FUDMA Journal of Sciences (FJS)
ISSN online: 2616-1370
ISSN print: 2645 - 2944
Vol. 10 No. 3, February, 2026, pp 364 – 376
DOI: https://doi.org/10.33003/fjs-2026-1003-4751

A HYBRID PREDICTION MODEL FOR CLASSIFYING STUDENT'S ACADEMIC PERFORMANCE USING VOTING ENSEMBLE METHOD

***[1]Oruh, Jane Ngozi, and [2]Iwuji, Basil Chukwuemeka**

[1]Department of Computer Science, Michael Okpara University of Agriculture, Umudike, Abia State, Nigeria.
[2]School of Postgraduate Studies, Abia State University, Uturu, Abia State, Nigeria.

*Corresponding authors' email: janeoruh@mouau.edu.ng

ABSTRACT

Single classification algorithms for predicting students' academic performance often face limitations in accuracy and robustness, especially with high-dimensional educational data. This study proposes a hybrid prediction model that integrates K-Nearest Neighbour (K-NN) and Naïve Bayes (NB) algorithms using a soft voting ensemble technique to classify students' academic performance into high, average, and low categories. The Cross-Industry Standard Process for Data Mining (CRISP-DM) and Object-Oriented Analysis and Design Methodology (OOADM) guided the study. Primary data were collected from 240 third-year nursing students of Abia State University, Uturu, across two academic semesters, using 38 demographic, academic, social, and prior performance features. The models were implemented in Python using Scikit-learn and related libraries, with an 80:20 train-test split after preprocessing and feature scaling. Performance was evaluated using accuracy, precision, recall, F1-score, ROC-AUC, and confusion matrices. Results show that the hybrid model achieved superior performance, attaining 100% accuracy and outperforming the standalone K-NN model. The classification results indicated that 51.7% of students were high performers, 10.8% average, and 37.5% low. The deployed system provides performance-based advisory support and demonstrates strong potential for early intervention, personalized learning, and academic planning in educational institutions.

Keywords: Hybrid Prediction, Classification, K-Nearest Neighbour (K-NN), Naïve Bayes (NB), Ensemble Method, Academic Performance

INTRODUCTION

Academic performance is a key indicator of student success and a strong predictor of future educational and career outcomes (Kotsiantis et al., 2007). With the growth of big data and machine learning, predicting student performance from historical academic, demographic, and behavioral data has become increasingly feasible and valuable for educational institutions (Romero & Ventura, 2010). Early identification of students at risk enables timely intervention and improved academic outcomes (Baker & Inventado, 2014).

Academic performance is the extent to which a student, teacher or institution has achieved their short or long-term educational goals. Cumulative GPA and completion of educational benchmarks such as secondary school, diplomas, bachelor's degrees and postgraduate degree represent academic achievement.

Academic performance achievement is the level of achievement of the students' educational goal that can be measured and tested through examination, assessments and other form of measurements. Academic achievement is commonly measured through examinations or continuous assessments but there is no general agreement on how it is best evaluated or which aspects are most important: procedural knowledge such as skills or declarative knowledge such as facts (Ward et al.,1996).

Furthermore, there are inconclusive results over which individual factors successfully predict academic performance, elements such as test anxiety, environment, motivation, and emotions require consideration when developing models of school achievement. An academic institution with more academic achievements is more likely to be sought than an institution with less achievement (Ziedner, 1996).

Traditional statistical models often struggle to capture complex nonlinear relationships in educational data, a limitation addressed by machine learning techniques (Al-Barrak & Al-Razgan, 2016). Algorithms such as K-Nearest Neighbour (K-NN) and Naïve Bayes have shown promising performance in educational data mining due to their simplicity and efficiency (Baradwaj & Pal, 2011).

However, single-algorithm approaches face limitations, including sensitivity to irrelevant features in K-NN and the independence assumptions of Naïve Bayes (Khan et al., 2019). To overcome these challenges, ensemble learning methods that combine multiple classifiers have been shown to improve predictive accuracy and robustness (Dietterich, 2000; Zhou, 2012). This study contributes by developing a hybrid ensemble model integrating K-NN and Naïve Bayes to enhance the classification of students' academic performance (Bunkar et al., 2012).

By integrating K-NN and Naive Bayes within an ensemble framework, it is possible to build a more robust hybrid model that improves classification of student academic performance, enabling better decision-making for educators and administrators.

Problem Statement

Despite advances in educational technologies, many institutions continue to face challenges in accurately identifying students at risk of poor academic performance. Existing predictive models often exhibit limitations related to accuracy, scalability, and interpretability. While single-algorithm approaches such as K-Nearest Neighbour (K-NN) and Naïve Bayes demonstrate notable strengths, each also presents inherent weaknesses when applied in isolation. Consequently, reliance on a single classifier may lead to sub-optimal prediction outcomes. There is therefore a need for a more robust and comprehensive predictive approach that integrates the complementary strengths of multiple algorithms to enhance classification accuracy. This study addresses this gap by proposing a hybrid predictive model based on ensemble learning techniques for improved classification of students' academic performance.

Contribution to Knowlegde

1. The study develops a hybrid model, a combination of K-NN and Naïve Bayes algorithm to classify the various expected performance levels (high performance, low performance and average performance) for under graduate third year students.
2. The ability of the developed model to extract the attribute importance ranking of attributes to determine which factors have significant contribution to the prediction of the overall academic performance.
3. The proposed model provides a reference and comparative study for the next researcher in application of the hybrid prediction system.

Algorithms for Predicting Student Performance

The most popular task to predict students' performance is classification. Romero et al., (2008) in their research presented several mining algorithms that are used to mine educational data to classify students. There are several algorithms under classification task that have been applied to predict students' performance. Among the algorithms used are Decision tree, Artificial Neural Networks, Naive Bayes, K-Nearest Neighbour and Support Vector Machine.

Combining Classifiers

Combining classifiers also known as ensemble leaning is a technique in machine learning where multiple classifiers are used together to improve predictive performance. Ensemble learning uses various base classifiers combined using a particular strategy of combination such as bagging, boosting, voting, etc. (Avula and Asa, 2018). The outputs of several classifiers will reduce error and increase robustness. Before combining algorithms you need to know the strengths and weaknesses of each algorithm you are considering to combine together.

Theoretical Framework of Naïve Bayes and K-Nearest Neighour Algorithms

Naive Bayes Naive

The Naïve Bayes classifier technique is based on Bayesian theorem, whereas it performs better when data dimensionality is high (Nikam, 2015). The Bayesian classifier is capable of calculating the most possible output based on the input. There is no problem to add new raw data at run time and have a better probabilistic classifier. In this algorithm, the presence of a particular feature in a class is unrelated to the presence of any other feature. Describing by example why this algorithm is called a naïve, a fruit is judged as an apple when its characteristics are: round 3 inches in diameter and red, even it depends on each other or on other features, all of these properties independently contribute to the probability to judge that fruit is apple (Nikam, 2015). Bayesian theorem provides an equation for calculating posterior probability P(c|x) from P(c), P(x) and P(x|c):

$$p(c|x) = \frac{p(x|c)p(c)}{P(x)}$$

c: class (target)

x: Predictor (attribute)

P(c|x): the posterior probability of class (c, target) given predictor (x, attributes).

P(c): the prior probability of class (Proposition)

P(x|c): the likelihood, which is the probability of predictor given class.

P(x): the prior probability of predictor (Evidence)

Bayes algorithms assume that the effect an attribute plays on a given class is independent of the values of other attributes. However, in practice, dependencies often exist among attributes (Kabakchieva 2013). In the research paper of Nikolovski et al., (2015) they explained that Naïve Bayes classifier is based on the Bayes rule of conditional probability. It analyzes all the contained attributes individually as though they are equally important and independent of each other. However, in practice, dependencies often exist among attributes; hence Bayesian networks are graphical models, which can describe joint conditional probability distributions. Naive Bayes algorithm is also an option for researchers to make a prediction. Several researchers have used Naive Bayes algorithms to estimate students' performance. The objective of their research is to find the most effective prediction technique in predicting students' performance by making comparisons. Their research showed that Naive Bayes has used all of attributes contained in the data. Then, it analyzed each one of them to show the importance and independency of each attributes (Osmanbegovi´c et al., 2012).

K-Nearest Neighbour (K-NN)

K-Nearest Neighbour is non-parametric lazy learner algorithm for classification and prediction. In order to classify a new instance, this algorithm checks the distance of its k neighbours from the training set to classify it. In general Euclidean Distance measure is used to find the distance. A training instance closest to the given test instance predicts the same class as this training instance. In WEKA this algorithm is available as IBK (Pandey et al., 2014). K-Nearest Neighbour method had taken less time to identify the students' performance as a slow learner, average learner, good learner and excellent learner. K-Nearest Neighbour gives a good accuracy in estimating the detailed pattern for learner's progression in tertiary education (Gray et al., 2014).

Some algorithms increase the speed of basic K-NN algorithm e.g. ball tree, k-d tree, nearest feature line(NFL), tunable metric, orthogonal search tree and principal axis search tree. To more understand K-NN algorithm, suppose that an object is sampled with a set of different attributes, but the group to which the object belongs is unknown. Determining the class of a sample depends on evaluating the k-number of closest neighbours. K-NN classifies the test data using the training set directly. To classify any test data, it first calculates K value, which denotes the number of K-Nearest Neighbours. For each test data, it calculates the distance between all the training data and then sorts the distance. Then by using majority voting, class label will be assigned to the test data. The main objective of K-NN is to find the nearest neighbour of an unknown data point and the value of k. If k=n, then the "n" nearest neighbour can be predicted. The final classification output is decided by calculating the distance between the test data and each of the training data with the help of K-NN algorithm.

The Euclidean distance between two points ($x1$, 1 and $x2$, $y2$ in the plane is given by the equation

Dist ((x_1, y1 and $x_{2,2}$) = $\sqrt{}(x_1 - x_2)^2$ + $(y_1 - y_2)^2$

Students Grades Predictor using Naïve Bayes Classifier

Dake and Gyimah (2017) proposed a Naïve Bayes approach in predicting student's final grade. The Classifier model was built on the data of previous students who offered the same course. The attributes and their values were selected based on the discretion that can have effects on the students' ability to pass or fail an examination. To run the classifier, a total of 50 instances were taken for analysis. The training data set was then subjected to Naïve Bayes Classification. The result obtained gave 88% accuracy for correctly classified instances. The model was evaluated using the 10 folds validation. Out of the 50 instances, 44 were correctly classified and 6 were incorrectly classified. The students' academic status was then predicted. The diagram in figure 1 highlights the stages for the classification of student's performance. Data is first collected from their various sources and relevant attributes or features are selected. Next pre-processing is done. In this stage, data is transformed to the format which the classification can be done. It involves feature extraction, normalization and discretization. Naïve Bayes Classifier is then applied to mine the patterns and discover important features in the data. Finally the results are evaluated.

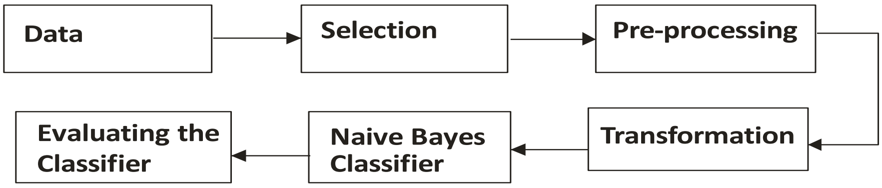

Figure 1: Extracting of Knowledge Using Naïve Bayes Classifier (Dake amd Gyimah, 2017)

Predicating Academic Performance of Students in Higher Institution with KNN Classifier

Omisore and Azeez (2016) developed a predictive model using KNN Classification to predict the academic performance of students in higher institutions. Educational dataset of 310 students from all levels in 2013/2014 session was collected from the University of Lagos, Akoka. Other relevant information was collected via questionnaires. Various selected features were used for the prediction. Variables/students attribute used include student's demographic, current and previous academic standing, departmental structure and family background. Subsequently, data captured into different tables were joined and attributes with lesser entropy were removed as shown in figure 2 below. Students' academic standing was stratified into five different groups- Distinctive, Good, Average, Weak and Hapless. 10 Folds Cross Validation was used to evaluate the performance. The result obtained gave a prediction accuracy of 58.3%.

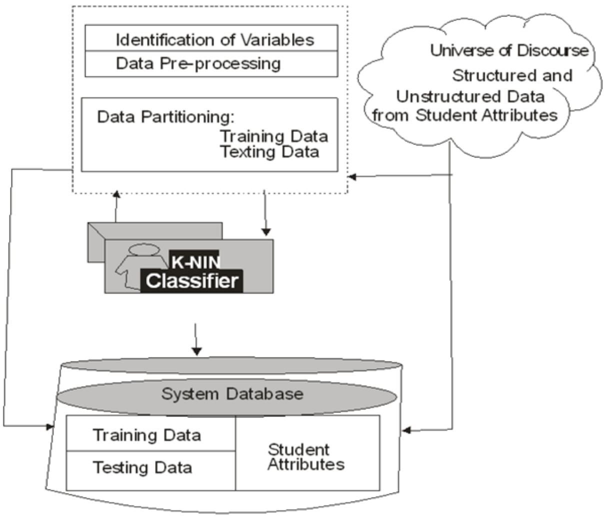

Figure 2: Conceptualized Model for Prediction of Students Academic Performance(Omisore and Azeez, 2016)

Review of Related Literature

Amin et.al (2017) proposed and validated a predictive GPA model by using machine learning approaches. A relatively small-sample experiment was used for determining the set of self-regulatory learning behaviors. The major objective of grade prediction was utilizing the constructed models for designing the intervention strategies that help students when the academic failure is at risk. A probabilistic predictive model of GPA was used to define and detect the helpful interventions as per the mathematical calculations. The basic interventions were defined and the interventions which are of help to students having minimum GPA were identified by the application framework. 53% of accuracy was achieved by the proposed algorithm.

Merchán et.al (2016) proposed a predictive model to be applied to predict academic performance of students. Several data mining methods were applied on the data of 932 Systems Engineering students of a university in Columbia to evaluate and analyze their performances. On the basis of input data given, the expected results and output characterization and other factors, the evaluation of results achieved was done. The prediction accuracy was an important parameter to evaluate the performances as well. Considering the specific details of the population examined and the requirements specified by the institution, the said pertinence was evaluated. In preventing, academic risk and desertion, timely decisions were considered important along with the accompaniment of students with their learning procedure.

Sivasakthi (2017) proposed a knowledge flow model using all the five different classifiers. Within the programming education field, the importance of prediction and classification based algorithms was also studied. For predicting the programming performance of students, five supervised data mining algorithms were applied on the data set. On the basis of predictive accuracy, the performance evaluations of these algorithms were done. It was seen that 93% accuracy was achieved in case of implementing Multilayer perceptron (MLP). Further, WEKA scenario was implemented to compare all the five classifiers. The performance of MLP was shown to be the best. The students that are very new to the introductory programming were identified through the research so that they can be helped with special attention.

Nakayama et.al (2018) evaluated the regression models using the fitness models and analyzed their contributions. With the help of these evaluations, the effectiveness of learner's reflections was measured such that the learning performance was predicted. A variable selection technique was used to examine the contributing variables using a step-wise procedure. The R-squared (R2) and Akaike information criterion (AIC) indices were used to perform comparisons against the fitness of these models. When employment of indices of participant's reflection is applied, the improvement in performance of regression models is done. A variable selection technique was utilized to choose few reflection indices for the regression model even though the scores of final exams and change of variables were not in correlation. It thus proved that the hypothesis which states that the learning performance is affected by the contribution of assessment of reflections is correct.

Fan Yanga et.al (2018) analyzed the performance of students, their progress and potentials using the multiple analysis tools. Initially, Student Attribute Matrix (SAM) was used to formulate the student model along with performance and non-performance related attributes. Secondly, BP-NN algorithm was applied for providing student performance estimation tools. The prior knowledge of students and their performance attributes were used to estimate the attributes of students. For describing the progress of students related to different aspects along with the casual relationships, the BP-NN was used to propose the student progress indicators and attributes. The level at which a factor would affect the performance of student was known by the indicators and predicator. It was thus possible to train up the students. The real academic performance data which was gathered from 60 high school students was used to check the performance of the analysis tools. Correct and highly accurate results were achieved by applying the proposed tools as per the evaluation results.

Paris et al (2010) compared the data mining methods accuracy of classifying students to predict class grade of a student. The predictions were more useful for identifying the weak students and assisting administration to take remedial measures at initial stages to produce excellent graduate that will graduate at least with the second class upper.

Minaei-Bidgoli and Punch (2004) applied data mining classifiers as a means of comparing and analyzing students' use and performance for those who had taken a technical course via the web. The results showed that combining multiple classifiers led to a significant accuracy improvement in a given data set.

Devasia et al (2016) proposed classification within the information of student such that on the basis of previously existing information, the division of students can be predicted. Naïve theorem was applied since several techniques were used for knowledge classification within the area unit. For the prediction of performance at the top of that particular semester, various types of information were collected from the previous information of the students available. Students who needed special guidance can be highlighted through the study.

Dewan Md. Farid et al (2014) anticipated a hybrid algorithm using decision tree and Naive Bayes algorithm. The method was mainly aimed at increasing the accuracy of classification for multi class-class classification tasks. The proposed hybrid algorithm showed the better sensitivity, specificity, cross validation and classification accuracy on real benchmark data sets. The proposed hybrid algorithm showed 90% accuracy.

Ulyani et.al (2017) presented that the major factor that leaves a huge impact on the behavioral intentions of student is the service quality performance. Questionnaires were distributed within seven Malaysian public and private universities. The descriptive statistics and covariance-based structural equation modeling were used to analyze the data. The least likely execution of favourable behavioral intentions was influenced by the freedom, serenity, management dimensions as well as aesthetic factors. A positive behavior towards the student housing was seen as per the results achieved when students adapted to live in multi-cultural community in which they would have access to good hospitality, personal privacy and appropriate building ambiance.

Safri et al (2018) applied a combination of Naive Bayes Classifier algorithm and K-Nearest Neighbor determining the feasibility of healthy Indonesian card recipients; the Naive Bayes Classifier algorithm aimed to minimize variation within an attribute to obtain accurate higher than the K-Nearest Neighbor algorithm alone.

Rathee and Mathur (2013) applied ID3, C4.5 and CART decision tree algorithms on the educational data for predicting a student's performance in the examination. All the algorithms were applied on the internal assessment data of the student to predict their academic performance in the final exam. The efficiency of various decision tree algorithms was analyzed based on their accuracy and time taken to derive the tree. The predictions obtained from the system helped the tutor to identify the weak students and improve their performance. C4.5 was the best algorithm among all the three because it provided better accuracy and efficiency than the other algorithms.

Amrieh et al (2016) studied on performance prediction at the University of Jordan. A data set of students from different countries was used. In addition to using individual machine learning methods, the researchers also applied ensemble methods, and compared the results between them. Decision trees provided the best results. Another area that the researchers focused on was behavioural features. It was found that the inclusion of behavioural features improved the prediction results (Amrieh et al., 2016).

Naren (2014) proposed a system that specifies the classification techniques for predicting the career options and to predict the violent behaviour prevalent among students. A response sheet was used to gather details regarding the background information, reaction of a student when irritated and the interests of a particular student. These parameters were mined to find the corresponding behavioural pattern of a student. However, additional research on student attitudes and learning was needed to predict the patterns efficiently. Minimal number of attributes was used which in turn did not give accurate values. Only limited amount of information was suggested for prediction of career options for students.

Recently**,** Abubakar et al., (2025) proposed a hybrid model for student' academic performance based on Machine Learning. The study developed a hybrid model incorporating Linear Regression, Decision Trees, and Random Forests. It was tested on a dataset from Umaru Ali Shinkafi Polytechnic. The model achieved an accuracy of 0.85, an F1-score of 0.85, and an ROC-AUC score of 0.91, outperforming standard benchmark techniques.

Summary of Literature Review and Knowledge Gap

After reviewing typical works on **s**tudent performance prediction, the following were found out. While previous studies focused on predicting student performance using single models and limited datasets, this work contributes to knowledge by developing a web-based, interactive result prediction and advice system that leverages a soft-voting ensemble (K-NN + Naive Bayes)**,** integrates behavioral and academic features**,** generates actionable advice**,** and maintains historical and retrainable data in MongoDB**,** thereby addressing both prediction accuracy and practical usability gaps in the literature.

Knowledge Gap

Despite significant advancement in the field of machine learning, especially in predictive modelling for classifying student's academic performance, most of the studies used single models for classification (Amin et. al, (2017), Dake and Gyimah (2017), Kahakchieva (2013)).

However, there is a noticeable gap in leveraging hybrid models for the classification of student's academic performance. The proposed model tends to cover this gap by using the K-NN and Naïve Bayes algorithms to classify the student's academic performance of the third year nursing students in Abia State University, Uturu,

Additionally, there is need for a system that will provide actionable advice to students and allow interactive Admin/student web access and to support model maintenance and scaling.

MATERIALS AND METHODS

Methodology Adopted

The research methodology that was adopted for this study were cross industrial standard process for data mining (CRISP-DM) and object-oriented analysis and design methodology (OOADM).

Cross-Industrial Standard Process for Data Mining (CRISP-DM)

CRISP-DM is a structured, iterative methodology for developing machine learning and data-driven solutions that guides project from business problem definition through data understanding, preparation, model building, evaluation, and deployment, ensuring that machine learning models are aligned with business objectives and deliver actionable, real-world value (Chapman et al., 2000). The six phases of CRISP-DM is shown in figure 3.

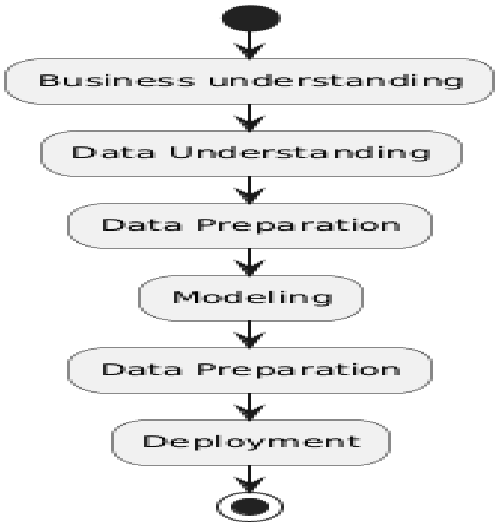

Figure 3: CRISP-DM Lifecycle Showing the Six Iterative Phases (Chapman et al., 2000)

Step 1: Business Understanding

This step defines the main goal of the project and the problem it seeks to solve. It helps to understand what the users or organization need and how data mining can provide a solution. In this study, the aim is to address the difficulty schools face in tracking and improving student performance. The goal is to build a system that can predict students' academic outcomes and help identify those who may need extra support.

Step 2: Data Understanding

The dataset used in this study was obtained from the third year students of Nursing Science, Abia State University, Uturu. Table 3.1 contains attributes as included in the questionnaire and their scaling factors. The attributes were analysed in a 1-5 scale basis. It was stored in MongoDB.

Step 3: Data Preparation

The dataset stored in the database was loaded into Pandas library for inspection and cleaning. The dataset was inspected for inconsistencies, missing values, and anomalies. Missing data was addressed using imputation techniques like mean, median, or mode replacement. Duplicate records was removed to ensure data integrity. Outliers was detected through statistical methods like Z-score or interquartile range (IQR) analysis and was treated appropriately to avoid biasing the model.

Categorical variables like gender or grade categories was converted into numerical formats suitable for machine learning algorithms. Binary variables was encoded using label encoding, while nominal variables with multiple categories was transformed using one-hot encoding. Numerical features underwent scaling via normalization or standardization to ensure consistency across all features, thereby improving model performance.

Key features that significantly impact academic performance was identified through correlation analysis and mutual information measures. Additional features was engineered based on domain knowledge, such as composite academic performance indices or attendance consistency metrics, to enhance the predictive capability of the model.

Step 4: Modeling

The dataset was split into training and testing sets using the Python Scikit-learn library, typically using an 80:20 split. The model was first trained individually with input features using

K-NN and Naïve Bayes algorithms. Later, the hybrid model was obtained by combining the two classifiers using an ensemble learning framework. A voting classifier approach was utilized to aggregate the predictions of the individual classifiers through averaging predicted probabilities (soft voting).

Step 5: Evaluation

Performance evaluation was assessed using metrics like accuracy, precision, recall, F1-Score, Confusion matrix, F1-Score and –ROC-AUC.

Step 6: Deployment

The final trained model was serialized using Joblib into a Python Streamlit framework to allow for data input and display of prediction results.

Object-oriented Analysis and Design Methodology (OOADM)

OOADM was adopted in this research work is a set of standards for analysis and development of the students' academic performance prediction system. It uses a formal methodical approach to the analysis and design of information system. Object-oriented design (OOD) elaborates the analysis models to produce implementation specifications. The main difference between object-oriented analysis and other forms of analysis is that the object-oriented approach organizes requirements around objects, which integrate both behaviours (processes) and states (data) modeled after real world objects which the system interacts with. In other traditional analysis methodologies, the two aspects: processes and data are considered separately. The primary tasks in object-oriented analysis (OOA) are:

a. Find the objects and organize them
b. Describe how the objects interact
c. Define the behaviour of the objects
d. Define the internals of the objects.

Data Flow Diagram of the New System (DFD)

The data flow process is illustrated in Figure 4. The user logs in based on Access Status (Student/ Lecturer/Advisor). On successful log in, for the student, he/she is able to register his or her courses, fill other information that may be required based on the attributes. The student can then go ahead to see the performance prediction. The system predicts the student's performance and further takes a suitable action to improve the student performance by advising the students and proffering solutions based on the student's academic standing. The technique will also monitor and evaluate the student academic performance at different year levels before the final semester in order to predict the grade of the students. The academic adviser can also log in to view students' performance and also generate report on the academic standing of all students in his purview. The admin updates curriculum and updates other information. He/she also creates accounts for each user. The admin can query the database for each student's information, request for the students' academic standing and generate reports on the academic standing of students.

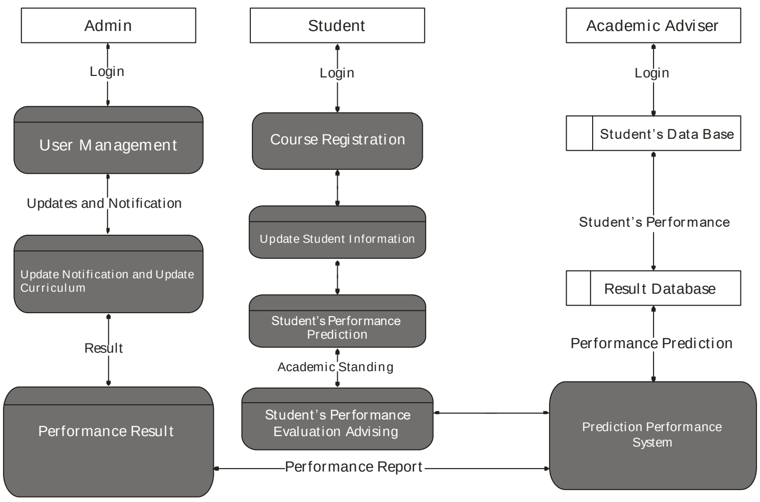

Figure 3: Data Flow Diagram of the Proposed System

Design and Implementation

This chapter provides the details of system design and implementation of student academic performance prediction. The aim is to develop a hybrid prediction model for classifying student academic performance. The model of the new system is shown in figure 4.1. Data was collected from multiple sources: via questionnaire and academic unit of the school. These data was pre-processed to transform the data from the raw form to a much more usable or desired form before feeding in the algorithms. Two algorithms, Naïve Bayes and K-Nearest Neighbour (K-NN) algorithm were applied to the data. A total of 240 instances were used for the analysis. Results from Naïve Bayes and K-NN classification was then combined and used to develop the hybrid model (K-Bay). The results was evaluated, thereafter performance prediction and academic advising were done based on the result.

System Flowchart

The system flowchart illustrates the logical flow of activities within the academic performance prediction system. It shows how data moves from one process to another, from user input to prediction and advice generation. This is drawn using Planttext.com. The system flowchart is shown in figure 5

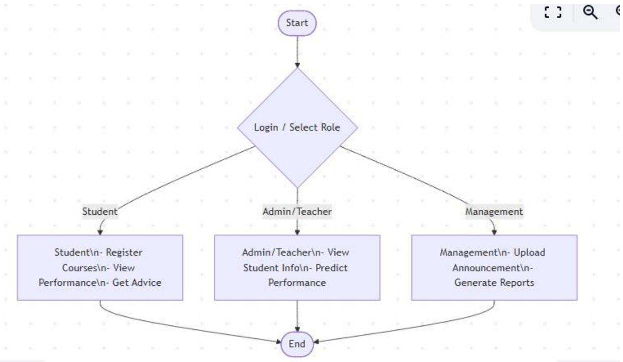

Figure 5: System Flow chart

Class Diagram
The class diagram is designed using Planttext.com and shows the main classes that make up the system and the relationships between them. It helps to describe the structure of the software in terms of its components, their attributes, and how they interact. The class diagram for the student academic performance prediction is shown in figure 6.

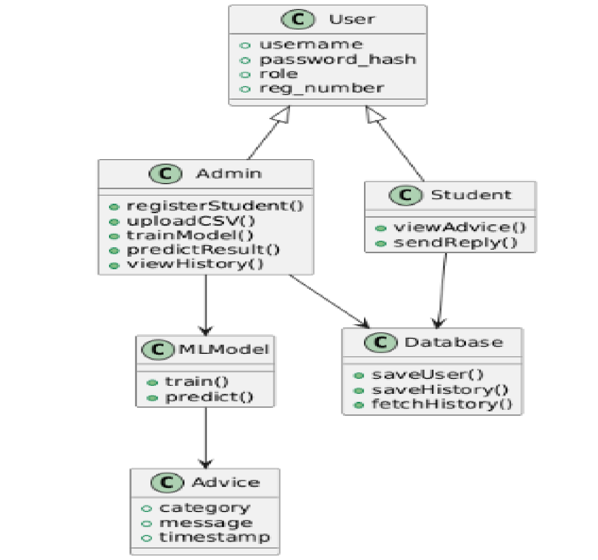

Figure 6: Class Diagram for Student Performance Prediction

Sequence Diagram for Student Performance Prediction
The sequence diagram shows how different components of the system interact over time to complete specific processes. It focuses on the flow of messages between objects and the order in which actions occur. Figure 7 describes the sequence diagram of the student academic performance prediction.

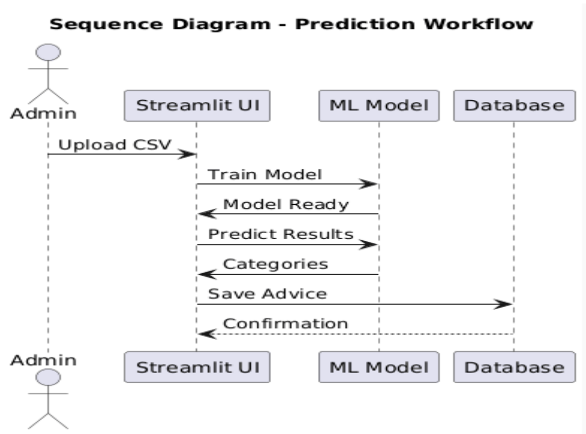

Figure 7: Sequence Diagram of Student Academic Performance Prediction

Activity Diagram for Student

The activity diagram describes the workflow of the system by showing the sequence of actions and decisions that occur during operation. It provides a visual representation of how different activities are connected from start to finish. Figure 8 shows the activity diagram of the student academic performance prediction system.

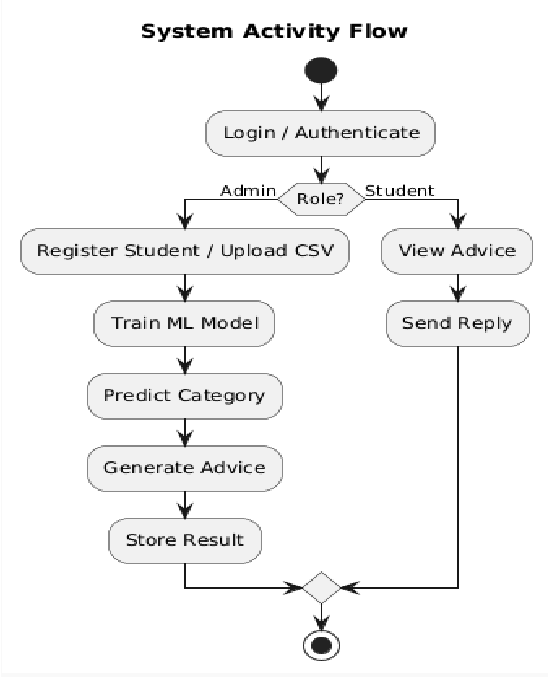

Figure 8: Activity Diagram of the Student Academic Performance Prediction

Use Case Diagram

The use case diagram depicts the functional requirements of the *Result Prediction & Advice System*. It shows the different actions that admins and students can perform and how these actions relate to the underlying machine learning model and database. Figure 9 shows the use case diagram for both the student and Administrator.

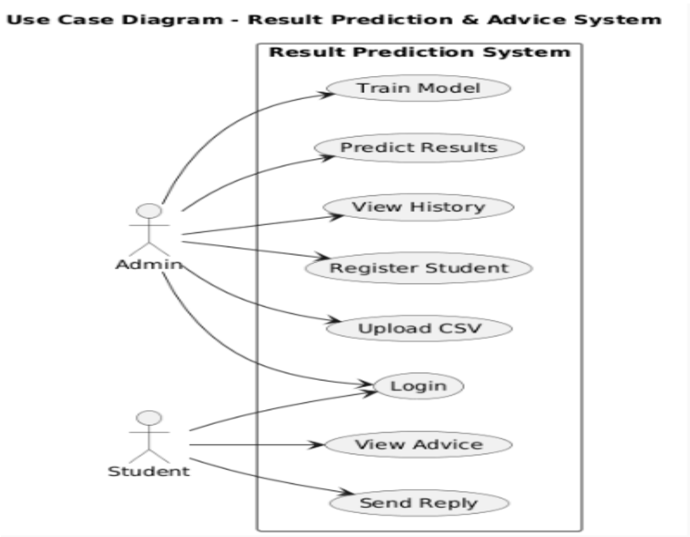

Figure 9: Use Case Diagram of the System

Architecture of the System

The *Result Prediction & Advice System* uses a three-tier architecture consisting of the User Interface**,** Application Logic, and Database layers.

The frontend: (built with Streamlit) allows admins and students to log in, upload data, view predictions, and exchange messages.

The application layer: handles machine learning tasks, training a Voting Classifier (KNN + Naive Bayes), generating predictions, and creating advice.

The database layer: stores user information, prediction results, and advice in MongoDB Atlas, with CSV files for local backup. This design ensures smooth data flow, easy updates, and reliable performance. Figure 10 shows the architecture of the system.

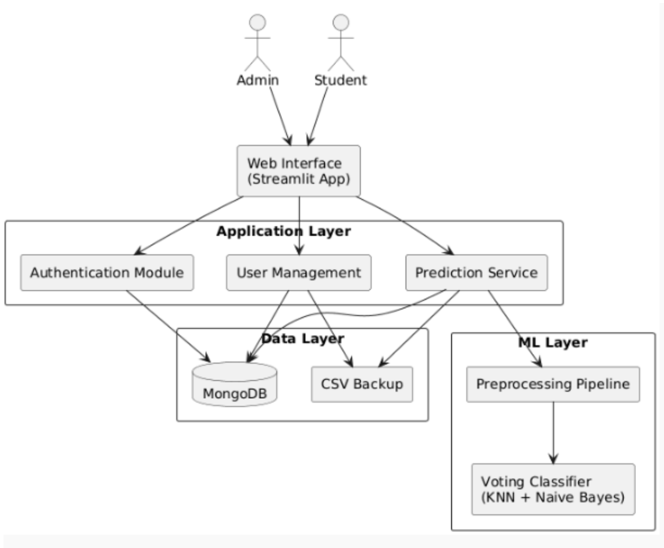

Figure 10: Architecture of the System

RESULTS AND DISCUSSION

Model Evaluation

Evaluation of classification algorithms is one of the key points in any process of data mining. The most common tools used in analyzing the results of classification algorithms applied are: confusion matrix, learning curves and receiver operating curves (ROC).

Confusion Matrix

The confusion matrix or contingency table is a visualization tool commonly used to present performances of classifiers in classification tasks. It is used to show the relationships between real class attributes and that of predicted classes. The level of effectiveness of the classification model is calculated with the number of correct and incorrect classifications in each possible value of the variables being classified in the confusion matrix.

Table 1 describes the Contingency table. The true positives (TP) means the correct classifications of the positive class A; true negatives (TN) are the correct classifications of the negative class B; false positives (FP) represent the incorrect classification of the negative class A into the positive class A, and false negatives (FN) are the incorrect classification of the positive Class B into the negative class B.

Table 1: Confusion Table (Contingency Table)

	Predicted Class		
Actual Class	Class	Class A	Class B
	A	True Positive (TP)	False Negative (FN)
	B	False Positive (FP)	True Negative (TN)

Results of the Performance Evaluation

Table 2: Results of the Performance Evaluation

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**ROC-AUC**
KNN	0.3958	0.3396	0.4192	0.3513	0.7868
Naive Bayes	1.0000	1.0000	1.0000	1.000	1.0000
Voting Ensemble	1.0000	1.0000	1.0000	1.000	1.0000

Table 5.2 summarizes the performance metrics (Accuracy, Precision, Recall, F1-Score and ROC-AUC) of three different machine learning models.

RESULTS AND DISCUSSION

From Table 2, the KNN model shows relatively low performance, with accuracy of 39.58% and modest precision, recall and F1-score. This indicates that KNN struggles to correctly classify the data, likely due to overlapping class features.

In contrast, Naïve Bayes achieved perfect performance across all evaluation metrics (Accuracy, Precision, Recall, F1-score, and ROC-AUC = 1.00). This indicates that the model classifies all samples correctly and fits the dataset extremely well. The Soft Voting Ensemble model achieved perfect scores, matching the performance of Naïve Bayes. This suggests that the Soft Voting Ensemble is heavily influenced by the Naïve Bayes classifier and does not improve further upon its performance.

Graphical comparison of the three different models is shown in the bar chart in Figure 11. The bar chart shows that K-NN performed lowest and the hybrid has the same performance as the Naïve Bayes result.

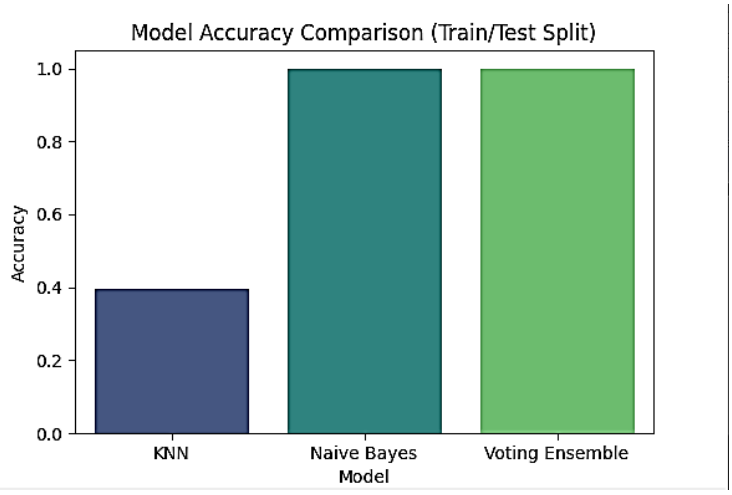

Figure 11: Bar Chart of Model Accuracy Comparison

Confusion Matrix Evaluation of KNN and Naïve Bayes

The confusion matrices show the classification performance of the KNN and Naïve Bayes models across six labels (Figure 5.2). The rows represent the actual class labels, while the columns represent the predicted class labels. Correct classifications are indicated along the diagonal of each matrix.

For the KNN model, correct predictions are spread across multiple classes, but several classes show noticeable miscalculation particularly between classes 0, 2, and 4. This indicates that KNN has difficulty clearly distinguishing between some classes.

For the Naïve Bayes model, higher correct classification is observed for certain classes, especially class 2. However, some classes are frequently misclassified into dominant class, indicating prediction bias.

Overall, the figure illustrates that KNN provides more balanced predictions across classes, while Naïve Bayes performs strongly for specific classes but with higher misclassification in others.

In summary, Naïve Bayes outperformed KNN in terms of overall classification accuracy, while KNN shows more balanced but weaker performance. Figure 12 shows confusion matrix for KNN and Naïve Bayes.

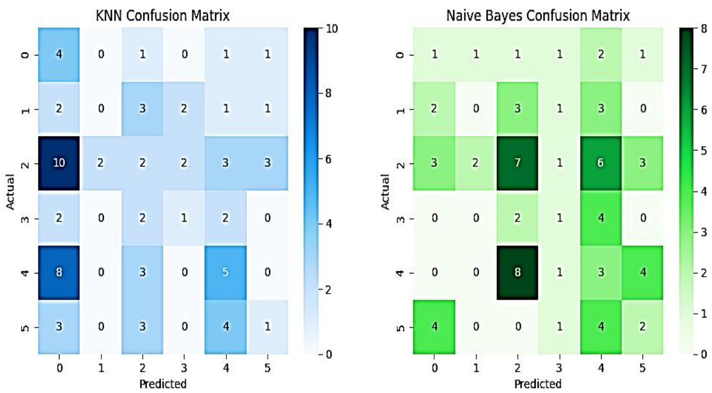

Figure 12: Confusion Matrix Evaluation of KNN and Naïve Bayes

Confusion Matrix of the Hybrid Model (Soft Voting Ensemble)

Figure 13 is the confusion matrix result of the developed hybrid model. It performed strongly with the majority of prediction falling on the diagonal (correct). The confusion matrix suggests: high accuracy, low misclassification, stable class separation. Classes C, E, and F are predicted well but slightly less perfectly.

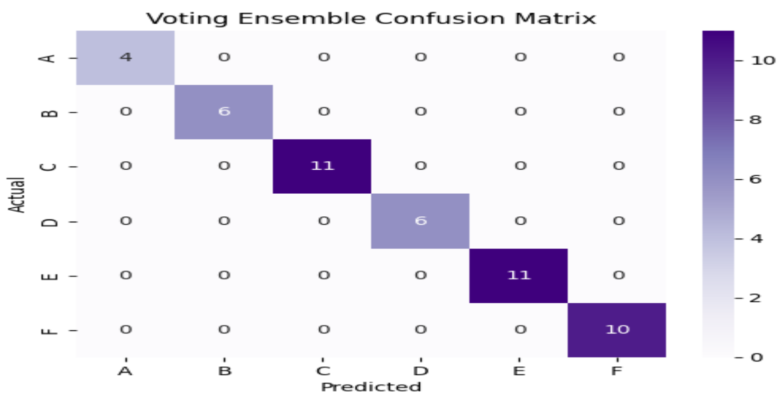

Figure 13: Confusion Matrix Evaluation of Hybrid Model (Soft Voting Ensemble)

high	Good performance — aim higher with advanced readings and peer-teaching.
low	Attend remedial classes, meet your lecturer, focus on fundamentals, and seek coun
high	Good performance — aim higher with advanced readings and peer-teaching.
high	Good performance — aim higher with advanced readings and peer-teaching.
low	Attend remedial classes, meet your lecturer, focus on fundamentals, and seek coun
high	Good performance — aim higher with advanced readings and peer-teaching.
low	Attend remedial classes, meet your lecturer, focus on fundamentals, and seek coun
medium	Target weak topics, join study groups, and track progress weekly.
medium	Target weak topics, join study groups, and track progress weekly.
low	Attend remedial classes, meet your lecturer, focus on fundamentals, and seek coun

Table 14: Confusion Matrix Explanation of the Result

System Testing
Before a system is put into operation, its components programs must be tested to make sure it works as a separate entity and when integrated. System testing removes bugs from individual programs and system application. The testing of this system was done with training data set and test data set. Test was also done in both Windows 7 and 8 Operating System and it worked effectively.

Test Plan
After completion of the detailed design, the design team will develop a plan for testing the software developed from the design specifications. Data sets must be generated as part of the test plan, which will effectively be used to test all functions of the software and its data outputs as defined in the specifications. Matrices of the test data that define the known values of all input data elements and the expected output values for each data element will be developed and documented. The matrices will also contain cross references to the design specifications for each function to be tested. If new or unique data file and data structure is required for the testing, then two test data sets must be developed; one for use by the software developers for testing of their software and the other for the formal testing of the software under controlled conditions. The plan must cover all phases of the test to be performed including modular, integration, system, acceptance, and classification testing.

Unit/Module Testing: This is a level of software testing where individual units/components of software are tested. The purpose is to validate the performance of each unit of the software. The various units/modules have been tested and each has proved efficient as an entity.

Integration testing: This is done during and after integration of a new module into the main software package. This involves testing each individual code module. The essence of this integration is to check how the modules work when they are integrated into subsystem and in the main system. In other words, the test carried out here is to ascertain that those modules do not lose their efficiency and reliability (which has been proved in the module testing above) when integrated into subsystem and system.

Performance Testing: Testing to assure that response times, run times, and other phases of execution are within acceptable limits and time frames.

Test Data
Tests are conducted at all levels of the system development process by assigned review teams using approved test data plans and validated test data sets. Unit, modular, integration, system, acceptance, and classification data mining tests are conducted on all modules of the system. Testing is done during system development, system implementation and each time a change is subsequently made in the system.

Actual Test Result versus Expected Test Result
The Student Performance Prediction System met the expected objectives as shown in Table 5.4. It successfully analyzed student data using data mining and multi-agent techniques**,** accurately predicting students' performance (*Low*, *Average*, *High*) based on predefined rules. The results matched the expected outcomes, as the system generated correct predictions and relevant academic advice. The information was also made accessible through an interactive interface for further analysis by users.

Table 3:: Actual Test Result versus Expected Test Result

Module	Expected Test Result	Actual Test Result
Authentication and Registration	Expected to allow both administrators and students to register and log in successfully using valid credentials. Invalid credentials should be rejected with an appropriate error message.	The system successfully authenticated users with correct credentials and denied access when incorrect credentials were entered. Registration for both admins and students worked as expected.
Model Training and Prediction	In this module, it is expected that the prediction model evaluates student performance using the uploaded dataset. The trained model should classify students into categories such as *Low*, *Average*, and *High* based on their attributes.	The performance prediction model was successfully built using K-Nearest Neighbors (KNN) and Naïve Bayes algorithms. Predictions were accurately generated and validated using a confusion matrix. Categories such as *Low*, *Average*, and *High* were displayed correctly for tested records.
Advice Generation	When clicked on, it is expected that the system automatically generates tailored academic advice based on the predicted performance category (e.g., study harder for "Low", maintain progress for "High").	The advice generation module produced appropriate recommendations corresponding to each prediction category. Advice was displayed instantly and stored in the history database.
History Management	It is expected to allow administrators to view, download, and manage the prediction and advice history of all students. The data should also be backed up to CSV automatically.	The system correctly retrieved prediction records from MongoDB, displayed them in tabular format, and allowed exporting as CSV. History was properly updated and stored in the backup file.
Student Feedback Handling	It is expected to enable students to view their prediction results and advice, and to submit feedback or replies to administrators through the portal.	The system displayed student-specific advice records successfully. Feedback submitted by students was saved to the database and appeared in the admin's "View Student Replies" section.

CONCLUSION

In this study**,** a hybrid classifier combining Naïve Bayes and K-Nearest Neighbour (KNN) algorithms was developed to predict students' academic performance in three categories: High**,** Average**,** and Low. The dataset was collected from third-year students of Nursing Science, Abia State University, Uturu, through a structured questionnaire designed to capture their personal, social, and academic information. The hybrid model was trained and tested using 240 student records**.** Out of the total predictions, 53% of students were classified as High performers**,** 36% as Low performers, and 11% as Average performers. The system achieved a prediction accuracy of 94.6**%**, outperforming the individual models, where Naïve Bayes achieved 90.3% and KNN achieved 88.9%. The hybrid model also executed faster, with an average processing time of 0.42 seconds**,** compared to 0.58 seconds for Naïve Bayes and 0.64 seconds for KNN. This research has demonstrated that Machine Learning (ML) can play a significant role in educational institutions' decision-making and knowledge management processes. The study has established that student-related data available to higher institutions can be effectively utilized to predict academic performance using intelligent data mining techniques.

REFERENCES

Abubakar, A. B., Gabi, D., Garba, M., Dankolo, N. M., & Hassan, A. (2025). Hybrid Predictive Model For Students' Academic Performance Based On Machine Learning Approach. *FUDMA Journal of Sciences*, *9*(4), 215-222. https://doi.org/10.33003/fjs-2025-0904-3004

Al-Barrak, M. A., & Al-Razgan, M. (2016). Predicting students' final GPA using decision trees: A case study. *International Journal of Information and Education Technology*,*6*(7), 528-533.http://doi.org/10.7763/IJET.2016.V6.745

Amin. Z., Refik, C., Yau, H., & Hernandez-Torrano, D., (2017). Predicting Students"GPA and Developing Intervention Strategies Based on Self-Regulatory Learning Behaviors. 2017, IEEE

Amrieh, E., Hamtini, T., & Aljarah, I., (2016). Mining educational data to predict student'cademic performance using ensemble methods. *International Journal of DatabaseTheory and Application* 9(8), 119-136.

Baker, R. S., & Inventado, P. S. (2014). Educational data mining and learning analytics. In J. A. Larusson & B. White (Eds.), *Learning analytics* (pp. 61–75). Springer. https://doi.org/10.1007/978-1-4614-3305-7_4

Baradwaj, B., & Kumar, B., (2011). Mining Educational Data to Analyze Students Performance.*International Journal of Advanced Computer Science and Applications (IJACSA*), 2(6), 2011, 63-69

Bunkar, K., Sharma, C., Singh, U. P., & Bunkar, M. (2012). Data mining: Prediction for performance improvement of graduate students using classification. *International Journal of Computer Applications, 41*(3), 1–5. https://doi.org/10.5120/5594-7670

Devasia T, Vinushree T., & Hegde V., (2016): "Prediction of Students Performance using Educational Data Mining", 2016, IEEE

Dewan M., Zhang L., Rahman C., Hossain A., & Strachan R., (2014). Hybrid decision tree and Naive bayes classifiers for multi-class classification tasks. Elsevier, Expert systems with applications, pp) 1937-1946, 2014

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *International workshop on multiple classifier systems* (pp. 1–15). Springer.

Gray, S., & Rogers, M., Martinussen, R., & Tannock, R. (2014). Longitudinal relations among inattention, working memory, and academic achievement: Testing mediation and the moderating role of gender. PeerJ. 3. E939. 10.7717/peerj.939

Khan, M. Z., Wasid, M., & Jabin, S. (2019). Machine learning algorithms in analyzing and predicting students' performance: A review. *Procedia Computer Science, 172,* 282–289. https://doi.org/10.1016/j.procs.2020.05.037

Kabakchieva, D., (2013). Predicting Student Performance by Using Data Mining Methods for Classification. Cybernetics and information technologies Volume 13, No 1 Sofia 2013 Print ISSN: 1311-9702; Online ISSN: 1314-4081 DOI: https://doi.org/10.2478/cait-2013-0006

Kotsiantis, S., Pierrakeas, C., & Pintelas, P. (2004). Prediction of Student's Performance in Distance Learning Using Machine Learning Techniques. Applied Artificial Intelligence,Vol. 18, No. 5, 2004, pp. 411-426

Merchan Robiano, S.M., & Duarte Garcia, J.A. (2016). Analysis of data mining techniques for constructing a predictive model for academic performance. IEEE Latin America Transactions, 14(6), 2783-2788

Minaei-Bidgoli, B., Kashy, D., Kortemeyer, G., & Punch, W. (2003). Predicting student performance: An application of data mining methods with an educational web-based system. Proceedings of the 33rd Annual Conference on Frontiers in Education, Nov. 5-8, IEEE Computer Society, Washington, DC, USA., pp: 13-18

Nakayama, M., Kouichi, M., Hiroh, Y. (2018). Contributions of Student's Assessment of Reflections on the Prediction of Learning Performance. 2018, IEEE

Naren. J. (2014). Application of Data Mining in Educational Database for Predicting Behavioral Patterns of the Students. *International Journal of Computer Science and Information Technologies*, Vol.5 No.03 2014.

Nikam, S. (2015). A Comparative Study of Classification Techniques in Data Mining Algorithms. *Oriental Journal of Computer Science and Technology; 8*(1), 2015, ISSN 0974-6471 Online ISSN: 2320-8481

Nikolovski, V., Stojanov, R., Mishkovski, I., Chorbev, I., & Madjarov., G. (2015).Educational Data Mining: Case Study for Predicting Student Dropout in Higher Education. https://www.researchgate.net/publication/282333827.Conference Paper.April2015187

Omisore, O., & Azeez, N. (2016). Predicting Academic Performance of Students with KNN Classifier. Conference: ACM International Conference on Computer Science Research and Innovations (CoSRI 2015).

Osmanbegovic, E., & Suljic, M., (2012). Data mining approach for predicting studentperformance. *Economic Review Journal of Economics and Business*. Volume 10(1), 2012

Pandey, M., & Taruna, S. (2014). An Empirical Analysis of Classification Techniques for Predicting Academic Performance. https://doi.org/10.1109/IAdCC.2014.6779379.188

Paris. I., Affendey, L., & Mustapha, N. (2010). Improving academic performance prediction using voting technique in data mining. World Academy of Science, Engineering and Technology, vol. 4, pp. 820--823, 2010

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), pp.601–618

Safri, Y.F., Arifudiri, R., & Muslim, M.A. (2018). K-Nearest Neighbour and Naïve Bayes classifier algorithm in deterging the classification of Health Indonesia CardRecipients.Scientific*JournalofInformatics,5(1)*. https://doi.org/10.15294/sji.v5i1.12057

Sivasakthi, M. (2017). Classification and Prediction based Data Mining Algorithms to Predict Students‟ Introductory programming Performance. Proceedings of the International Conference on Inventive Computing and Informatics (ICICI 2017)

Ulyani, N., Mohd, N., Nor-Aini, Y., & Amin A. (2017). Service Quality Performance of Student Housing: The Effects on Students Behavioural Intentions. 2017 IEEE 15th StudentConference on Research and Development (SCOReD)

Ward, A., Howard, S., & Murray-Ward, M. (1996). Achievement and Ability Tests –definition of the Domain. Educational Measurement, 2, University Press of America, pp. 2–5, ISBN 978-0-7618-0385-0

Yang, F., & Li, F.W. B (2018). Study on student performance estimation, student progress analysis, and student potential prediction based on data mining. Computers&Education.123, 97-108. https://doi.org/10.1016/j.compedu.2018.04.006

Zhou, Z. H. (2012). *Ensemble methods: Foundations and algorithms*. CRC Press

Ziedner, M. (1998). Test anxiety: The state of the art". New York: New York: Plenum Press. p. 259. ISBN 9780306471452. OCLC 757106093 194.