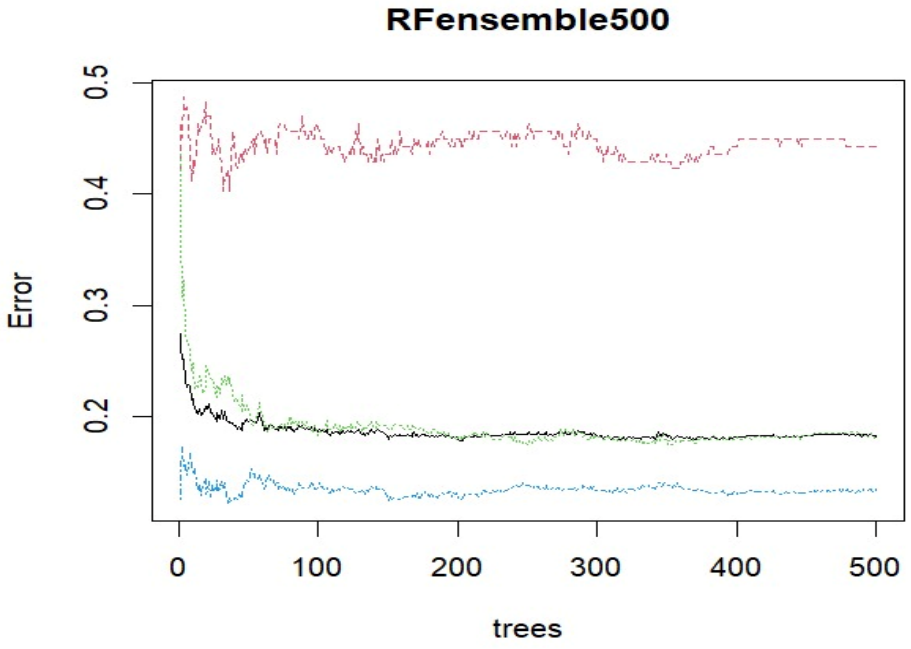

Figure 12 Random Forest error plot with K-fold Cross Validation

Figure 11 displays the error rates associated with different numbers of trees created for the random forest model. The black line in the middle of the graph represents the average error rate for predicting both the binary and continuous target variables. As more trees are added to the model, this average error rate decreases and then stabilizes, indicating that additional trees beyond a certain point do not significantly improve performance. Overall, the figure demonstrates how random forest model accuracy generally improves with more with exponential increase in its plateau.

Table 5 Mean Decreasing GINI

Gender	35.58313
Age	26.92246
Educational level	42.67841
Institution type	49.15945
IT. student	24.41488
Location	37.62417
Load shedding	24.12313
Financial condition	69.99551
Internet type	35.40063
Network type	34.91658
Self LMS	27.25256
Device	23.91363

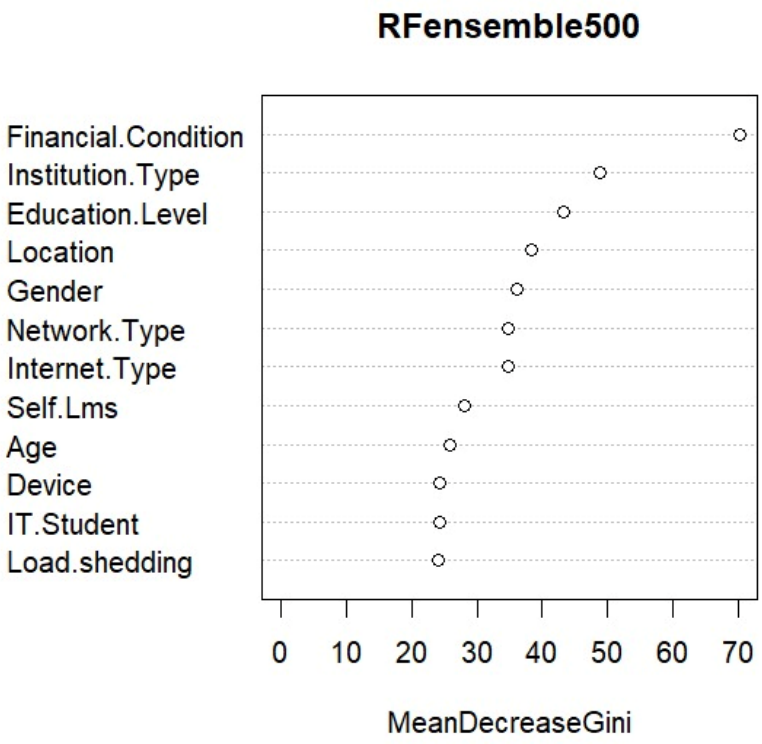

Figure 13Variable Importance based on Mean Decrease Gini in Random Forest model

Figure 12 illustrates the variable importance scores from the Random Forest model as measured by Mean Decrease in Gini index. The variable "financial condition" had the highest importance score in this model. However, when Random Forest was run with K-fold cross-validation, the variable "institution type" had the highest importance score instead.

6.5 Hierarchical Clustering

Hierarchical clustering was utilized to examine the dataset. Relevant features were chosen and transformed into numerical formats to get the data ready for cluster analysis. Any missing numerical values were replaced with column averages. A distance matrix was calculated between data points, then hierarchical clustering was applied on this matrix using average linkage. The dendrogram was plotted after cutting the hierarchical tree into 3 clusters to visualize the grouping structure.

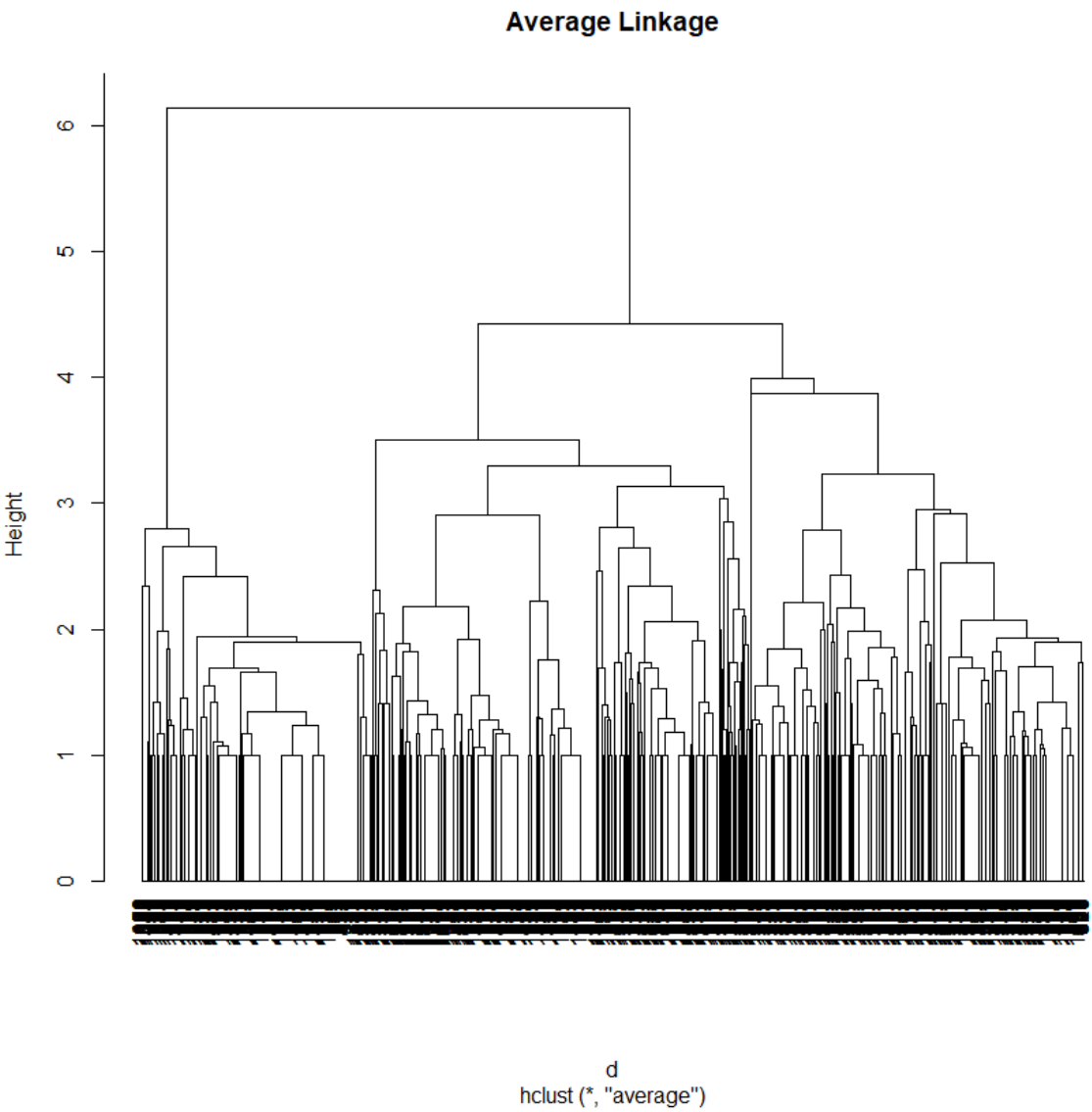

Figure 14 H-Clustering Dendrogram by average linkage

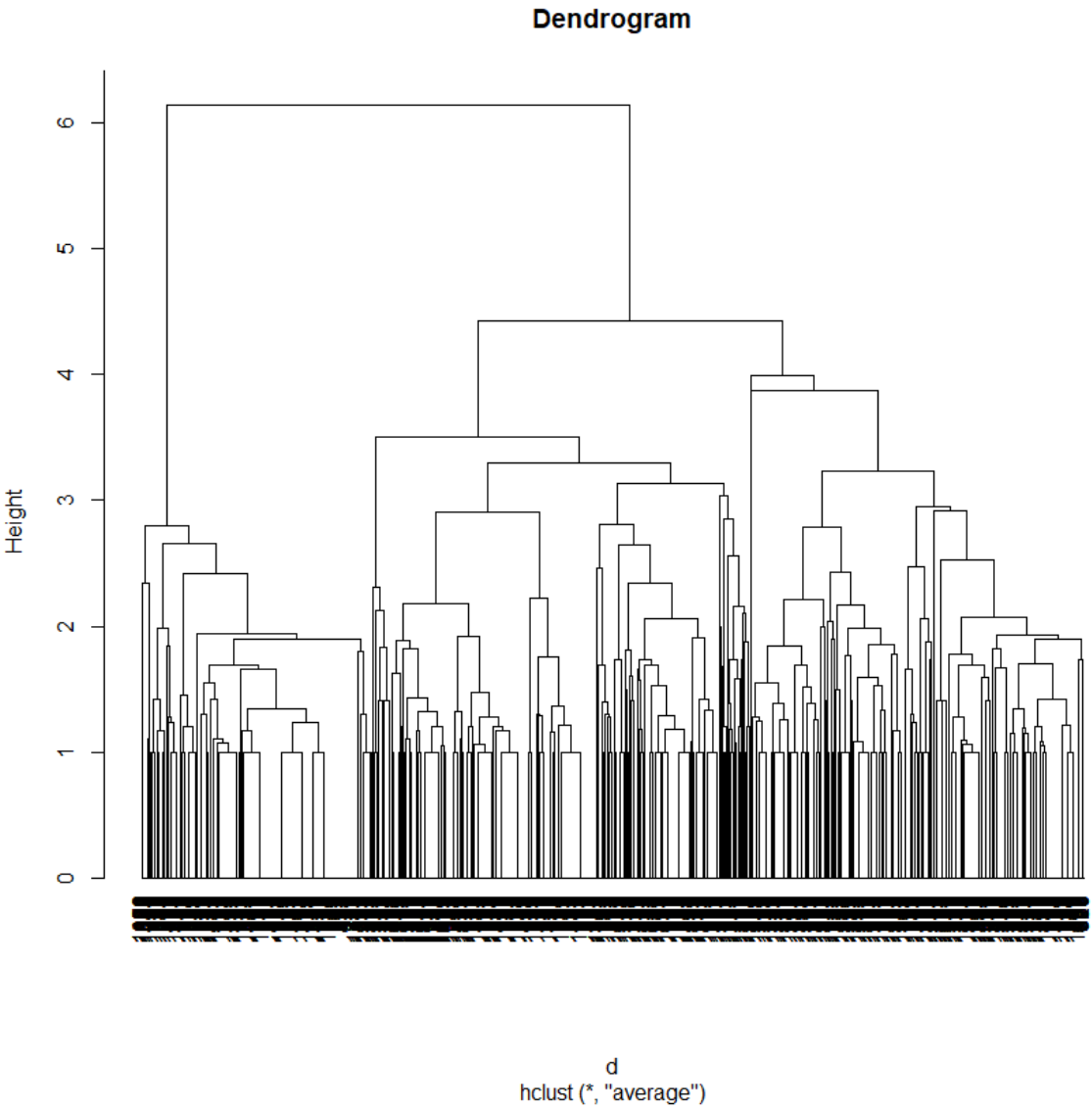

Figure 15 H-clustering Dendrogram

From Figure 14 and Figure 15, The clusters in figure 16 produces the best accuracy using the Gaussian mixture model. The table below illustrates the accuracy and number of clusters

Table 6 Clustering Algorithm comparison

Clustering Algorithm	**Number of clusters**	**Accuracy**
Hierarchical Clustering	3	85
Gaussian mixture model	10	87

6.6 Gausian Mixture Model

The dataset columns were converted to numeric factors to prepare the data for clustering analysis. Numeric columns were extracted and missing values are extrapolated using column means. Gaussian mixture models (GMMs) were fitted to the pre-processed data with varying cluster counts from 1 to 10. For each GMM, cluster assignments are extracted and silhouette scores computed to evaluate clustering quality. The Bayesian Information Criterion (BIC) values for each GMM are plotted to assess model complexity and fit trade-offs. A plot of silhouette scores versus number of clusters is generated to determine the optimal clusters based on separation. The BIC plot provides insights on suitable cluster counts based on model criteria while the silhouette plot indicates the ideal clusters based on data separation.

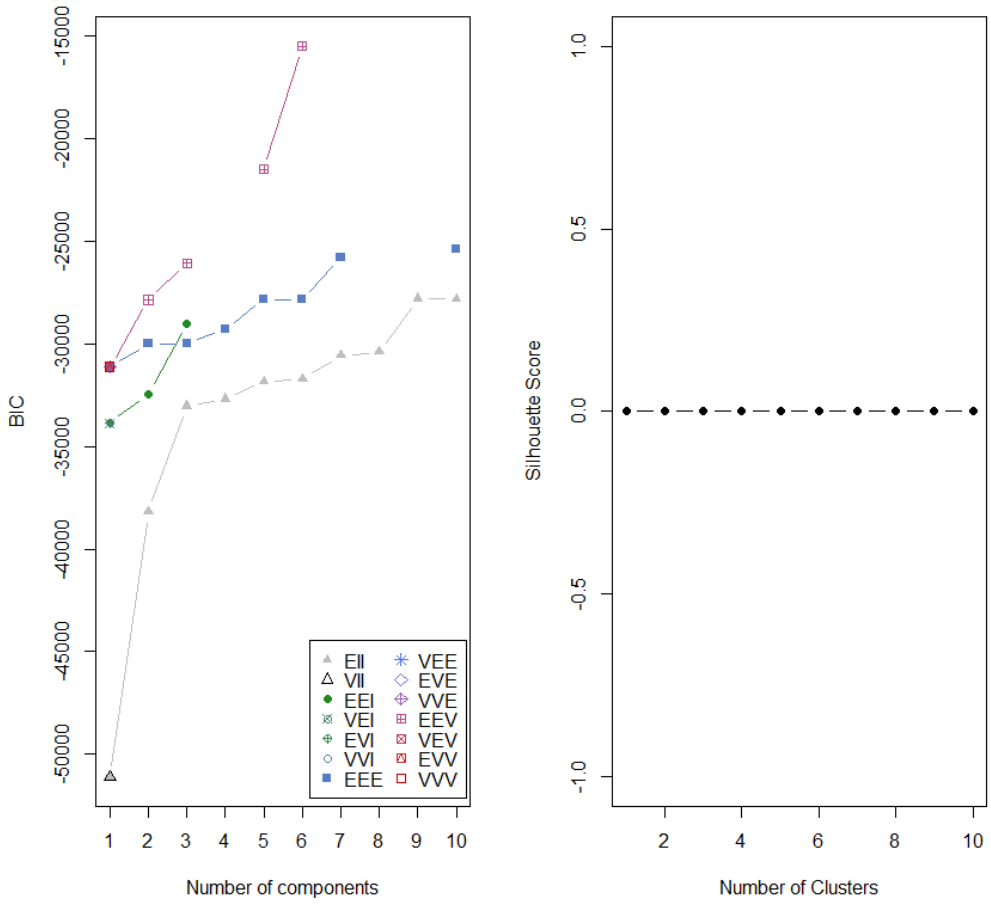

Figure 16 Gausian Mixture Model Cluster result

6.7 Deep Learning

The input and target variable data were prepared for training the neural network model. Specifically, the target variable 'y' representing the adaptability level of learners is extracted from the dataset. The categorical values "Low" and "High" in 'y' are replaced with 0 and 1 respectively, and converted to a numeric vector. The input matrix 'x' is constructed from selected columns Gender, Institution. Type, IT. Student, Location, Load. Shedding, Financial. Condition, Internet. Type, Self.Lms and Device, using feature selection to the data frame, and converted to a numeric matrix with 10 columns. This prepares the input and target data in appropriate formats for model training.

A neural network model 'nn' was created using the neural net function, with the formula specifying the input variables and 5 hidden layer neurons. The prepared input data frame is passed to the data argument to train the model. Model predictions are generated and further transformed into binary values based on a threshold. A confusion matrix is constructed to evaluate model performance by comparing to the actual target values.

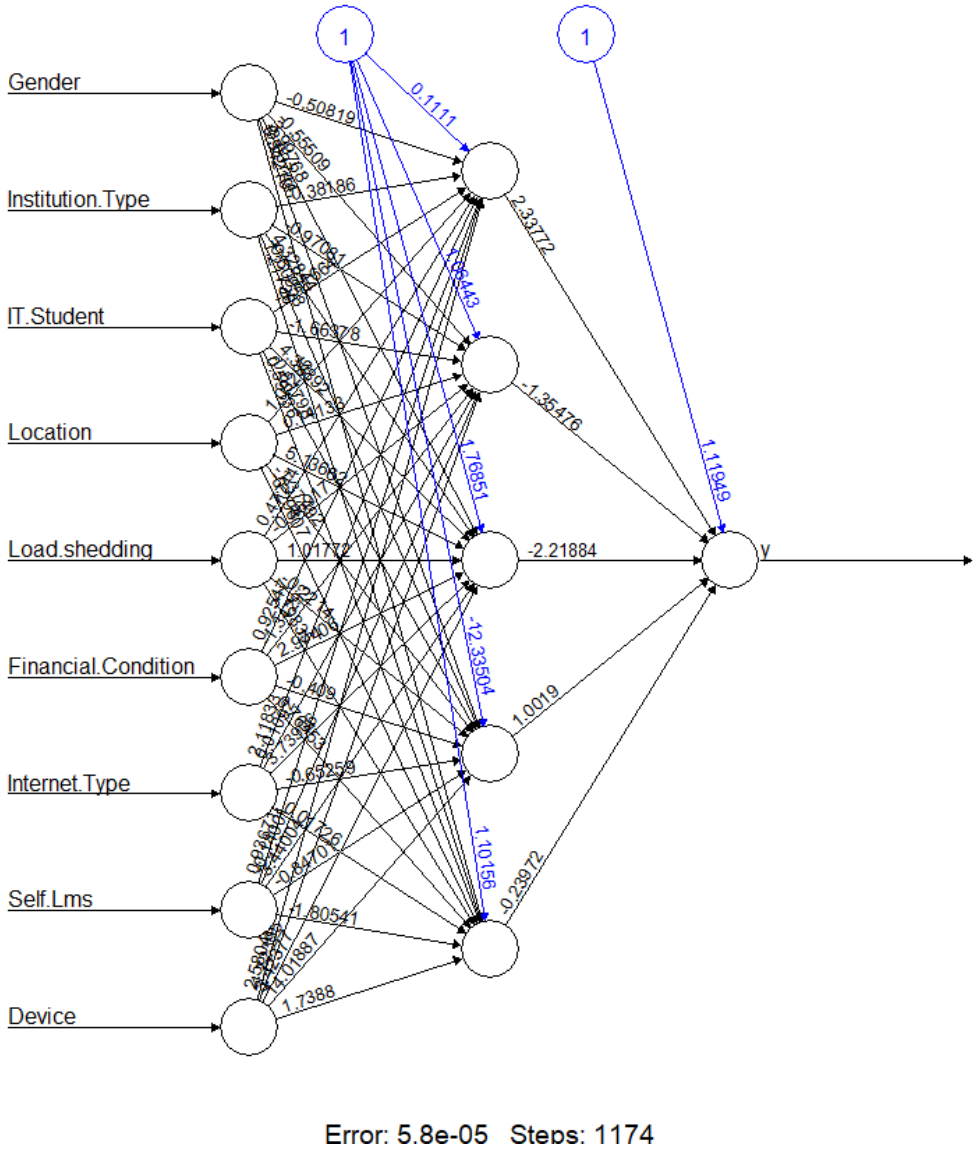

Figure 17 Deep Learning with Neural net

The calculated value from the yhat table was constructed and the accuracy of the model was 99%. The implication of the finding reveals that with certain features in play the student's performance and adaptability level increases exponentially compared the other models and algorithms.

Table 7 Yhat statistics table

Y	YHAT	
	0	1
1	244	0
2	0	1264

7.0 DISCUSSIONS AND CONCLUSIONS

To verify our problem statement, we explored six (6) data mining techniques and chose the accuracy score as the evaluation metric.

Table 8 Data Mining Technique Accuracy Comparison

Data mining technique	Accuracy (%)
Association Rule mining	**82**
Naïve Bayes	**42.3**
Random Forest	**83.7**
Hierarchical Clustering	**84**
Gausian Mixture Model	**87**
Deep Learning	**99**

The results from **table 7** indicates the deep learning algorithm, clustering algorithm, Random Forest and association rule mining produced the best accuracy scores with a mean of 79.6%. The deep learning algorithm had the best accuracy score of 99%, while naïve bayes had the lowest accuracy score. As the amount of training data increases, it is expected that these accuracy scores will improve. However, from the literature review naïve bayes predicts other factors in educational mining with top notch accuracy.

Further research needs to be carried out in the dataset to reveal other insights within the data. The findings of the research would play a key role in policy and decision making in the educational sector which will in turn enhance efficient resource allocation.

REFERENCE

Adekitan, A. I., & Salau, O. (2019). The impact of engineering students' performance in the first three years on their graduation result using educational data mining. Heliyon, 5 (2019), e01250,. doi: 10.1016/j.heliyon.2019.e01250

Agrawal, R., Imieliński, T. and Swami, A., 1993, June. Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD international conference on Management of data (pp. 207-216).

Al-Shehri, H. et al. (2017) 'Student performance prediction using Support Vector Machine and K-Nearest Neighbor', in, pp. 1–4. doi: 10.1109/CCECE.2017.7946847.

Amirhajlou, L. et al. (2019) 'Application of data mining techniques for predicting residents' performance on pre-board examinations: A case study', Journal of education and health promotion, 8, p. 108. doi: 10.4103/jehp.jehp_394_18.

Asif, R. et al. (2017) 'Analyzing undergraduate students' performance using educational data mining', Computers & Education, 113, pp. 177–194. doi: 10.1016/j.compedu.2017.05.007.

Baker, R. S. J. d. and Yacef, K. (2009) 'The State of Educational Data Mining in 2009: A Review and Future Visions', Journal of Educational Data Mining, 1(1), pp. 3–17. doi: 10.5281/zenodo.3554657.

Bergin, S. et al. (2015) 'Using Machine Learning Techniques to Predict Introductory Programming Performance', International Journal of Computer Science and Software Engineering (IJCSSE). International Journal of Computer Science and Software Engineering, 4(12), pp. 323–328. Available at: https://mural.maynoothuniversity.ie/8682/.

Burgos, C., Campanario, M. L., de la Peña, D., Lara, J. A., Lizcano, D., & Martínez, M. A. (2018). Data mining for modeling students' performance: A tutoring action plan to prevent academic dropout. Computers& Electrical Engineering, 66, 541–556. doi: 10.1016/J.COMPELECENG.2017.03.005

Bydžovská, H., 2016. A Comparative Analysis of Techniques for Predicting Student Performance. International Educational Data Mining Society.

Canagareddy, D., Subarayadu, K., & Hurbungs, V. (2019). A machine learning model to predict the performance of university students. Smart and Sustainable Engineering for Next Generation Applications (pp. 313–322). Springer International Publishing. doi: 10.1007/978-3-030-18240-3_29

Chanlekha, H. and Niramitranon, J. (2018) 'Student performance prediction model for early-identification of at-risk students in traditional classroom settings', Proceedings of the 10th International Conference on Management of Digital EcoSystems. Available at: https://api.semanticscholar.org/CorpusID:53243782.

Corrigan, O., Smeaton, A.F., Glynn, M. and Smyth, S., 2015. Using educational analytics to improve test performance. In Design for Teaching and Learning in a Networked World: 10th European Conference on Technology Enhanced Learning, EC-TEL 2015, Toledo, Spain, September 15-18, 2015, Proceedings 10 (pp. 42-55). Springer International Publishing.

Daud, A., Aljohani, N. R., Abbasi, R. A., Lytras, M. D., Abbas, F., & Alowibdi, J. S. (2017). Predicting student performance using advanced learning analytics. In Proceedings of the 26th International Conference on World Wide Web Companion. (pp. 415-421).ACM Press. doi: 10.1145/3041021.3054164

Del Río, C. and Insuasti, J.P., 2016. Predicting academic performance in traditional environments at higher-education institutions using data mining: A review. Ecos de la Academia-Universidad Técnica del Norte, 2(04), pp.185-201.

El-Seoud, S. et al. (2014) 'E-Learning and Students' Motivation: A Research Study on the Effect of E-Learning on Higher Education', International Journal of Emerging Technologies in Learning (iJET), 9, pp. 20–26. doi: 10.3991/ijet.v9i4.3465.

Francis, B.K. and Babu, S.S., 2019. Predicting academic performance of students using a hybrid data mining approach. Journal of medical systems, 43, pp.1-15.

González-Marcos, A., Olarte-Valentín, R., Meré, J.B.O. and Alba-Elías, F., 2019. Predicting Students' Performance in a Virtual Experience for Project Management Learning. In CSEDU (1) (pp. 665-673).

Goodfellow, I., Bengio, Y. and Courville, A., 2016. Deep learning. MIT press.

Hasheminejad, S. M. H. and Sarvmili, M. (2019) Journal of AI and Data Mining. Shahrood University of Technology, 7(1), pp. 77–96. doi: 10.22044/jadm.2018.5506.1662.

Huang, A. Y., Lu, O. H., Huang, J. C., Yin, C. J., & Yang, S. J. (2019). Predicting students' academic performance by using educational big data and learning analytics: evaluation of classification methods and learning logs. Interactive Learning Environments, 28(2), 206-230. doi: 10.1080/10494820.2019.1636086

Hussain, M., Hussain, S., Zhang, W., Zhu, W., Theodorou, P., & Abidi, S. M. R.. (2018). Mining moodle data to detect the inactive and low-performance students during the moodle course. Presented at the 2nd International Conference on Big Data Research - ICBDR 2018 (pp. 133-140). ACM Press. doi: 10.1145/3291801.3291828

Hussain, S., Dahan, N. A., Ba-Alwib, F. M., & Ribata, N. (2018). Educational data mining and analysis of students' academic performance using WEKA. Indonesian Journal of Electrical Engineering and Computer Science, 9(2), 447–459. doi: 10.11591/ ijeecs.v9.i2.pp447-459

Jain, A.K. and Dubes, R.C., 1988. Algorithms for clustering data. Prentice-Hall, Inc..

Johnson, S.C., 1967. Hierarchical clustering schemes. Psychometrika, 32(3), pp.241-254.

Kamal, P. and Ahuja, S., 2019. An ensemble-based model for prediction of academic performance of students in undergrad professional course. Journal of Engineering, Design and Technology, 17(4), pp.769-781.

Kitchenham, B. (2010) ‘What’s up with software metrics? – A preliminary mapping study’, Journal of Systems and Software, 83(1), pp. 37–51. doi: 10.1016/j.jss.2009.06.041.

Kumar, S.A. and Vijayalakshmi, M.N., 2011. Envison of students cconcert using supervised learning techniques. IJCES International Journal of Computer Engineering Science, 1(2).

LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. nature, 521(7553), pp.436-444.

Livieris, I. E., Drakopoulou, K., Mikropoulos, T. A., Tampakas, V., & Pintelas, P. (2018). An ensemble-based semi-supervised approach for predicting students' performance. In: Mikropoulos T. (Eds) Research on e-Learning and ICT in Education. (pp. 25-42). Springer Cham. doi: 10.1007/978-3-319-95059-4

Livieris, I. E., Drakopoulou, K., Tampakas, V. T., Mikropoulos, T. A., & Pintelas, P (2019). Predicting secondary school students' performance utilizing a semi-supervised learning approach. Journal of Educational Computing Research, 57(2), 448–470. doi: 10.1177/0735633117752614

Livieris, I. E., Kotsilieris, T., Tampakas, V., & Pintelas, P. (2019). Improving the evaluation process of students' performance utilizing a decision support software. Neural Computing and Applications, 31, 1683–1694. doi: 10.1007/s00521-018-3756-y

Livieris, I. E., Tampakas, V., Kiriakidou, N., Mikropoulos, T., & Pintelas, P. (2019). Forecasting students' performance using an ensemble ssl algorithm. Communications in Computer and Information Science (pp. 566–581). Springer. doi: 10.1007/978-3-030-20954-4_43

Livieris, I., Mikropoulos, T. and Pintelas, P., 2016. A decision support system for predicting students’ performance. Themes in Science and Technology Education, 9(1), pp.43-57.

Livieris, I., Mikropoulos, T., & Pintelas, P. (2016). A decision support system for predicting students’ performance. Themes in Science and Technology Education, 9(1), 43–57.

M. Saravanan, V.L. Jyothi: A NOVEL APPROACH FOR PERFORMANCE AND ESTIMATION OF STUDENTS BY USING HYBRID MODEL. IIOABJ, 2019; Vol.10 (2):9-12.

Mahboob, T., Irfan, S., & Karamat, A.. (2017). A machine learning approach for student assessment in E-learning using Quinlan's C4.5, Naive Bayes and Random Forest algorithms. Presented at the 19th International Multi-Topic Conference, INMIC 2016. (pp. 1 -8). doi: 10.1109/INMIC.2016.7840094

Mhetre, V., & Nagar, M. (2017). Classification based data mining algorithms to predict slow, average and fast learners in educational system using WEKA. Presented at the International

Conference on Computing Methodologies and Communication (ICCMC). IEEE. (pp. 475-479). doi: 10.1109/iccmc.2017.8282735

Miguéis, V. L., Freitas, A., Garcia, P. J., & Silva, A. (2018). Early segmentation of students according to their academic performance: A predictive modelling approach. Decision Support Systems, 115, 36–51. doi: 10.1016/j.dss.2018.09.001

Moscoso-Zea, O., Saa, P. and Luján-Mora, S., 2019. Evaluation of algorithms to predict graduation rate in higher education institutions by applying educational data mining. Australasian Journal of Engineering Education, 24(1), pp.4-13.

Namomsa, G., & Sharma, D. P. (2018). Comparative analytics and predictive modeling of student performance through data mining techniques. SSRN Electronic Journal. doi: 10.2139/ssrn.3395126

Nuñez, H., Maldonado, G., & Astudillo, C. A. (2018). Semi-supervised regression based on tree SOMs for predicting students performance. Presented at the 9th International Conference on Pattern Recognition Systems (ICPRS 2018). Institution of Engineering and Technology. (pp. 12-19). doi: 10.1049/cp.2018.1288

Pandey, M. and Taruna, S. (2016) 'Towards the integration of multiple classifier pertaining to the Student's performance prediction', Perspectives in Science, 8, pp. 364–366. doi: 10.1016/j.pisc.2016.04.076.

Papamitsiou, Z. and Economides, A.A., 2014. Learning analytics and educational data mining in practice: A systematic literature review of empirical evidence. Journal of Educational Technology & Society, 17(4), pp.49-64.

Parmar, K., Vaghela, D., & Sharma, P. (2015). Performance prediction of students using distributed data mining. Presented at the: International Conference on Innovations in Information, Embedded and Communication Systems 2015 (ICIIECS), (pp. 1–5). doi: 10.1109/ICIIECS.2015.7192860

Peña-Ayala, A. (2014) 'Educational data mining: A survey and a data mining-based analysis of recent works', Expert Systems with Applications, 41(4, Part 1), pp. 1432–1462. doi: 10.1016/j.eswa.2013.08.042.

Polyzou, A. and Karypis, G., 2018. Feature Extraction for Classifying Students Based on Their Academic Performance. International Educational Data Mining Society.

Reynolds, D.A., 2009. Gaussian mixture models. Encyclopedia of biometrics, 741(659-663).

Romero, C., & Ventura, S. (2010). Educational data mining: A review of the state of the art. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 40 (6), 601–618. doi: 10.1109/TSMCC.2010.2053532

Ruby, J., & David, K. (2015). Analysis of nfluencing factors in predicting students performance using MLP -A comparative study. International Journal of Innovative Research

in Computer and Communication Engineering (An ISO Certified Organization), 3(2), 1085–1092. doi: 10.15680/ijircce.2015.0302070

S. Abu-Oda, G., & M. El-Halees, A. (2015). Data mining in higher education : University Student dropout case study. International Journal of Data Mining & Knowledge Management Process, 5(1), 15–27. doi: 10.5121/ijdkp.2015.5102

Santoso, L. W., & Yulia. (2019). The analysis of student performance using data mining. In: Bhatia S., Tiwari S., Mishra K., Trivedi M. (Eds) Advances in Computer Communication and Computational Sciences (pp. 559-573). Springer. doi: 10.1007/978-981-13-6861-5_48

Sara, N. B., Halland, R., Igel, C., & Alstrup, S. (2015). High-school dropout prediction using machine learning High-School Dropout Prediction Using Machine Learning: A Danish Large-scale Study. Presented at the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, (pp. 319–324). Bruges.

Satyanarayana, A. and Nuckowski, M. (04 2016) 'Data Mining using Ensemble Classifiers for Improved Prediction of Student Academic Performance', in.

Schmidhuber, J. (2015) 'Deep learning in neural networks: An overview', Neural Networks, 61, pp. 85–117. doi: 10.1016/j.neunet.2014.09.003.

Shrivas, A. K., & Tiwari, P. (2017). Comparative analysis of models for student performance with data mining tools. International Journal of Computer Trends and Technology, 46(1), 42–46. doi: 10.14445/22312803/ijctt-v46p109

Singh, J., & Kaur, E. H. (2018). Implementation of classification based algorithms to prediction of students performance using educational system. International Journal of Engineering Science and Computing, 8, 19233–19238.

Sukhija, K., Jindal, M., & Aggarwal, N. (2015). The recent state of educational data mining: A survey and future visions. Presented at the 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE), (pp. 354–359). doi: 10.1109/MITE.2015.7375344

Thakar, P., Anil Mehta, A. (2015). Performance analysis and prediction in educational data mining: A research travelogue. International Journal of Computer Applications, 110(15). 60-68.

Xu, R. and Wunsch, D. (2005) 'Survey of clustering algorithms', IEEE Transactions on Neural Networks, 16(3), pp. 645–678. doi: 10.1109/TNN.2005.845141.

Yehuala, M. A. (2015). Application of data mining techniques for student success and failure prediction the case of Debre Markos University. International Journal of Scientific & Technology Research,4 (4), 91–94.

Zhang, X., Sun, G., Pan, Y., Sun, H., He, Y., & Tan, J. (2018). Students performance modeling based on behavior pattern. Journal of Ambient Intelligence and Humanized Computing, 9(5), 1659–1670. doi: 10.1007/s12652-018-0864-6

Zhou, Q., Zheng, Y., & Mou,. (2015). Predicting students' performance of an offline course from their online behaviors. In 2015 Fifth International Conference on Digital Information and Communication Technology and its Applications (DICTAP), (pp. 70–73). doi: 10.1109/DICTAP.2015.7113173

R CODE

DATA PREPROCESSING

#SAMUEL

```
# Install and load required packages
install.packages("tidyverse")
install.packages("ggplot2")
install.packages("gridExtra")
library(ggplot2)
library(gridExtra)
library(tidyverse)

# Load your dataset (replace 'your_data.csv' with your actual data file)
data <- read.csv("students_adaptability_level_online_education (1).csv")

# Basic preprocessing
# Assuming you might need to convert factors to character for consistency
data <- data %>%
  mutate_if(is.factor, as.character)

# Data visualization and EDA
# Summary statistics
summary(data)

# Visualize distribution of categorical variables (bar plots)
categorical_vars <- c("Gender", "Education Level", "Institution Type", "IT Student", "Location",
                      "Load-shedding", "Financial Condition", "Internet Type", "Network Type",
                      "Self Lms", "Device", "Adaptivity Level")

# Create a list to store individual box plots
box_plots <- list()
```

```
# Iterate through each column and create a box plot
for (col_name in colnames(data)) {
  if (col_name != "Gender" && col_name != "Location" && is.numeric(data[[col_name]])) {
    plot <- ggplot(data, aes_string(y = col_name)) +
      geom_boxplot(fill = "lightgreen", color = "black") +
      labs(title = paste("Box Plot of", col_name), y = col_name) +
      theme(axis.text.x = element_blank(), axis.ticks.x = element_blank())  # Remove x-axis labels and ticks

    box_plots[[col_name]] <- plot
  }
}

# Create a list to store individual bar plots for non-numeric columns
bar_plots <- list()

# Iterate through each column and create bar plots for non-numeric columns
for (col_name in colnames(data)) {
  if (col_name != "Age" && !is.numeric(data[[col_name]])) {
    plot <- ggplot(data, aes_string(x = col_name)) +
      geom_bar(fill = "purple", color = "black") +
      labs(title = paste("Bar Plot of", col_name), x = col_name, y = "Count") +
      theme(axis.text.x = element_blank(), axis.ticks.x = element_blank())  # Remove x-axis labels and ticks

    bar_plots[[col_name]] <- plot
  }
}

# Arrange the box plots and bar plots in separate grids
grid_arrange_box <- do.call(grid.arrange, c(box_plots, ncol = 2))
```

```
grid_arrange_bar <- do.call(grid.arrange, c(bar_plots, ncol = 2))

# Display the grid of box plots and grid of bar plots
print(grid_arrange_box)
print(grid_arrange_bar)
```

ASOCIATION RULE MINING

#MUHAMMAD

```
install.packages("arules")
library(arules)

#Save R Output to Text File
sink("Rules Output default(sup= 0.1, conf= 0.7, minlen=2).txt")

dataAR <- read.csv("adapt_processed.csv")

#If you set the support threshold too high, you might miss out on discovering less
#frequent but still interesting associations.
#A higher confidence threshold will result in more reliable and strong rules,
#but it may also lead to fewer rules being generated.

#rules<-apriori(dataAR)#default setting supp = 0.1, conf = 0.8
#rules<-apriori(dataAR,parameter=list(supp=0.3,conf=0.3,minlen=2))
#rules<-apriori(dataAR,parameter=list(supp=0.5,conf=0.5,minlen=2))
#rules<-apriori(dataAR,parameter=list(supp=0.3,conf=0.8,minlen=2))
rules<-apriori(dataAR,parameter=list(supp=0.2,conf=0.7,minlen=2))

inspect(rules)

# Sort the association rules based on lift in descending order
sorted_rules <- sort(rules, by = "lift", decreasing = TRUE)
```

```
# Select the top 10 rules with the highest lift
top_10_rules <- sorted_rules[1:10]

# Close any open connections for writing (e.g., sink())
sink()

# Install the 'arulesViz' package for visualization of association rules
install.packages("arulesViz")
install.packages("cli")
library(arulesViz)

# Plot the top 10 rules using default visualization settings
plot(top_10_rules)

# Plot the top 10 rules using a graph-based visualization
plot(top_10_rules, method = "graph", control = list(type = "items"),
     shading = "confidence",   # Use shading to indicate confidence
     edge.label = "lift",      # Display lift values as edge labels
     arrow.size = 0.5,         # Adjust arrow size
     vertex.size = 5,         # Adjust vertex (node) size
     vertex.label.cex = 0.8,   # Adjust vertex label size
     edge.label.cex = 0.8,     # Adjust edge label size
     main = "Top 10 Association Rules",    # Set plot title
     sub = "Based on Lift and Confidence") # Set plot subtitle
```

NAÏVE BAYES

#MUHAMMAD

```
install.packages("ElemStatLearn")
install.packages("klaR")
```

```
install.packages("caret")
library(ElemStatLearn)
library(klaR)
library(caret)

# Read the dataset "adapt_processed.csv" into 'dataNB'
dataNB <- read.csv("adapt_processed.csv")

# Split the data into training and test sets
set.seed(123)  # For reproducibility
inx <- sample(nrow(dataNB), round(nrow(dataNB) * 0.7))
train <- dataNB[inx, ]
test <- dataNB[-inx, ]

# Separate features (x) and target variable (y) for training set
x_train <- train[, -13]  # Excluding column 13
y_train <- train$Adaptivity.Level

# Separate features (x) and target variable (y) for test set
x_test <- test[, -13]
y_test <- test$Adaptivity.Level

# Train a Naive Bayes model using cross-validation
nb_model <- train(x_train, y_train, method = 'nb',
                  trControl = trainControl(method = 'cv', number = 5))

# Make predictions using the trained Naive Bayes model on the test set
preds <- predict(nb_model, newdata = x_test)

# Create a confusion matrix to evaluate the performance of the model
tbl <- table(Actual = y_test, Predicted = preds)
```

```
print(tbl)

# Calculate and print the accuracy of the Naive Bayes model
accuracy <- sum(diag(tbl)) / sum(tbl)
cat("Accuracy:", accuracy, "\n")

# Create a confusion matrix and calculate additional metrics using caret's confusionMatrix function
conf_matrix <- confusionMatrix(table(preds, y_test))
print(conf_matrix)
```

RANDOM FOREST

#MUHAMMAD

```
# Install and load the randomForest package
install.packages("randomForest")
install.packages("caret")
library(caret)
library(randomForest)

# Read the CSV file "adapt_processed.csv" into the dataRF dataframe
dataRF <- read.csv("adapt_processed.csv")

# Assuming Adaptivity.Level is categorical
# Encode the categorical variable using ordinal encoding
dataRF$Adaptivity.Level <- as.integer(factor(dataRF$Adaptivity.Level))

# Fit a Random Forest ensemble model with 300 trees using the encoded Adaptivity.Level as the response variable
RFensemble300 <- randomForest(factor(Adaptivity.Level) ~ ., data = dataRF, ntree = 300)

# Display the summary of the RFensemble300 model
RFensemble300
```

Fit a Random Forest ensemble model with 300 trees and mtry=1 (consider only 1 feature at each split)

RFensemble300.mtry1 <- randomForest(factor(Adaptivity.Level) ~ ., data = dataRF, ntree = 300, mtry = 1)

Display the summary of the RFensemble300.mtry1 model

RFensemble300.mtry1

Fit a Random Forest ensemble model with 500 trees using the encoded Adaptivity.Level as the response variable

RFensemble500 <- randomForest(factor(Adaptivity.Level) ~ ., data = dataRF, ntree = 500)

Display the summary of the RFensemble500 model

RFensemble500

Create a plot of the RFensemble500 model

plot(RFensemble500)

Calculate feature importance for the RFensemble500 model

importance(RFensemble500)

Create a variable importance plot for the RFensemble500 model

varImpPlot(RFensemble500)

H-CLUSTERING

#SAMUEL

install.packages("cluster")

install.packages("factoextra")

library(cluster)

library(factoextra)

```
# Load data

data <- read.csv("converted_data.csv")


data[, 1:(ncol(data) - 1)] <- lapply(data[, 1:(ncol(data) - 1)], function(x) as.numeric(as.factor(x)))# Hierarchical clustering

numeric_data <- data[, sapply(data, is.numeric)]


# Impute missing values with column means

numeric_data[is.na(numeric_data)] <- colMeans(numeric_data, na.rm = TRUE)


dist_matrix <- dist(numeric_data)

d = dist_matrix

hc <- hclust(d, method="average")


# Plot dendrogram

plot(hc, hang=-1, main="Dendrogram")


# Color labels by cluster (k=3)

clusters <- cutree(hc, k=3)

rect.hclust(hc, k=3, border="red")

labels_col <- rainbow(3)[clusters]

labels(hc, col=labels_col)


# Silhouette analysis

sil <- silhouette(clusters, dist=d)

mean_sil <- mean(sil[,3]) # Average silhouette width


# Compare methods

hc_single <- hclust(d, method="single")

hc_avg <- hclust(d, method="average")
```

```
plot(hc_single, main="Single Linkage", hang=-1)
plot(hc_avg, main="Average Linkage", hang=-1)
```

GAUSIAN MIXTURE MODEL

#SAMUEL

```
install.packages("mclust")
library(mclust)

# Load data
data <- read.csv("converted_data.csv")

# Preprocess data
data[, 1:(ncol(data) - 1)] <- lapply(data[, 1:(ncol(data) - 1)], function(x) as.numeric(as.factor(x)))
numeric_data <- data[, sapply(data, is.numeric)]
numeric_data[is.na(numeric_data)] <- colMeans(numeric_data, na.rm = TRUE)

# Fit GMM model
gmm <- Mclust(numeric_data, G = 1:10)

# Calculate silhouette scores
num_clusters <- 1:10
silhouette_scores <- numeric(length(num_clusters))

for (i in num_clusters) {
  cluster_assignments <- gmm$classification[, i]
  silhouette_scores[i] <- silhouette(numeric_data, cluster_assignments)$avg.width
}

# Plot BIC and silhouette scores
par(mfrow = c(1, 2))  # Create a 1x2 grid of plots
```

```
plot(gmm, what = "BIC")

plot(num_clusters, silhouette_scores, type = "b", pch = 19,
     xlab = "Number of Clusters", ylab = "Silhouette Score")
```

DEEP LEARNING

#SAMUEL

```
install.packages("neuralnet")
library(neuralnet)

# Read the CSV file
data <- read.csv("students_adaptability_level_online_education (1).csv")

# Convert all columns to numeric except the last column
data[, 1:(ncol(data) - 1)] <- lapply(data[, 1:(ncol(data) - 1)], function(x) as.numeric(as.factor(x)))

# Print the converted data frame
print(data)

write.csv(data, "adapt_data.csv", row.names = FALSE)
#Parameter
y = as.matrix(data[,12])
y[which(y=="Low")] = 0
y[which(y=="High")] = 1
y = as.numeric(y)
x = as.numeric(as.matrix(data[,2:11]))
x = matrix(as.numeric(x),ncol=10)

# Neural net model
nn <- neuralnet(y ~ Gender  + `Institution.Type` + `IT.Student` +
                Location + `Load.shedding` + `Financial.Condition` +
```

```
          `Internet.Type` + `Self.Lms` + Device,
        data = data, hidden = 5)

# Predict results
yy = nn$net.result[[1]]
yhat = matrix(0,length(y),1)
yhat[which(yy > mean(yy))] = 1
yhat[which(yy <= mean(yy))] = 0
cm = print(table(y,yhat))
cm

#Plot Model
plot(nn)
```