J. Electrical Systems 20-3s (2024): 899-919

[1]Duo Chen

[2]Fang Xu

Design and Implementation of Machine Learning Algorithms in Automatic Grading of Students' Assignments

Abstract: - This study introduces machine learning methods for automatically grading students' assignments, catering to the changing demands of education in the digital age. By utilizing diverse techniques, the aim is to simplify the grading process, give prompt feedback, and improve overall efficiency in assessment practices. The research delves into the different types of assignments such as written responses and coding tasks, and presents a range of ML algorithms including NB, Decision Trees (DT), RF, SVM, Linear Regression (LR), KNN, and Ensemble Methods (EMs). The implementation involves extracting relevant features from assignments, preprocessing data to handle noise and outliers, and training models on a varied set of examples. The grading system prioritizes interpretability, accuracy, and efficiency while aligning with educational objectives and grading policies. Evaluation is done using suitable metrics like recall, accuracy, precision, F1 score for each algorithm. The study contributes to the progress of automated grading systems by providing valuable insights for educators on the potential impact of machine learning in enhancing assessment processes. Additionally, the designed algorithms demonstrate adaptability for different types of assignments and serve as a foundation for future enhancements in grading precision and efficiency.

Keywords: Machine Learning Algorithms, Automatic Grading of Students, Naive Bayes (NB), Decision Trees (DT), Random Forest (RF), Support Vector Machines (SVM), Linear Regression (LR), K-Nearest Neighbors (KNN), and Ensemble Methods (EMs)

I. INTRODUCTION:

Assessing the achievements of students is a crucial aspect of the education system. It provides important feedback for educators, students, parents, institutions, and policymakers, allowing them to evaluate the effectiveness of instruction. Large-scale assessments of student knowledge play a significant role in this evaluation process and can drive educational reforms and policy changes. National assessments, in particular, they have significant impact on country education system. In this discussion, we will focus on national assessments specifically related to mathematical knowledge. Understanding the factors that may affect student performance is essential in this context. Predictive investigations can provide strategic insights into quality of teaching and learning by identifying which variables can predict educational outcomes [1]. These predictive studies are valuable tools for understanding how factors like school environment, teaching methods, individual student characteristics influence academic success. The results of predictive studies can also be useful for policymakers in making informed decisions about educational policies.

In recent times, the transformation of procedures in every aspect of public life into digital formats has sparked an increased interest in researching how to utilize the gathered data for effective management decision-making. This holds particularly true for higher education institutions (HEIs) that currently depend on a range of software systems. These systems, including those for learning management, student information, human resources, research reporting, and student admission, serve as repositories for extensive data on both students and staff members. These systems track their activities and enable the use of technology to extract valuable knowledge. The overwhelming growth of this stored data presents a challenge for HEI managers to incorporate it into their decision-making processes and ultimately enhance the quality of services provided by the institution [2]. With this comprehensive data collection, administrators can create data-driven policies and decisions, providing the foundation for the development of artificial intelligence-driven software. Such software can optimize the ongoing processes within HEIs.

Flipped classrooms (FC) unique teaching approach have reverses traditional method by allowing for more efficient use of in-class time. In traditional classrooms, students often come to class unprepared and the teacher

[1]Business English Department, School of Foreign Languages, Harbin University, Harbin, Heilongjiang, China, 150086

[2]Business English Department, Harbin Finance University, Harbin, Heilongjiang, China, 150030

Corresponding author's email: franki007@163.com

must spend valuable time introducing basic concepts. This leaves little time for engaging in-class activities, and more complex tasks are left to be completed as homework. However, with the FC method, these limitations are eliminated [3]. By presenting various resources before class, students can learn the necessary concepts at their own pace, considering their individual needs. At the start of class, a brief quiz assesses students' understanding and allows for feedback from both teacher and peers. This leaves more time for in-class activities that involve critical thinking and problem-solving. The teacher takes on a guiding role while students actively participate in their own learning.

At present, there has been progress in forecasting student grades, but there are still challenges that need to be addressed. One issue is that grade prediction is typically done too late, leaving little time for students and teachers to make necessary changes to their study methods or provide intervention. This means that teachers can only offer assistance to struggling students and cannot focus on top-performing students. Additionally, student grade prediction typically only predicts the final grade and does not track progress throughout a semester. The absence of this information makes it challenging for teachers to intervene and provide accurate and effective guidance to students [4].

The particular scenario, use of ML algorithms has the potential to enhance academic achievement. Through the examination of students' individual performance data, these algorithms have the capability to identify their strengths and weaknesses and provide personalized recommendations for improving their academic success. For instance, a machine learning algorithm can analyze a student's previous exam scores, homework assignments, and class participation records to pinpoint areas where they may need additional support. It can then suggest specific learning materials or areas of focus to aid the student in their progress [5]. Additionally, machine learning algorithms can facilitate adaptive testing by adjusting the difficulty level of questions according to a student's performance. This approach ensures that students are adequately challenged, leading to a more precise evaluation of their comprehension and ultimately enhancing their academic achievement. Furthermore, these algorithms have the capacity to analyze large amounts of educational data, such as student records and demographic information, in order to uncover patterns and insights that can inform educational policies and practices. For example, through analyzing a school's student population's performance data, a machine learning algorithm can identify areas where additional support may be needed. This information can then be used to implement targeted interventions like tutoring or mentoring programs for students who may be struggling [6].

Machine learning is effective with limited data, but not with vast amounts of information. There are three main techniques used to train the model. In supervised machine learning, a human supervisor and existing data are necessary for learning from the data. Unsupervised machine learning can occur without human oversight. We use of reinforcement in ML is declining. These algorithms utilize past data to make accurate decisions. A growth of AI can be attributed to advancements in machine learning, as it relies less on human programming and allows for self-learning machine to analyze data and complete tasks [7]. This can be advantageous in decision making and evaluating students' ability to select a major and have confidence in choices.

RSS have developed to assist students in determining their desired field of study. This approach can greatly enhance a student success in the chosen discipline. The utilization of AI and ML techniques allows for the customization of RSS based on factors such as a student's knowledge level, skills, gender, work experience, and learning style. While AI is defined as the replication and improvement of human intelligence through artificial means, important decisions are not solely entrusted to automated machines. Instead, algorithmic knowledge is utilized to solve specific problems. For instance, RSS can generate life insurance policies, monitor real-time systems, perform task take previously required human expertise. Furthermore, these systems are highly valuable to students as they aid in selecting a university specialization by analyzing their individual traits and introducing they to market demand [8].

The purpose of this system is to aid students in making well-informed choices for their own benefit. The integration of expert systems into educational guidance has significantly improved learning outcomes, specialized training, student achievement, and self-assessment. Our study aims to introduce a smart recommendation system that takes into account previous knowledge and data as well as gender to assist students in selecting the most suitable college major for them. To achieve this goal, we employed various ML algorithms including the NB, DT, RF, SVM, LR, KNN, and EMs. This paper presents two primary contributions: (1) identifying the most precise ML classifier for predicting student major selection using the features mentioned above, and (2) determining the most influential features in predicting student majors. The following sections structure the paper: Section 2 reviews previous research on this topic; Section 3 covers the fundamentals of the ML algorithms utilized in this study; Section 4 outlines the proposed methods; Section 5 presents the results with a comprehensive discussion; and finally, Section 6 summarizes the findings and highlights future research directions.

II. RELATED WORKS:

In [9], a study on a flipped anatomy course demonstrated a predictive model using the Naïve Bayes algorithm, achieving an overall classification accuracy of 68% and identifying at-risk students with 71% accuracy. This model assists in targeted interventions for students facing challenges, thereby enhancing the effectiveness of the course. In [10], the research identified key predictors of school bullying victimization among primary and secondary school students, such as teacher-student relationships, peer relationships, and family cohesion. These findings have the potential to guide targeted interventions and enhance the effectiveness of anti-bullying measures in schools. In [11], the study focused on the use of ML algorithms to predict and understand declining student performance during the COVID-19 era, especially in the context of the rise of e-learning. It sheds light on the impact of habits such as sleep, study time, and screen time on academic success, drawing comparisons between online and offline learning data for valuable insights. In [12], a multi-output hybrid ensemble model was developed to predict student grades, leveraging data from an educational platform. This approach combines various ML techniques to provide a comprehensive prediction framework for student performance. By achieving higher accuracy in predicting mid-term and final grades, and lower errors in homework and experiment grades, it aims to enhance student learning outcomes and teaching effectiveness in blended learning environments. In [13], this study employs SVM and K-means clustering to identify distinct student clusters based on holistic characteristics and learning outcomes. By developing prediction models for each cluster, it enhances accuracy in predicting student performance, surpassing traditional semester-grade-based methods. In [14], two-phase ensemble classification model to predict learning success in MOOCs, leveraging machine learning to identify low-performance students and optimize learning outcomes. By integrating silhouette score-based feature selection and Bayesian optimization, this method surpasses current algorithms in predicting learner grades, thereby improving the effectiveness of online learning.

In [15], utilizes supervised machine learning techniques to predict undergraduate majors, focusing on academic history and job market factors. By applying hyper-tuning, it outperforms previous research, with random forest achieving 97.70% accuracy and identifying key features like degree percentage and entry test results for program recommendations. In [16], An innovative federated transfer learning framework is introduced for student classification, aiming to tackle privacy concerns and address data inconsistency within the realm of educational data mining. By introducing domain adaptation and federated learning, it effectively handles heterogeneous datasets, demonstrating improved student grades classification in small data environments. In [17], semi-regression algorithm for predicting undergraduate students' final exam grades in an online course, leveraging three distinct feature views. Through experiments, it demonstrates improved early prognosis of at-risk students compared to traditional supervised models, offering potential benefits to the educational domain. In [18], A machine learning model is developed to predict students' retention and graduation status in higher education. Utilizing key features like Average high school grade and Entry score, RG-DMML achieves high accuracy rates, aiding in timely interventions and sustainable educational practices. In [19], this study investigates factors influencing student performance using Pearson correlation, highlighting past failures (-0.36) and mother's education (0.22). Machine learning models, especially MLP 12-Neuron, accurately predict student grades, confirming the impact of these factors.

In [20], project-based ML course for high school students, improving the understanding of AI concepts and computational thinking skill. Despite minor changes in learning attitude, the course positively impacts students' perception of AI education, laying a foundation for future AI curriculum development in K-12 settings. In [21], an intelligent system using the XGBoost algorithm to predict academic failure, showing superior performance and highlighting the importance of economic, health, and social factors in students' academic performance. In [22],A semi-regression algorithm is utilized to predict undergraduate students' final exam grades in an online course, leveraging three distinct feature views. Through experiments, it showcases improved early prognosis of at-risk students. In [22], the study predicts middle- and high-school students' academic performance using ML algorithms. It discovers that a health-conscious lifestyle positively correlates with academic success, while stress has a negative impact. Gender is not a significant predictor in the model. The earlier model analysis is described in table 1.

Table 1 – Earlier Model Analysis

Ref. No	Algorithm	Methodology	Advantages	Disadvantages	Performance	Accuracy	Features Used	Measurements Applications	Energy Results
[9]	Naïve Bayes	Predictive model for at-risk students in anatomy course using interaction metrics, quizzes	Effective for predicting student grades, especially at-risk students	Dependent on data quality and assumptions	68%	71%	Weekly interaction metrics, quiz scores, pretest scores	Anatomy course at-risk student prediction	N/A
[10]	GBDT	Examining factors predicting school bullying victimization among students	Protective factors assciated and Identifies risk with bullying victimization	Requires quality data, interpretation may be complex	N/A	N/A	Individual, family, peer, and school environment factors	Predicting school bullying victimization	N/A
[11]	Various (SVM, Decision Tree, etc.)	Predicting student performance based on online and offline learning properties	Predicts declining student performance, compares online/offline data, identifies factors affecting performance	Requires normalization of databases, algorithm selection, interpretation of results	N/A	N/A	Various metrics including study habits such as sleep, screen time, study time	Predicting student performance and interventions	N/A
[12]	Multi-output hybrid ensemble model	Predicting grades using Superstar Learning Communication Platform (SLCP) data	Improves grade prediction accuracy, addresses issues of lag, enhances predictive attributes	Requires SLCP data, model training, and tuning, may be complex to implement	78.37%	3-8% higher than comparison models	SLCP data for mid-term, final grades; GBDT model for homework, experiment grades	Improving student learning quality and teaching effectiveness	N/A

[13]	SVM, K-means clustering	Clustering students into performance clusters and predicting performance holistically	Identifies student clusters based on performance, predicts performance holistically	Requires classification and clustering methods, feature selection	Higher accuracy than using semester grades	N/A	Features forming clusters for low, average, and high performance	Predicting student performance holistically	N/A
[14]	Ensemble method	Predicting learning success in MOOCs using two-phase classification model	Predicts learner grades in MOOCs, improves learning through grouping and intervention methods	Utilizes ensemble method, requires computation of new features, Bayesian optimization	Outperforms state-of-the-art algorithms	N/A	Silhouette score-based feature selection, Bayesian optimization	Predicting learning success in MOOCs	N/A
[15]	Decision Tree, Random Forest, SVM	Predicting undergraduate majors based on academic history and job market data	Recommends undergraduate programs based on academic and job market data, improves on previous research	Requires hyperparameter tuning, dataset availability and quality	97.70%	N/A	Academic history, job market data	Recommending undergraduate majors	N/A
[16]	Federated transfer learning	Student classification using privacy-protected federated transfer learning framework	Addresses privacy issues, enables classification in small datasets	Requires federated learning environment, domain adaptation method	N/A	N/A	Features for student classification	Student classification with privacy protection	N/A
[17]	Co-training	Predicting grades of undergraduate students in online course using multi-view learning	Accurately predicts at-risk students, utilizes multi-view learning for robust models	Limited use in educational data mining, requires interpreting features	Accurate prognosis of at-risk students	N/A	Three independent feature views from online course data	Predicting grades of online course students	N/A
[18]	RG-DMML (Ensemble	Graduation status and Predicting students'	Efficient prediction model for graduation	Requires suitable features for	High Precision, Recall, Accuracy,	N/A	Student data on retention and	Predicting students' retention and	N/A

	Algorithm)	retention in education	status, retention, validated with high performance metrics	constructs, model tuning, and validation methods	F1-Score		graduation, features suitable for constructs	graduation status	
[19]	Pearson correlation, MLP, Random Forest, Decision Tree	Identifying factors affecting student performance, predicting grades	Identifies factors affecting student performance, predicts grades based on factors, compares ML models performance	Requires correlation analysis, model training and evaluation	RMSE values: MLP (4.32), Random Forest (4.52), Decision Tree (5.69)	N/A	Factors affecting student performance	Predicting student grades	N/A
[20]	Supervised learning (ML)	Project-based ML course for high school students on AI technology	Improves understanding of AI concepts, enhances computational thinking skills	Limited impact on learning attitude, need for tailored learning scaffolding	N/A	N/A	AI concepts, computational thinking, learning attitude	Implementing ML course for AI education in high schools	N/A
[21]	XGBoost	Predicting academic failure using student data from university	Superior performance in predicting academic failure, feature importance analysis	Requires XGBoost algorithm, feature extraction, oversampling	High performance prediction model	N/A	Student information features from university data	Predicting academic failure in university students	N/A
[22]	MLAs (Logistic Regression, ANN, Random Forest, Gradient Boosting, Stacking)	Predicting middle- and high-school student academic performance	Gradient boosting outperforms other methods, identifies lifestyle factors affecting performance	Requires data on socio-demographic, school-related, and student-related variables	N/A	N/A	Socio-demographic, school-related, student-related variables	Predicting middle- and high-school student performance	N/A

III. FUNDAMENTALS OF MACHINE LEARNING ALGORITHM:

3.1 Naive Bayes:

Naive Bayes is a machine learning method that makes use of Bayes' Theorem and relies on probabilities. Its main purpose is for classifying data. The "naive" aspect of Naive Bayes is its assumption that features are unrelated to one another, given the assigned class. Despite this simplification, Naive Bayes tends to yield positive results in real-world scenarios, particularly with tasks such as identifying spam and analyzing sentiment in text.

Bayes' Theorem: A hypothesis data in this theorem calculates the probability. Represents as,

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)} \tag{1}$$

Naive Assumption: The "naive" aspect stems from the belief that all characteristics are unrelated when considering the class. Therefore, the likelihood of a group of features X occurring given class C can be computed as:

$$P(X|C) = P(x_1|C).P(x_2|C)...p(x_n|C) \tag{2}$$

Classification: To categorize a fresh example, Naive Bayes computes the likelihoods of every category based on the characteristics and chooses the one with the greatest likelihood.

3.2 Decision Trees (DT):

Decision Trees are widely used and easy to understand method for classifying and predicting data. They construct a diagram that resembles a flowchart by analyzing the characteristics of the dataset. The nodes within the tree represent evaluations on specific attributes, the branches indicate the results of these evaluations, and the end nodes indicate the classification or numerical value. DTs are a widely used and easy-to-understand method for classifying and predicting data. They construct a diagram that resembles a flowchart by analyzing the characteristics of the dataset. The nodes within the tree represent evaluations on specific attributes, the branches indicate the results of these evaluations, and the end nodes indicate the classification or numerical value. DT algorithm as describe in Figure 1.

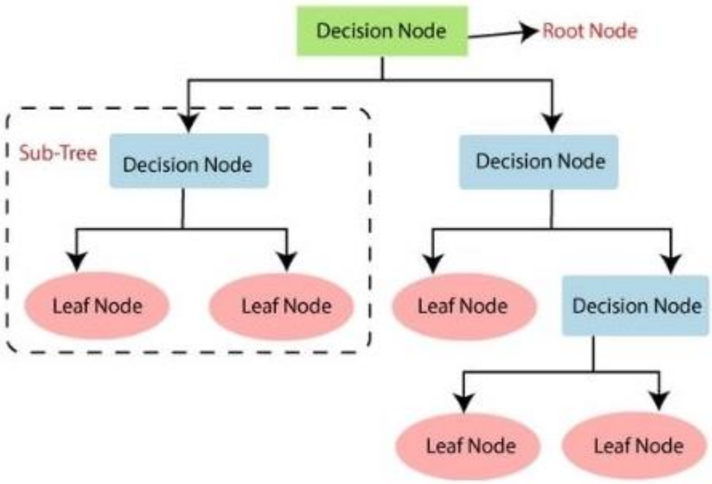

Figure 1 - Decision Tree Algorithm

Measures for Selecting Attributes: In order of determine most suitable attribute for splitting the data, DTs utilize various measures, including:

- Information Gain: This metric quantifies the amount of information a feature provides about the class. It calculates the level of uncertainty (entropy) before and after the split.
- Gini Impurity: Used in classification, this measure evaluates the likelihood of incorrectly classifying a randomly chosen class based on its random labeling according to the class distribution in a particular node.

- Gain Ratio: An improvement on Information Gain, this measure takes into account the number of branches produced by a split.

Making predictions with a Decision Tree is a straightforward process:

- Start at the root node and proceed down the tree by following the path determined by the feature values.
- When reaching a leaf node, use its associated class label (or regression value) as the predicted outcome.

3.3 Random Forest (RF):

RF is type of ensemble learning is built upon DT's. RF is commonly utilized in machine learning for tasks that involve classification and regression due to its adaptability. Instead of relying on a single decision tree, RF constructs a collection of trees and combines their predictions, overcoming the constraints of using just one DT. In this conversation, we will examine how RF operates, its advantages, and the potential applications for it. The ultimate prediction is determined through majority voting, where each tree "casts a ballot" for a class and the class with the most votes is selected as the predicted class.

a) Ensemble Learning:

Random Forest fall under category of ensemble learning techniques, which utilize multiple machine learning models to enhance performance and prevent over fitting. Instead of relying on a single Decision Tree, Random Forest constructs a group of trees and combines their predictions.

b) Constructing a Random Forest: Here is an overview of how a RF is built:

- Random Sampling: From a dataset with N samples and M features, RF randomly selects a subset (with replacement) to construct each tree. This process is known as bootstrapping.
- RF Selection: At each node of the DT, RF only considers random subset of features to determine the best split. This introduces diversity among the trees.
- Growing Trees: Each tree in RF is grown to its maximum depth without pruning, allowing it to learn as much as possible from the data.
- Voting: For classification tasks, the final prediction is determined through majority voting. Each tree "votes" for a class, and the predicted class for the input data point is the one that receives the most votes.
- Averaging: In regression task, final prediction is calculated by taking average of all individual tree predictions. A RF algorithm as explained in figure 2.

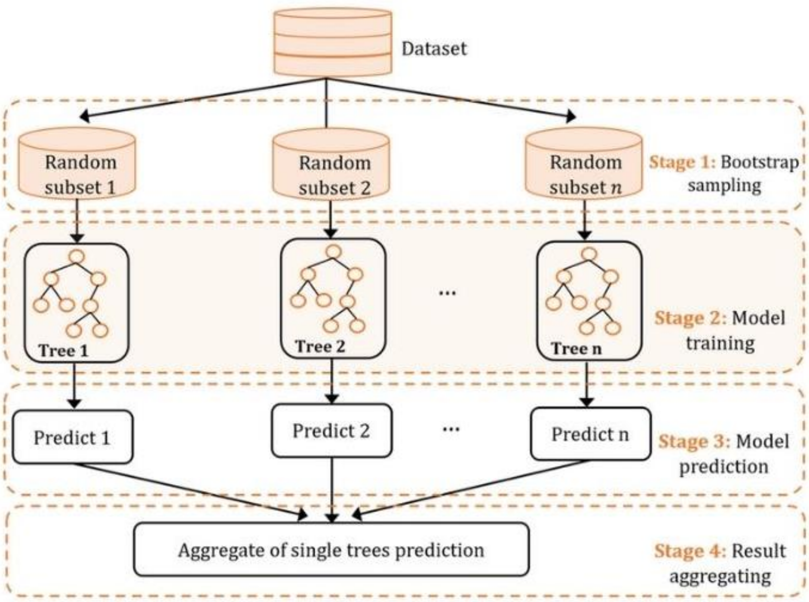

Figure 2 - Random Forest Algorithm

Random Forest has a variety of applications, such as credit scoring, healthcare for disease and patient outcome prediction, finance for stock market prediction, image classification, remote sensing, text mining and sentiment analysis, anomaly detection, and recommended systems. To optimize performance, Random Forest has several hyper parameters that can be adjusted, including the number of trees, maximum depth of trees, minimum samples required to split a node, minimum samples required at each leaf node, and maximum number of features to consider at each split. RF is a potent ensemble learning algorithm that overcomes the drawbacks of a single DT. Renowned for its high accuracy, capacity to handle complex datasets, and resistance to over fitting, RF delivers robust and dependable results across diverse machine learning tasks by combining predictions from multiple tree.

3.4 Support Vector Machines:

SVMs stand out of potent supervised learning tools applied in both classification and regression scenarios. They excel notably in high-dimensional spaces and are favored for their capability to manage intricate datasets. SVMs strive to identify the most fitting hyperplane that distinctly divides classes within a dataset. This discussion will explore the inner workings of SVMs, focusing on hyperplanes, the kernel trick, and the merits and drawbacks they entail. Table 2 describe about the SVM algorithm in detail.

Table 2 – Details of SVM Algorithm

Aspect	Description	Notes
Algorithm Type	SL algorithm for both regression tasks and classification	NIL
Optimal Hyperplane	Seeks to find the hyperplane with maxi margin between class	Support Vectors: Data points closest hyperplane, influencing its position
Kernel Trick	Transform data into higher-dimensional space for non-linear separability	Avoids computational expense of explicit transformation
Kernel Functions	Linear, Polynomial, Gaussian (RBF), etc.	Adjust Regularization and Kernel Coefficient for best results
Soft Margin SVM	Introduces slack variables for data that is not perfectly separable	Balances margin maximization and error minimization with regularization parameter C
Advantages	Robust to Over fitting, especially with proper C selection	Efficient Memory Usage with Support Vectors
Disadvantages	Computationally Intensive, especially with large datasets	Hard to Interpret Hyperplane
Accuracy & Performance	Provides high accuracy in complex datasets	Can handle large feature spaces effectively
Efficiency	Can be memory efficient by using support vectors	Kernel computation can be time-consuming
Feature Uses	Handles both numerical and categorical data	Extracts the most relevant features for classification
Applications	Text Classification (e.g., spam detection)	Image Classification (e.g., object detection)

The concept of Non-Linear Separation and Kernel Trick involves the use of Support Vector Machines (SVMs) to handle data that cannot be separated by a straight line. This is achieved through a technique called the "kernel trick," which involves transforming the input space into a higher-dimensional representation. By doing so, SVMs are able to identify a hyperplane that can effectively separate the data classes, even in cases where they are not linearly separable in the original space. The kernel function plays a crucial role in this process as it calculates the dot product of two points in the higher-dimensional space without explicitly transforming the data into that space. Some commonly used kernel functions are polynomial, RBF, and sigmoid.Linear Kernel**:**

$$K(x, x' 2) = x^T x' \tag{3}$$

Polynomial Kernel: $K(x, x') = (x^T x' + c)^d$ (4)

Gaussian (RBF) Kernel: $K(x, x') = \exp(-\gamma \| x - x' \|^2)$ (5)

Kernel Trick Benefits: This approach enables SVMs to manage intricate, non-linear relationships within the data while circumventing the computational overhead of directly converting the data into higher dimensions. Figure 3 as briefly explained about SVM algorithm applications.

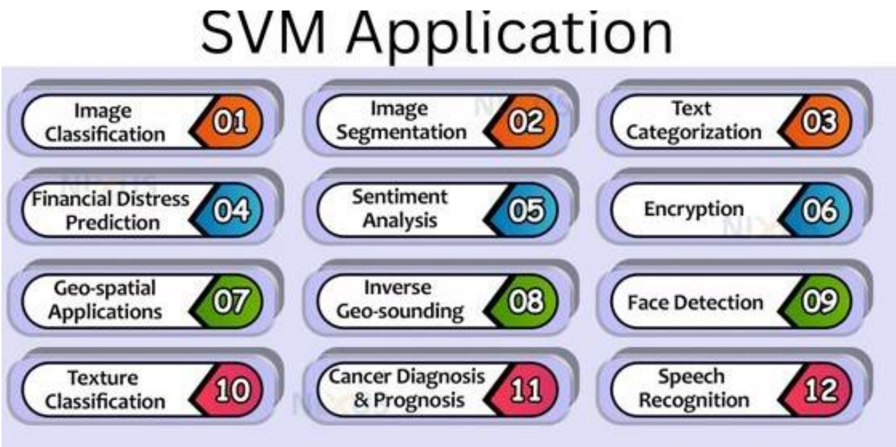

Figure 3 - SVM Application

3.5 Linear Regression (LR):

LR stands as one of the simplest yet highly utilized statistical and machine learning methods for forecasting a continuous variable using one or more input features. Its purpose lies in modeling the connection between a dependent Y and one or more independent (X). In this overview, we will explore the mechanics of LR, its underlying assumptions, the process of training, as well as its strengths and weaknesses.

3.5.1 Model Representation**:**

The fundamental concept of LR involves discovering the optimal linear equation that characterizes the connection between the input (X) and the output (Y). This model can be expressed as:

$$Y = \beta_0 + i\beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n + \in \quad (6)$$

The principles of LR: To ensure accurate and dependable findings, Linear Regression rests on a set of fundamental assumptions. One of these is that the connection between the dependent variable and the independent variables is linear, meaning that any alterations in the independent variables will have a proportional impact on the dependent variable. This assumption also implies that errors in predicting one data point are unrelated to errors in predicting another. In other words, the variability of the residuals should remain consistent across all levels of the independent variables. In simpler terms, this means that as the values of independent variables change, the spread of residuals should be uniform. Additionally, the residuals should adhere to a normal distribution with a mean of zero. This normality assumption is essential for statistical inference and hypothesis testing, ensuring that the estimates and confidence intervals derived from the model are accurate and reliable. These assumptions collectively form the foundation on which the Linear Regression model is built, guiding its application and interpretation in statistical analysis and predictive modeling tasks.

3.5.2 Training a LR Model:

This approach, also known as Linear Regression, allows us to capture the linear relationship between the input variables (X) and the output variable (Y) accurately. The primary aim of training a Linear Regression model is to find the most suitable values for the coefficients (β0, β1, ..., βn) that reduce the discrepancy between the predicted Y values and the actual Y values in the training dataset. This can be accomplished through a method called Ordinary Least Squares (OLS), which adjusts the coefficients to fit a line that minimizes the sum of squared differences between predicted and actual values. This effectively captures the linear association between input variables (X) and output variable (Y). OLS works by minimizing the squared vertical distances between each observed point and its corresponding predicted value, making it a reliable approach for Linear Regression. The formula for this objective function is expressed as follows:

$$\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \qquad (7)$$

3.5.3 Predictions:

Once the Linear Regression model is trained and the coefficients are estimated using the OLS method, we can use model to make predictions for new data point. This is accomplished by plugging the values of the input features into equation of linear regression model.

$$\hat{Y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n \qquad (8)$$

3.6 K-Nearest Neighbors (KNN):

KNN stand as a flexible, straightforward ML suitable for both classification and regression tasks. It belongs to the category of instance-based learning, where the algorithm stores the entire training dataset rather than constructing explicit models. The core idea behind KNN involves finding the "k" closest data points to a given query point in the feature space and using them to make predictions. This method relies on the similarity between data points, with closer points in the feature space deemed more alike. Table 3 discussed various types and advantages, disadvantages of the KNN algorithm.

Table 3 - Overview and some characteristic of KNN algorithm

Aspect	Description	Notes
Algorithm Type	Instance-based, lazy learning algorithm for classification and regression tasks	NIL
Basic Idea	- Predicts by finding K nearest data points to a given query point	- Assumes similar points have similar classifications/values
Training Phase	- Memorizes the entire training dataset	- No explicit model creation
Prediction Phase	- For Classification: Assigns most frequent class among neighbors as prediction	- For Regression: Takes average of neighbors' values as prediction
Distance Metrics	$\sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$ - Euclidean Distance	$(\sum_{i=1}^{n} \| x_i - y_i \|^p)^{\frac{1}{p}}$ Manhattan Distance
Choosing K	- Trial and Error: Trying different values and evaluating model performance	NIL
Advantages	- Simple and Intuitive: Easy to understand and implement	- No Training Phase: Suitable for online learning
Disadvantages	- Computationally Expensive: Time-consuming for large datasets	- Memory Intensive: Requires storing entire dataset
Accuracy & Performance	- Performs well with non-linear data and when decision boundaries are complex	- Sensitivity to K value and choice of distance metric
Applications	Handwritten Digit Recognition	- Medical Diagnosis based on Patient Data
Implementation Tips	- Feature Scaling: Ensure all features contribute equally to distance calculations	- Handling Categorical Variables: One-hot encoding for categorical features

3.7 Ensemble Methods (EMs):

Ensemble Methods (EMs) in machine learning refer to a set of techniques that amalgamate the predictions of numerous individual models to enhance overall performance. The primary concept driving ensemble methods is to harness the collective power of multiple models, aiming to generate a final prediction that is more resilient and precise than what any single model could achieve independently. By combining diverse models, ensemble methods often surpass the performance of individual models by mitigating issues related to bias, variance, or both. In our exploration, we will delve into the operational mechanics of Ensemble Methods, the various types available, the advantages they offer, as well as prevalent applications across different domains.

Ensemble Methods in machine learning operate by training multiple individual models on same data set, then combining the prediction to create a more accurate final prediction. The fundamental principle underlying ensemble learning is the concept of the "wisdom of the crowd," where the collective insights from diverse models often result in better overall predictions compared to any single model alone. Figure 4, EM's is explained in using models. There are two primary types of ensemble methods: Bagging and Boosting. Bagging, such as Bootstrap Aggregating, aims to reduce model variance by training multiple instances of base model on different subsets of training data. It involves randomly sampling subsets with replacement from data set, training base models independently, aggregating predictions through methods like majority averaging.

On the other hand, Boosting focuses on improving the performance of weak learners by training them sequentially and adjusting the weights of training instances based on errors. Boosting methods prioritize instances that were misclassified by previous models, gradually improving prediction accuracy. Additionally, Stacking, also known as Stacked Generalization, combines predictions from multiple base models using a meta-model or blender. Base models make predictions on the training data, and the meta-model learns to combine these predictions to generate the final prediction on new data. Ensemble methods offer several advantages, including enhanced performance through model diversity, robustness against outliers and noise, and the ability to capture complex patterns in data. They find applications in a wide range of fields such as finance for credit scoring, healthcare for medical diagnosis, business analytic for customer churn prediction, and cybersecurity for anomaly detection.

However, there are also drawbacks to consider. Ensemble methods can be especially with large datasets, computationally intensive, complex models. The final prediction of an ensemble may be less interpret able than that of a single model, and there is a risk of overfitting if the base models are overly complex or the ensemble is too large. Choosing the right ensemble method depends on the problem type, datasets characteristics, and performance metrics. Popular libraries such as Scikit-learn, XGBoost, and caret provide implementations of various ensemble methods, making it easier to apply these techniques in practice. In conclusion, Ensemble Methods offer a powerful approach to machine learning, leveraging strengths of multiple model to improve prediction accuracy, reliability. They are widely used across domains and can significantly enhance the performance of machine learning systems when appropriately selected and tuned, despite the trade-offs in computational complexity and interpret ability.

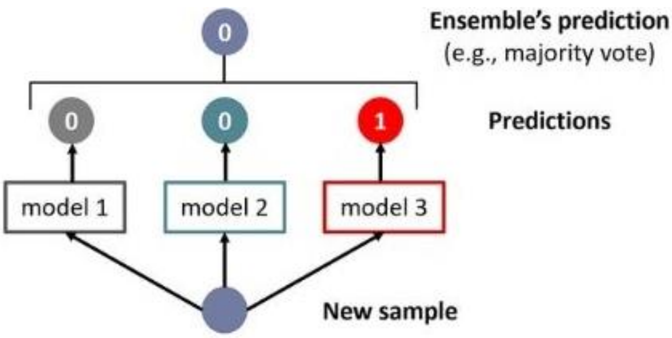

Figure 4 - EM's Algorithm using models

IV. PROPOSED AGS-MLS MODEL:

Developing a recommendation system to aid in choosing the ideal university major generally involves several essential stages. These stages include data gathering, data preparation and visualization, as well as the application of machine learning techniques. The research utilizes publicly accessible data from Kaggle, derived from a study of MBA students conducted at CMS Business School in January 2020. This data-set includes information on academic backgrounds and the job market, encompassing 216 student samples with 13 input attributes, and specialization as the target variable. The specializations are categorized into two fields: marketing and finance, and marketing and human resources.. This datasets was chosen due to its public accessibility and unique inclusion of both student academic performance and labor market status related to their specialization, indicating postgraduation employment and satisfactory salaries. Additionally, this datasets was previously used by, allowing for a comparison between our findings and theirs.

In this phase, the data was cleaned and prepared for future visualization and machine learning tasks. We used a Python module to apply various methods of preprocessing in order to clean and prepare the data. Our initial focus was on addressing any missing data, which we found to be present in 67 instances within the datasets. Specifically, we dealt with null values in the salary columns by replacing them with zeros, as they indicated students who had not yet been placed in a job after graduation. Next, we utilized the Label Encoder tool to convert categorical feature labels into numerical values, making it easier to apply machine learning techniques later on. To ensure consistency, we also performed data normalization using MinMaxScaler which scaled all values between 0 and 1 due to the presence of outliers in the datasets. Finally, the datasets were divided into training and testing sets at an 80:20 ratio, resulting in 173 samples. Figure 5 provides a detailed explanation of the workflow for our machine learning-based model for recommending majors at higher education institutions.

$$x' = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{9}$$

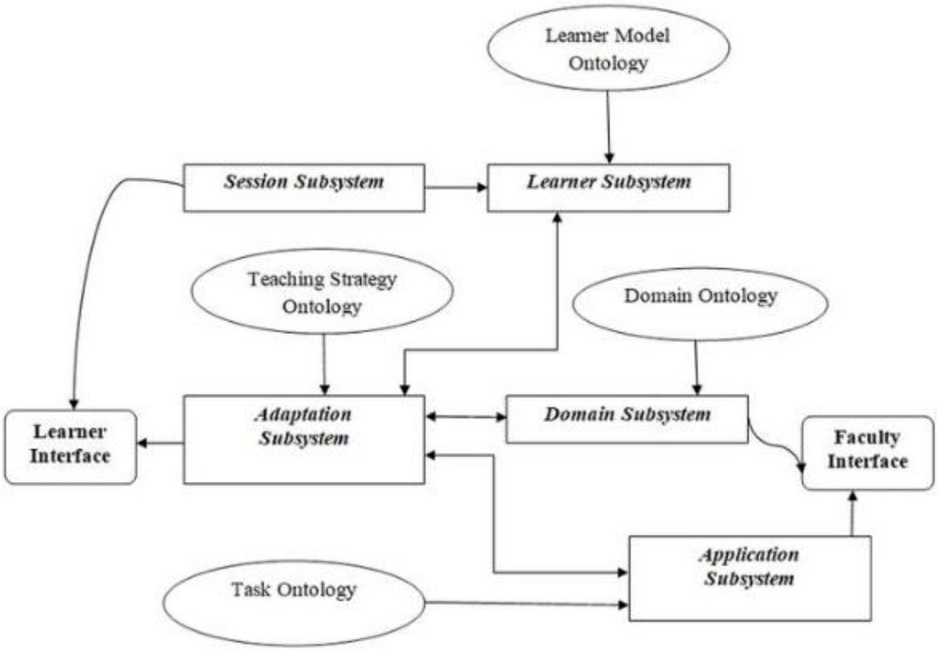

Figure 5 - The workflow of a ML model

Data visualization plays a crucial role in enhancing comprehension, aiding in discerning data quality, trends, relationships, model selection, and decision-making processes. From the visualization, we observe the numbers of male and female students enrolled in Marketing, Finance and Human Resources majors. Notably, female student count is lower than males in the sample, with a similar distribution across both programs, while males exhibit greater interest in Marketing and Finance. All remaining input attributes are deemed suitable for constructing a machine learning model. We employed three ML algorithms SVM, RF, DT alongside hyper-parameter tuning to optimize the parameters of the ML model for superior performance. The optimization of hyper parameters was conducted using the GridSearchCV technique. Hyper parameters play a crucial role in configuring algorithms and minimizing loss functions. GridSearchCV identifies the best-performing parameters by exploring predetermined values of the hyper parameters, ensuring a significant enhancement and better performance than random parameter selection.

Furthermore, GridSearchCV incorporates cross validation during training phase. A datasets is divided into testing data, training, with cross validation splitting training data into k subsets. During each iteration, one subset have reserved for testing while remaining k-1 subsets are used for model training. This process continues with each iteration using a different subset for testing until all subsets have been used. The model's performance is recorded at each step, and the average of these results is computed. Cross-validation is a widely adopted data resampling technique to assess the predictive model's ability to generalize and to prevent over fitting. It evaluates how well final model might perform on new data. DT algorithm is a supervised tree based algorithm used for both classification, regression task. Each path from root node to leaf node represents a sequence of data divisions, ultimately leading to a classification outcome at the leaf node. The splitting process primarily relies on information gain, which measures the knowledge acquired from variables in the datasets. DT hyper parameters include:

- Criterion: A function measuring the quality of splits.
- A Max_depth: maxi depth of the tree.
- A Min_samples_leaf: mini number of samples required to form a leaf node.
- A Min_samples_split: mini number of samples required to perform a split.

SVM is supervised learning method employed for tasks such classification, regression, and outlier detection. It operates by establishing a hyperplane or multiple hyperplane to segregate data, thereby assigning labels to training data based on optimal hyperplane, which in turn categorizes new samples. SVM's hyper parameters consist of the kernel, typically initialized to RBF, which was utilized for fine-tuning the C parameter, a regularization parameter, along with the Gamma parameter, representing the kernel coefficient. RF is supervised classifier beneficial for both regression, classification analyses. The concept behind RF involves building a series of decision trees using the training data and generating predictions based on highly accurate trees through a majority vote mechanism. RF demonstrates a strong classification rate is adept at handling outliers and noise, exhibiting lower susceptibility to over fitting. The hyper parameters for the RF algorithm include:

- Criterion: The function measuring the quality of splits.
- A Max_depth: maxi depth of each decision tree.
- A Min_samples_leaf: mini no. of samples required to form a leaf node.
- A Min_samples_split: mini no. of samples required to perform a split.
- A N_estimators: number of decision trees to construct within Random Forest.

V. SIMULATION RESULTS AND DISCUSSION

The proposed AGS-MLS model analysis includes the ML algorithms like NB [23], DT [24], RF [25], SVM [26], LR [27], KNN [28]. The analysis of these algorithm with respect to the parameters like MAE, MSE, RMSE and R^2 are given in table 4.

Table 4 - MAE, MSE, RMSE and R^2 Calculation

ML Algorithms	MAE	MSE	RMSE	R^2
Naive Bayes	0.5418	0.595	0.7716	0.86
Decision Trees	0.4073	1.2018	1.414	0.92
Random Forest	0.3982	0.8194	0.9052	0.94
Support Vector Machines	0.4194	0.506	0.7115	0.9
Linear Regression	0.4642	1.231	0.894	0.88
K-Nearest Neighbors	0.5068	0.511	0.7153	0.91
Proposed AGS-MLS	0.3504	0.2124	0.5158	0.93

5.1 MAE Calculation: The abbreviation "MAE" in ML algorithms for grading students' assignments refers to MAE, which have widely used metric for assessing accuracy of regression models, including those used in grading systems. MAE calculates the average absolute deviation between the predicted grades from the machine learning model and the actual grades given by human graders. This metric quantifies the level of agreement between the model's predictions and the true grades. The mathematical representation for computing MAE is shown in equation (10).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \qquad (10)$$

From equation (10), the terms n implies the samples count, y_i implies the i[th] sample actual grade and $\widehat{y_i}$ implies the i[th] sample predicted grade. A smaller MAE value signifies superior performance, as it suggest that model forecast for more accuracy in relation to the true grades. When compared to other algorithms, the newly proposed AGS-MLS achieved the lowest MAE value of 0.3504. The graphical represented of the MAE calculation is illustrated in figure 6.

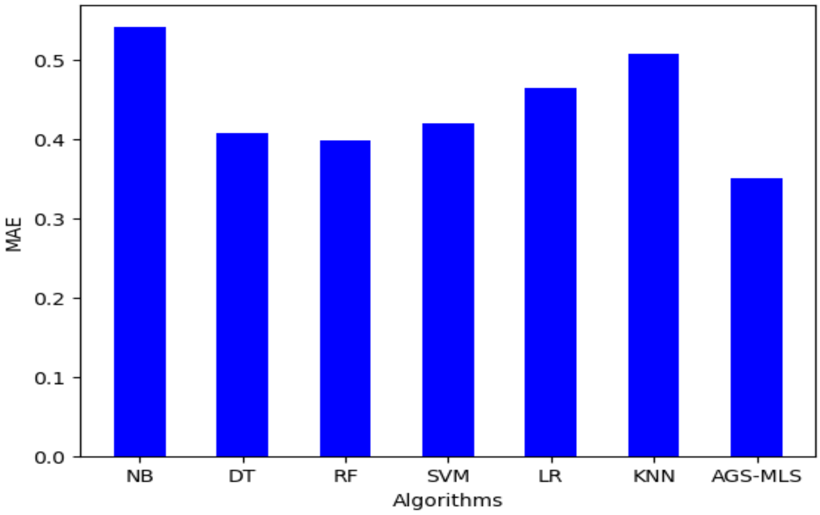

Figure 6 - MAE Calculation

5.2 MSE Calculation: When students utilize ML algorithms for their assignments, the acronym "MSE" represents Mean Squared Error. This is a frequently used measure for assessing the effectiveness of regression models. MSE calculates the average squared disparity between the predicted grades produced by the machine learning model and the grades given by human graders. It offers a numerical evaluation of how accurately the model's predictions align with the true grades. The mathematical formula for computing MSE is shown as equation (11).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{y_i}|^2 \qquad (11)$$

From equation (11), the terms n implies the samples count, y_i implies the i[th] sample actual grade and $\widehat{y_i}$ implies the i[th] sample predicted grade. Similar to MAE, a decreased MSE value represents improved performance, as it suggests the model's forecasts are more accurate. MSE gives more weight to significant errors due to the squaring process, causing it to be more affected by extreme values. In comparison to alternative methods, the newly proposed AGS-MLS achieved the most minimal MSE value of 0.2124. The graphical represented of the MSE calculation is illustrated in figure 7.

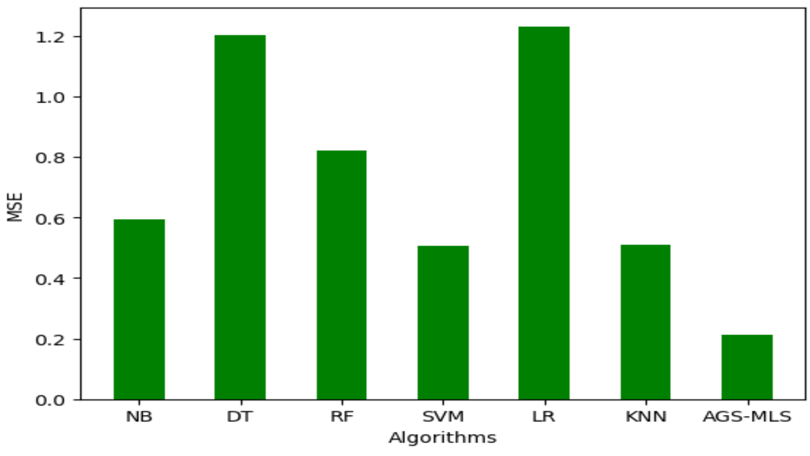

Figure 7 - MSE Calculation

5.3 RMSE Calculation: It is an abbreviation used in machine learning algorithms that grade students' assignments. It represents Root Mean Squared Error and is a widely used measurement for evaluating the effectiveness of regression models. This metric measures average size of discrepancies between grades predicted by the model and those assigned by human graders. It gives a numerical evaluation of how accurately the model's predictions align with the actual grades, similar to MSE. The calculation for RMSE involves finding square root of the mean of squared differences between predicted, actual grades, as shown in equation (12).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y_i}|^2} \tag{12}$$

In the equation (12), the terms n implies the samples count, y_i implies the i^{th} sample actual grade and $\hat{y_i}$ implies the i^{th} sample predicted grade. RMSE is a useful way to determine the average size of errors using the same units as the grading system, such as percentage points. This allows for easy understanding in the context of grading assignments. Lower RMSE values, like with MSE indicate better performance by model as it means its predictions are more accurate. AGS-MLS method had the lowest RMSE value of 0.5158 compared to other methods, as shown in figure 8 where MSE is graphically represented.

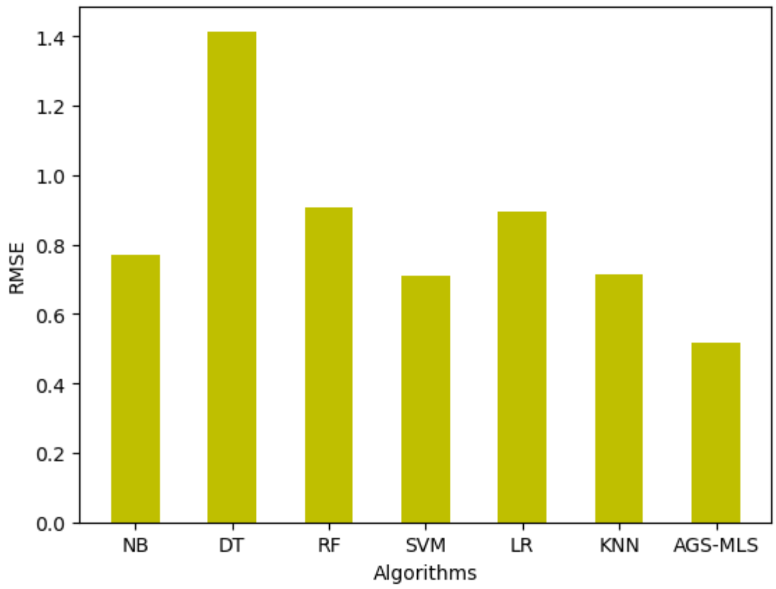

Figure 8 - RMSE Calculation

5.4 R^2 Calculation: In machine learning algorithms designed for grading students' assignments, R^2 is statistical measurement utilized to assess effectiveness of regression models. This metric evaluates the amount of variation in the desired outcome (such as grades) that can be attributed to the regression model. It serves as a gauge for how closely the model aligns with the data, compared to a basic baseline model (typically a straight line representing the average of the desired outcome). The values for R^2 range from 0 to 1, with 1 indicating a perfect fit and 0 meaning that model does not explain any of the variation in target variable. Mathematical formula for computing R^2 is shown in equation (13).

$$R^2 = 1 - \frac{SSR}{SST} \tag{13}$$

From the equation (13), the sum of squared residuals, or SSR, represents the errors in the model, while SST is total sum of squares, measures overall variability in target variable. A higher R^2 value signifies that the model can account for a greater portion of the variance in predicting students' assignment grades. A closeto1 R^2value indicates a strong fit to the data, whereas values near 0 suggest inadequate performance. The AGS-MLS method had the RMSE value of 0.93 compared to other methods, as shown in figure 9 where R^2 is graphically represented.

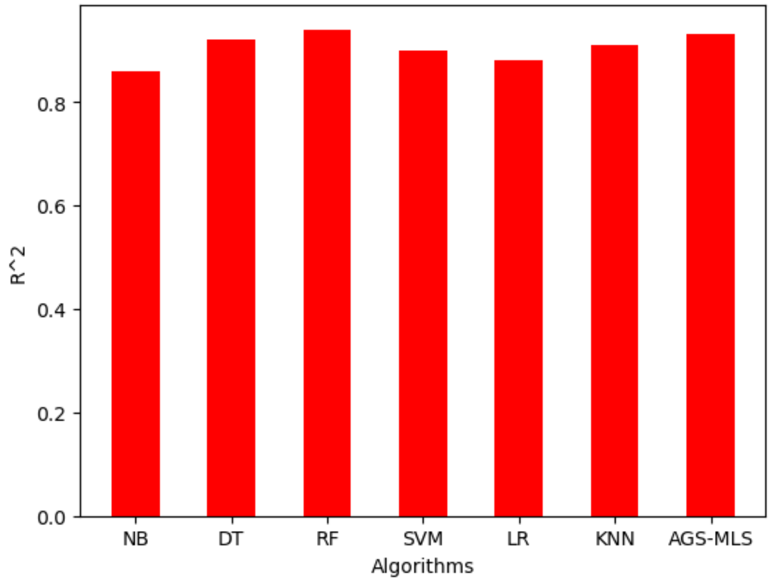

Figure 9 - R^2 Calculation

The analysis of considered algorithms with respect to the parameters like accuracy, Precision, Recall and F1-score are given in table 5.

Table 5 – Performance Concerned with accuracy, Precision, Recall and F1-score

ML Algorithms	Accuracy	Precision	Recall	F1-Score
Naive Bayes	0.91	0.93	0.94	0.93
Decision Trees	0.97	0.96	0.93	0.92
Random Forest	0.96	0.95	0.92	0.93
Support Vector Machines	0.98	0.97	0.95	0.96
Linear Regression	0.88	0.89	0.84	0.85
K-Nearest Neighbors	0.9	0.92	0.89	0.9
Proposed AGS-MLS	0.99	0.98	0.98	0.97

5.1.5 Accuracy Calculation: In ML methods used for evaluating students' assignments, accuracy serves as a measure of performance that assesses how accurately the model predicts grades in comparison to those given by human evaluators. The typical calculation for a ML model's accuracy involves the following equation (14):

$$Accuracy = {}^{No\ of\ correct\ predictions}\!/_{Total\ No\ of\ Predictions} \quad (14)$$

The accuracy of predictions is determined by the number of instances where the predicted grade matches the actual grade given by humans. A total number of assignments in data set also plays a role in determining this accuracy. In terms of accuracy, compared with the other algorithms our proposed AGS-MLS performs better and it attains accuracy up to 99%. The calculation of accuracy of all the presented algorithms in the proposed AGS-MLS system is illustrated in figure 10.

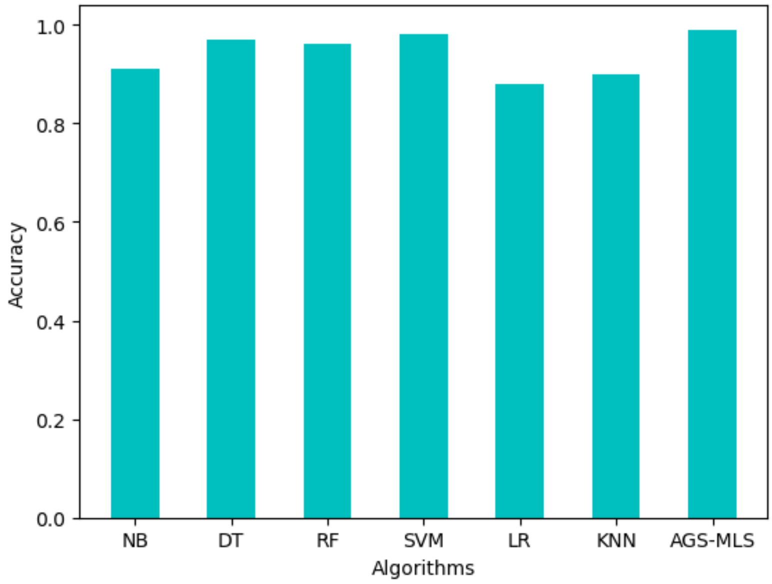

Figure 10 - Accuracy Calculation

5.1.6 Precision Calculation: In ML, precision is a measure of how well a model accurately predicts positive outcomes. When grading students' assignments automatically, precision can be used to gauge the accuracy of the model's predicted grade for each assignment. It is determined by using the formula (15):

$$Precision = \frac{True\ Positives\ (TP)}{True\ Positives + False\ Positives\ (FP)} \quad (15)$$

From the above equation the term TP refers to cases where model accurately predicts a positive outcome, such as correctly graded assignments. On other hand, FP refers to instances where model mistakenly predicts a positive outcome, such as assignments that were incorrectly graded as positive. In the proposed AGS-MLS system compared with other algorithms our proposed AGS-MLS system obtained maximum precision up to 98%. The calculation of precision of all the presented algorithms in the proposed AGS-MLS system is illustrated in figure 11.

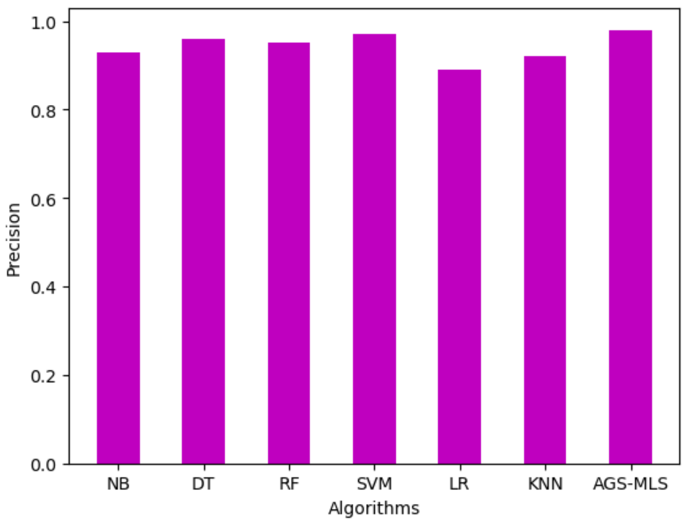

Figure 11 - Precision Calculation

5.1.7 Recall Calculation: Recall, also referred to as sensitivity or true positive rate a measure used in ML model to evaluate how accurately a model can identify all relevant cases of a given class. In the scenario of automatically grading students' assignments, recall measures the model's aptitude in recognizing all assignments that genuinely fall under a particular grade level. The formula for calculating recall is as follows:

$$Recall = \frac{True\ Positives\ (TP)}{True\ Positives + False\ Negatives\ (FN)} \quad (16)$$

In the above formula, TP refer to cases where model accurately predicts a positive outcome, such as correctly graded assignments. FN, on other hand, refers to cases where model inaccurately predicts a negative outcome, such as assignments that were actually positive but predicted as negative. In the proposed AGS-MLS system, compared with other algorithms the proposed AGS-MLS achieved better recall value up to 98%. The calculation of recall of all the presented algorithms in the proposed AGS-MLS system is illustrated in figure 12.

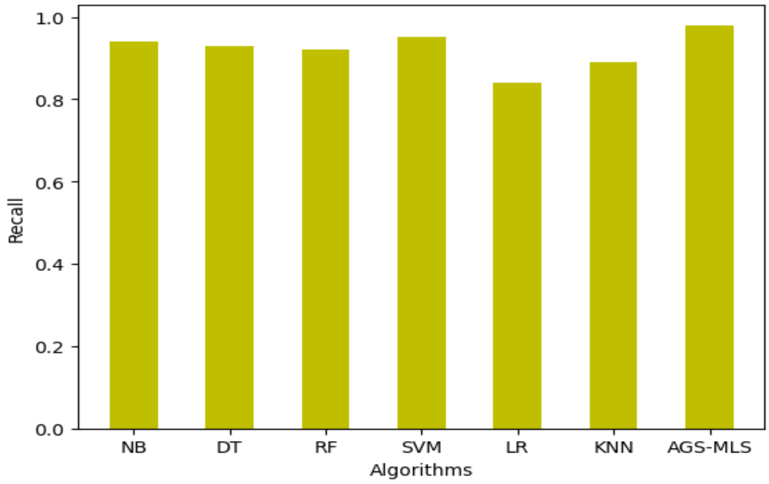

Figure 12 - Recall Calculation

5.1.8 F1-Score Calculation: The F1-score is a measurement utilized in ML which combines precision and recall to create a unified value, offering an equitable assessment of a model's effectiveness. When evaluating students' assignments through automated grading, the F1-score is beneficial in cases where achieving a balance between precision, recall is crucial. A calculation of the F1-score follows this formula:

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (17)$$

In the given formula, Precision quantifies the ratio of accurate positive predictions to all positive predictions made, whereas recall gauges the ratio of accurate positive predictions to all actual positive instances. F1-score, which can range from 0 to 1, is a useful metric for evaluating performance, especially in cases where there is an unequal distribution of classes or when false positives and false negatives have varying effects on the overall task. Comparing with other algorithms the proposed AGS-MLS achieved better F1-score value up to 97%. The calculation of F1-score of all the presented algorithms in the proposed AGS-MLS system is illustrated in figure 13.

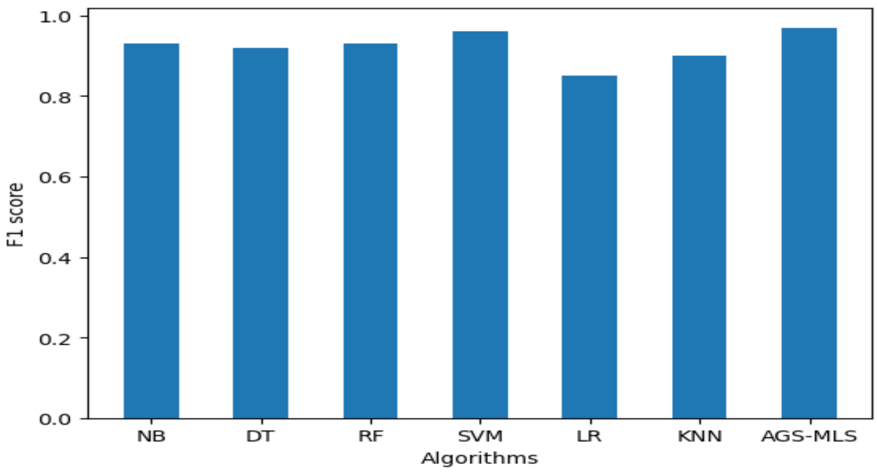

Figure - F1-Score Calculation

VI. CONCLUSION:

The focus of the article is to develop efficient ML algorithms that can automatically grade students' assignments, which is a significant advancement in the education field in terms of accuracy and scalability. The fundamentals of these ML algorithms are explained first, followed by their role in automating the grading process. This not only reduces the workload on educators, but also allows them to concentrate on more strategic and personalized aspects of teaching. The improved accuracy is particularly beneficial when dealing with large numbers of assignments, ensuring prompt feedback to students without compromising the quality of assessment. This leads to a fair and consistent evaluation system, promoting the overall integrity of academic evaluations. Additionally, these algorithms open up opportunities for personalized learning experiences. By analyzing patterns in student performance, the system can provide targeted feedback, identify areas for improvement, and adjust educational methods according to individual learning styles, thereby enhancing the educational journey for each student.

REFERENCES:

[1] Ahmet Emin Tatar, Dilek Düstegör, et.al, "Prediction of Academic Performance at Undergraduate Graduation: Course Grades or Grade Point Average?", Application science, vol. 10, pp. 4967, 2020, doi: 10.3390/app10144967

[2] Cristian D. González-Carrillo, Felipe Restrepo-Calle, et.al, "Automatic Grading Tool for Jupyter Notebooks in Artificial Intelligence Courses", Sustainability, vol. 13, pp. 12050, 2021, doi: 10.3390/su132112050

[3] Ilie Gligorea, Muhammad Usman Yaseen, et.al, "An Interpretable Framework for an Efficient Analysis of Students' Academic Performance", Sustainability, vol. 14, pp. 8885, 2022, doi: 10.3390/su14148885

[4] Daina Gudoniene, Evelina Staneviciene, et.al, "The Scenarios of Artificial Intelligence and Wireframes Implementation in Engineering Education", Sustainability, vol. 15, pp. 6850, 2023, doi: 10.3390/su15086850

[5] Daniel Doz, Mara Cotic, et.al, "Random Forest Regression in Predicting Students' Achievements and Fuzzy Grades", Mathematics, vol. 11, pp. 4129, 2023, doi: 10.3390/math11194129

[6] Silvia Gaftandzhieva, Ashis Talukder, et.al, "Exploring Online Activities to Predict the Final Grade of Student", Mathematics, vol. 10, pp. 3758, 2022, doi: 10.3390/math10203758

[7] Sujan Poudyal, Mahnas J. Mohammadi-Aragh, et.al, "Hybrid Feature Extraction Model to Categorize Student Attention Pattern and Its Relationship with Learning", Electronics, vol. 11, pp. 1476, 2022, doi: 10.3390/electronics11091476

[8] Rezvan Nazempour, Houshang Darabi, "Personalized Learning in Virtual Learning Environments Using Students' Behavior Analysis", Educational science, vol. 13, pp. 457, 2023, doi: 10.3390/educsci13050457

[9] Alper Bayazit, Nihal Apaydin, et.al, "Predicting At-Risk Students in an Online Flipped Anatomy Course Using Learning Analytics", Educational science, vol. 12, pp. 581 2022, doi: 10.3390/educsci12090581

[10] Tian Qiu, Sizhe Wang, et.al, "Predicting Risk of Bullying Victimization among Primary and Secondary School Students: Based on a Machine Learning Model", Behaviour science, vol. 14, pp. 73, 2024, doi: 10.3390/bs14010073

[11] Barnabás Holicza, Attila Kiss, "Predicting and Comparing Students' Online and Offline Academic Performance Using Machine Learning Algorithms", Behav. Science, vol. 13, pp. 289, 2023, doi: 10.3390/bs13040289

[12] Han Xue, Yanmin Niu, "Multi-Output Based Hybrid Integrated Models for Student Performance Prediction", Application science, vol. 13, pp. 5384, 2023, doi: 10.3390/app13095384

[13] Nur Izzati Mohd Talib, Nazatul Aini Abd Majid, et.al, "Identification of Student Behavioral Patterns in Higher Education Using K-Means Clustering and Support Vector Machine", Application science, vol. 13, pp. 3267, 2023, doi: 10.3390/app13053267

[14] Warunya Wunnasri, Pakarat Musikawan, et.al, "A Two-Phase Ensemble-Based Method for Predicting Learners' Grade in MOOCs", Applicaton science, vol. 13, pp. 1492, 2023, doi: 10.3390/app13031492

[15] Yara Zayed, Yasmeen Salman, et.al, "A Recommendation System for Selecting the Appropriate Undergraduate Program at Higher Education Institutions Using Graduate Student Data", Application science, vol. 12, pp. 12525, 2022, doi: 10.3390/app122412525

[16] Bin Xu, Sheng Yan, et.al, "A Federated Transfer Learning Framework Based on Heterogeneous Domain Adaptation for Students' Grades Classification", Application science, vol. 12, pp. 10711, 2022, doi: 10.3390/app122110711

[17] Stamatis Karlos, Georgios Kostopoulos, et.al, "Predicting and Interpreting Students' Grades in Distance Higher Education through a Semi-Regression Method", Application science, vol. 10, pp. 8413, 2020, doi: 10.3390/app10238413

[18] Kingsley Okoye, Julius T. Nganji, et.al, "Machine learning model (RG-DMML) and ensemble algorithm for prediction of students' retention and graduation in education", Computers and Education: Artificial Intelligence, vol. 6, pp. 100205, 2024, doi: 10.1016/j.caeai.2024.100205

[19] Nicholas Robert Beckham, Limas Jaya Akeh, et.al, "Determining factors that affect student performance using various machine learning methods", Procedia Computer Science, vol. 216, pp. 597–603, 2023, doi: 10.1016/j.procs.2022.12.174

[20] Wen-Yen Lu, Szu-Chun Fan, "Developing a weather prediction project-based machine learning course in facilitating AI learning among high school students", Computers and Education: Artificial Intelligence, vol. 5, pp. 100154, 2023, doi: 10.1016/j.caeai.2023.100154

[21] Aarón López-García, Olga Blasco-Blasco, et.al, "Early detection of students' failure using Machine Learning techniques", Operations Research Perspectives, vol. 11, pp. 100292, 2023, doi: 10.1016/j.orp.2023.100292

[22] Suchithra Rajendran, S Chamundeswari, et.al, "Predicting the academic performance of middle- and high-school students using machine learning algorithms", Social Sciences & Humanities Open, vol. 6, pp. 100357, 2022, doi: 10.1016/j.ssaho.2022.100357

[23] Zhu, Y, Wang, et.al, "Naïve Bayes classifier based on reliability measurement for datasets with noisy labels", Annals of Operations Research, vol. 1, no. 28, 2023.

[24] Karmagatri, M., Kurnianingrum, et.al, "Predicting Factors Related to Student Performance Using Decision Tree Algorithm", International Conference on Cybernetics and Intelligent System (ICORIS), pp. 1-6, 2023

[25] Doz, D, Cotic, M, and Felda, D, "Random Forest Regression in Predicting Students' Achievements and Fuzzy Grades", Mathematics, vol. 11, no. 19, pp. 4129, 2023

[26] Hameed, N. H and Sadiq, A. T, "Automatic Short Answer Grading System Based on Semantic Networks and Support Vector Machine" Iraqi Journal of Science, pp. 6025-6040, 2023

[27] Asrial, A., Syahrial, S., et.al, "Implementation of web-based character assessment on students' character outcomes: A review on perception and gender", JOTSE: Journal of Technology and Science Education, vol. 13, no. 1, pp. 301-328, 2023

[28] Munazhif, N. F., Yanris, G. J., and Hasibuan, M. N. S, "Implementation of the K-Nearest Neighbor (kNN) Method to Determine Outstanding Student Classes", Sinkron: jurnal dan penelitian teknik informatika, vol. 8, no. 2, pp. 719-732, 2023.