**Student Performance Prediction**

**A MINI PROJECT REPORT**

**18CSC305J - Artificial Intelligence**

***Submitted by***

**V Venkat Aditya [RA2111003011799]**

*Under the Guidance of*

**Dr. Lubin Balasubramanian**

Assistant Professor, Department of Computing Technologies

***in partial fulfillment for the award for the degree***

***of***

**BACHELOR OF TECHNOLOGY**

in

**COMPUTER SCIENCE ENGINEERING**

of

**FACULTY OF ENGINEERING AND TECHNOLOGY**



S.R.M. Nagar, Kattankulathur, Chengalpattu District

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

**BONAFIDE CERTIFICATE**

Certified that Mini project report titled **“Student Performance Prediction”** is the bonafide work of **Vellanki Venkat Aditya [RA2111003011799]** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**SIGNATURE**

Dr. Lubin Balasubramanian Associate Professor

Department of Computing Technologies

**SIGNATURE**

Dr. M. Pushpalatha Head of the department

Department of Computing Technologies

# ABSTRACT

The objective of studying student performance prediction is to develop accurate models that can predict students' academic achievement. Predictive analytics has garnered significant attention as a method to improve student outcomes, given the increasing significance of data-driven decision-making in the field of education. This abstract provides a comprehensive summary of the current state of research on predicting student performance. It specifically highlights key tactics, techniques, and challenges in this field. Various methodologies, including data mining techniques, machine learning algorithms, and statistical methods, have been employed to predict student achievement. These techniques construct forecast models by using a range of variables, such as social characteristics, attendance, past academic performance, and demographic information. Machine learning algorithms have shown potential in accurately predicting student performance by leveraging large amounts of data and identifying complex patterns. Moreover, the predictive models developed for student performance prediction have multiple applications, including the identification of students who may be at-risk and in need of additional support, the improvement of educational policies, and the optimisation of curriculum design. Moreover, these models can assist educators in their first efforts to prevent academic underachievement and student attrition. In addition, higher education institutions have utilised student performance prediction for tailored learning, course recommendations, and admissions determinations. While there has been progress in forecasting student performance, there are still unresolved concerns. Ensuring the ethical utilisation of student data, encompassing apprehensions over privacy, bias, and impartiality, is a significant challenge. Furthermore, the ability to gain the support and approval of stakeholders such as parents, teachers, and children relies on the degree to which prediction models can be easily understood and clarified.

**TABLE OF CONTENTS**

**Chapter No. Title Page No.**

|  |  |
| --- | --- |
| **ABSTRACT**  **TABLE OF CONTENTS**  1 **INTRODUCTION** | **iii iv 1** |
| 1.1 Importance of student performance prediction | 1 |
| 1.2 Techniques for Student Performance Prediction | 2 |
| 2 **LITERATURE SURVEY** | **3** |
| 2.1 Predictive Modeling Techniques | 3 |
| 2.1.1 Decision Trees | 3 |
| 2.1.2 Logistic Regression | 3 |
| 2.1.3 Support Vector Machines (SVM) | 3 |
| 2.1.4 Gradient Boosting | 3 |
| 2.1.5 Neural Networks | 4 |
| 2.1.6 Time Series Analysis | 4 |
| 2.2 Feature Selection and Feature Engineering | 4 |
| 2.2.1 Univariate Feature Selection | 4 |
| 2.2.2 Recursive Feature Elimination (RFE) | 4 |
| 2.2.3 Feature Importance from Tree-based Models | 4 |
| 2.2.4 Creating Aggregate Features | 5 |
| 2.2.5 Creating Interaction Features | 5 |
| 2.2.6 Normalizing or Scaling Features | 5 |
| 2.2.7 Handling Missing Data | 5 |
| 2.3 Evaluation metrics | 5 |
| 2.3.1 Classification Metrics | 5 |
| 2.3.2 Regression Metrics | 6 |
| 2.3.3 Cross-validation | 6 |
| 2.3.4 Cohort Analysis | 6 |
| 2.3.5 Predictive Power | 6 |

|  |  |  |
| --- | --- | --- |
| 2.4 | factors effecting student performance | 6 |
|  | 2.4.1 Demographic Factors | 6 |
|  | 2.4.2 Academic Factors | 7 |
|  | 2.4.3 Psychological Factors | 7 |
|  | 2.4.4 Environmental Factors | 7 |
|  | 2.4.5 Personal Factors | 7 |
|  | 2.4.6 Technology Factors | 7 |
| 2.5 | Applications and impacts of student performance prediction | 7 |
|  | 2.5.1 Early Warning Systems | 8 |
|  | 2.5.2 Personalized Learning | 8 |
|  | 2.5.3 Resource Allocation | 8 |
|  | 2.5.4 Curriculum Design | 8 |
|  | 2.5.5 Policy Making | 8 |
|  | 2.5.6 Parental Engagement | 9 |
| 3 | **SYSTEM ARCHITECTURE AND DESIGN** | **10** |
| 3.1 | Data Collection | 10 |
|  | 3.1.1 Identify Relevant Data | 10 |
|  | 3.1.2 Data Cleaning and Pre-processing | 10 |
|  | 3.1.3 Data Integration | 10 |
|  | 3.1.4 Data Storage | 10 |
|  | 3.1.5 Data Documentation | 10 |
| 3.2 | Data Pre-processing | 11 |
|  | 3.2.1 Handling Missing Values | 11 |
|  | 3.2.2 Handling Outliers | 11 |
|  | 3.2.3 Data Normalization | 11 |
| 3.3 | Machine Learning Models | 11 |
|  | 3.3.1 Linear Regression | 11 |
|  | 3.3.2 Logistic Regression | 12 |
|  | 3.3.3 Decision Trees | 12 |
|  | 3.3.4 Artificial Neural Networks (ANN) | 12 |

* 1. [Model Evaluation 12](#_bookmark0)
     1. Mean Squared Error (MSE) 12
     2. Root Mean Squared Error (RMSE) 12
     3. R-squared (R2) 12
     4. F1 Score 12
     5. Receiver Operating Characteristic (ROC) Curve 13
  2. [Model Deployment 13](#_bookmark1)
     1. Web Application 13
     2. REST API 13
     3. Mobile Application 13
  3. [User Interface 13](#_bookmark2)
     1. Input Data 13
     2. Model Output 13
     3. User Feedback 14
     4. Accessibility 14
     5. User Experience (UX) 14
  4. [Monitoring and Maintenance 14](#_bookmark3)
     1. Data Quality 14
     2. Model Performance 14
     3. Security 14
     4. Scalability 14
     5. Documentation 15
  5. [Data Security and Privacy 15](#_bookmark4)

|  |  |  |
| --- | --- | --- |
| 3.8.1 | Data Encryption | 15 |
| 3.8.2 | Access Control | 15 |
| 3.8.3 | Secure Storage | 15 |
| 3.8.4 | Regular Audits | 15 |
| 3.8.5 | Privacy Policies | 15 |
| 3.8.6 | Data Breach Response Plan | **15** |
| **4** | **METHODOLOGY** | **16** |

|  |  |  |
| --- | --- | --- |
| 4.1 | Define the problem | 16 |
| 4.2 | Data collection and preprocessing | 17 |
| 4.3 | Feature engineering | 17 |
| 4.4 | Exploratory data analysis (EDA) | 17 |
| 4.5 | Machine learning model selection | 17 |
| 4.6 | Model training and testing | 17 |
| 4.7 | Model tuning and validation | 17 |
| 4.8 | Model deployment | 17 |
| 4.9 | Model monitoring and maintenance | 17 |
| **5** | **CODING AND TESTING** | **19** |
| **6** | **RESULTS AND DISCUSSIONS** | **27** |
| **7** | **CONCLUSION AND FUTURE ENHANCEMENT** | **32** |
|  | **REFERENCES** |  |
| **34** |  |  |

CHAPTER 1 INTRODUCTION

# Importance of student performance prediction

Identifying students who are at risk of academic underachievement relies significantly on the capacity to forecast student performance. Educators can effectively intervene and provide extra assistance to difficult kids by accurately predicting student achievement. This may encompass customised tutoring, targeted interventions, or modifications to instructional approaches, all of which can facilitate children in attaining improved outcomes and excelling academically. Anticipating a student's academic performance can assist teachers in recognising individuals who have the potential to excel. This enables them to provide challenging and engaging educational opportunities that will facilitate the realisation of their maximum capabilities. Moreover, the ability to forecast student performance can aid administrators in effectively distributing resources and developing interventions based on empirical evidence to improve overall student achievement.

# Techniques for Student Performance Prediction

There are multiple strategies available for forecasting student success. Data analysis is a commonly used technique that involves analysing historical data on student performance, such as grades, attendance, and demographics, in order to identify patterns and trends. Regression analysis, decision trees, and machine learning algorithms are statistical techniques that can be employed to create prediction models using existing data, aiming to anticipate future performance. Extensive and intricate examination of datasets can be particularly valuable for machine learning techniques such as neural networks, random forests, and support vector machines, enabling the generation of accurate predictions.

CHAPTER II

LITERATURE SURVEY

Recently, there has been a significant focus on forecasting student performance in educational settings due to its ability to enhance educational outcomes and guide educational interventions. Researchers and educators have been utilising various strategies to predict student performance, identify students who may be struggling academically, and implement interventions to improve their learning. This is made possible by the increasing accessibility of educational data and the advancement of data analytics tools. This study of the literature will analyse significant research on forecasting student achievement in educational environments.

# Predictive Modelling Techniques:

Various predictive modelling techniques, such as machine learning algorithms, statistical methodologies, and data mining approaches, have been employed to predict student performance.

* + 1. Decision Trees:

Decision trees are hierarchical structures that visually represent choices or decisions based on certain criteria. Random Forest and C4.5 are decision tree-based algorithms that have been utilised to predict student achievement. These algorithms have the capability to handle complex feature interactions including both continuous and categorical data.

* + 1. Logistic Regression:

Logistic regression, a statistical tool, is used to predict the probability of an event happening. Logistic regression is a valuable tool for predicting student performance as it can estimate the likelihood of a student passing or failing a course.

* + 1. Support Vector Machines (SVM):

Support Vector Machines (SVM) is a popular machine learning method used for categorization tasks. Support Vector Machines (SVM) can be employed to forecast student performance by training a model with labelled data that include variables like attendance, engagement, and historical academic achievement. The objective of Support Vector Machines (SVM) is to determine the hyperplane that optimally separates the data points into separate categories, such as passing or failing a course.

* + 1. Gradient Boosting:

Gradient boosting is an ensemble method that enhances the accuracy of a prediction model by amalgamating multiple weak models. By iteratively including underperforming models that are trained on the errors made by the previous models, this approach can be employed to predict student performance. XGBoost and AdaBoost are two instances of gradient boosting algorithms employed for predicting student performance using various characteristics.

* + 1. Neural Networks:

Student performance prediction has also utilised neural networks and other deep learning approaches. Neural networks have the ability to capture intricate data patterns and may be trained on large datasets to generate accurate predictions. By analysing factors such as participation, attendance, and academic performance, one can predict the potential success of students.

* + 1. Time Series Analysis:

Time series analysis is a technique used to analyse data that changes over time. Time series analysis is a method that can be employed to study patterns and trends in student performance data over a period of time. This analysis can be used to forecast student achievement by examining academic performance trends throughout different semesters or academic years.

# Feature Selection and Feature Engineering:

Feature engineering and feature selection are crucial steps for developing accurate student performance prediction models. Researchers have explored many approaches to determine the most relevant factors that impact students' academic success.

* + 1. Univariate Feature Selection:

This approach entails choosing features based on their individual statistical importance or performance when considered alone. Standard methods involve employing statistical tests such as the chi-square test, t-test, or ANOVA to identify factors that have a substantial influence on student performance.

* + 1. Recursive Feature Elimination (RFE):

Recursive Feature Elimination (RFE) is an iterative method that entails training a model numerous times and progressively removing less significant features based on their importance or impact on the model's performance. This method aids in the identification of the most significant elements by assessing their relevance to the prediction problem.

* + 1. Feature Importance from Tree-based Models: Predictive models, like decision trees and random forests, offer feature significance scores that indicate the contribution of each feature to the model's performance. The scores can be utilised to prioritise and choose the most significant attributes for predicting student achievement.

Feature Engineering:

Feature engineering encompasses the process of generating novel features or modifying existing features to more accurately capture the fundamental patterns and associations within the data. This can result in enhanced predictive efficacy and more precise forecasts of student performance. Several prevalent methods for feature engineering in the context of predicting student performance.

* + 1. Creating Aggregate Features:

This entails consolidating data at a more comprehensive level, such as computing the mean or total of attributes over a specific timeframe, in order to catch trends or patterns that may not be evident at the individual level. By consolidating weekly attendance data into monthly attendance percentages, we can obtain a more significant characteristic for forecasting student performance.

* + 1. Creating Interaction Features:

Interaction features are formed by amalgamating two or more preexisting characteristics in order to capture the potential interactions or synergies that may arise between them. For instance, by amalgamating variables like study hours and previous academic achievements, we can generate an interaction feature that encapsulates the relationship between study habits and prior performance.

* + 1. Normalising or Scaling Features:

Scaling or normalising features can be beneficial for aligning features to a consistent scale, which is crucial for models that are responsive to the size of the features, such as logistic regression or support vector machines. Common methods for data normalisation include min-max scaling, z-score normalisation, and log transformation.

* + 1. Handling Missing Data:

The absence of data can have a negative effect on the precision of predictive models. To properly handle missing data and ensure a complete feature set for prediction, many techniques can be applied, such as imputation, deletion, or advanced methods like multiple imputation or k-nearest neighbour imputation.

# Evaluation metrics

Evaluation metrics are essential for evaluating the performance and precision of predictive models in student performance prediction programmes. Below are several frequently employed evaluation metrics.

Accuracy: Calculates the accuracy by dividing the number of accurately predicted instances by the total number of instances. It is frequently employed for datasets that have an equal distribution of data points across different classes or categories.

Precision: The true positive rate is a measure of the model's accuracy in accurately predicting positive cases, calculated by dividing the number of true positive predictions by the total number of projected positive instances.

Recall: The true positive rate is calculated by dividing the number of correctly predicted positive cases by the total number of real positive instances. This metric reflects the model's capability to accurately identify all positive instances.

F1-Score:The harmonic mean of precision and memory is a metric that provides a balanced measure of both precision and recall.

Area Under the Receiver Operating Characteristic (ROC) Curve and Precision-Recall (PR) Curve: The ROC curve illustrates the relationship between the true positive rate and the false positive rate, whereas the PR curve depicts the precision vs the recall. When the AUC-ROC and AUC-PR values are near to 1, it indicates that the prediction performance is excellent.

* + 1. Regression Metrics:

Mean Squared Error (MSE) and Root Mean Squared Error (RMSE): MSE calculates the mean of the squared discrepancies between anticipated and actual values, whereas RMSE is the square root of MSE, offering a measure of the average absolute prediction error.

R-squared (R2) Score: The measure quantifies the percentage of the overall variability in the target variable that can be accounted for by the predictive model. The R2 score is a metric that measures the goodness of fit of a model to the data. It runs from 0 to 1, where higher values indicate a stronger fit of the model to the data.

Cross-validation: Cross-validation techniques, such as k-fold cross-validation or stratified k-fold cross-validation, enhance the reliability and robustness of assessment results by training and assessing the model on distinct subsets of data.

* + 1. Cohort Analysis:

Analysing the performance of various student cohorts or groups, such as those based on gender or ethnicity, can offer valuable insights into how the model performs for distinct subpopulations.

* + 1. Predictive Power:

Evaluating the model's predictive capability by assessing its accuracy in forecasting future student performance or outcomes

# factors affecting student performance

Various variables can influence the academic performance of students within the framework of a student performance prediction project. These factors can be classified into various groups.

* + 1. Demographic Factors:

Student performance can be influenced by demographic factors such as gender, age, ethnicity, social level, and cultural background. Research indicates that there may be gender-based disparities in academic performance, since males and females often exhibit distinct patterns of achievement across various disciplines or educational levels. Likewise, the social situation and cultural background of students might affect their ability to get educational resources and opportunities, which can subsequently affect their academic achievement.

* + 1. Academic Factors:

Academic variables such as previous academic accomplishments, regular attendance, active participation in classroom activities, effective study habits, and strong motivation can have a substantial influence on student success. Students who possess a solid intellectual base, maintain regular attendance, actively participate in classroom activities, employ effective study techniques, and demonstrate high levels of drive are more inclined to achieve academic success.

* + 1. Psychological Factors:

Student performance can be influenced by various psychological aspects such as cognitive capacities, intellect, learning style, self-efficacy, self-regulation, and emotional well-being. Students who possess superior cognitive talents, adept self- regulation skills, and a favourable emotional state are more inclined to excel academically.

* + 1. Environmental Factors:

Student performance can be influenced by environmental factors, including the school's quality, classroom atmosphere, access to educational resources, and support from instructors, peers, and parents.

# APPLICATION AND IMPACTS OF STUDENT PERFORMANCE PREDICTION

The applications and effects of student performance prediction within the framework of a student performance prediction project might have a broad scope and significant consequences. Below are several possible applications and consequences of utilising predictive algorithms for forecasting student performance.

* + 1. Early Warning Systems:

Predictive algorithms can be employed to detect pupils who might be susceptible to academic underachievement or leaving school prematurely. Predictive models can identify students who may need extra support or interventions by assessing indicators such as past academic achievement, attendance, engagement, and behaviour. Early warning systems utilising predictive models can assist educators and administrators in promptly intervening and implementing focused interventions to enhance student outcomes.

* + 1. Personalized Learning:

Predictive models can assist in customising education to suit the specific needs of each learner. Predictive models can suggest customised learning paths, information, and resources by examining students' learning styles, interests, and performance trends. This can facilitate students in acquiring knowledge at their individual speed, addressing any deficiencies in their learning, and maximising their academic achievements.

* + 1. Resource Allocation:

Predictive models can assist in improving the allocation of educational resources. Through the analysis of student performance data, predictive models have the ability to pinpoint specific areas that may require extra resources, such as teaching staff, tutoring programmes, or learning materials. This can assist educational institutions in effectively distributing their resources and interventions to areas with the most need, hence optimising the use of available resources.

* + 1. Curriculum Design:

Predictive models have the ability to provide valuable insights that can be used to shape curriculum design and teaching practices. Through the analysis of student performance data, predictive models can discern the specific parts of the curriculum that may provide difficulties for students, the areas in which students succeed, and the areas that need to be enhanced. This information can provide valuable guidance to curriculum designers and educators in formulating instructional strategies, content, and assessments that are more closely aligned with the specific requirements of kids, ultimately resulting in enhanced student performance.

* + 1. Policy Making:

Forecasting models have the ability to provide guidance for the development of educational policies. Through the analysis of extensive student performance data, predictive models can offer valuable insights into the efficacy of educational policies and interventions. This can assist policymakers in making informed decisions based on data and implementing policies supported by evidence to enhance overall student performance and educational outcomes. This can facilitate parental involvement, facilitate timely interventions, and empower parents to actively participate in their child's academic career.

CHAPTER III

SYSTEM ARCHITECTURE AND DESIGN

An optimal system design for a "student performance prediction" project would normally consist of several interconnected components that collaborate to gather, manipulate, and scrutinise data in order to produce accurate forecasts. Below is a concise summary of a standard system architecture.

# Data Collection:

This component entails gathering pertinent data pertaining to students' performance, including previous academic records, attendance, engagement, behaviour, and other pertinent aspects. Data can be gathered from diverse sources, including student information systems, learning management systems, attendance records, surveys, and other pertinent databases.

* + 1. Identify Relevant Data:

To commence data collecting, it is imperative to ascertain the pertinent categories of data that can be used to forecast student performance. This encompasses various aspects such as academic records, attendance, engagement, behaviour, demographic information, socioeconomic position, learning style, study habits, and other elements that can potentially influence student performance. The data must be congruent with the precise goals of the project and should be ample enough to encompass the pertinent aspects that may impact student success.

* + 1. Data Cleaning and Preprocessing:

Data that has been gathered may need to undergo cleaning and preprocessing in order to address issues such as missing values, outliers, inconsistent data, and other problems related to the quality of the data. For data to be accurate and reliable, it must undergo cleaning, transformation, and preparation before analysis. Data pre- processing encompasses many techniques such as data normalisation, feature extraction, and feature engineering, which are employed to extract valuable insights from the data.

* + 1. Data Integration:

Occasionally, it is necessary to merge data from various sources to form a comprehensive dataset for study. Data integration includes the combining of datasets, the resolution of data discrepancies, and the management of data from diverse sources that possess differing formats and structures.

* + 1. Data Storage:

It is imperative to safely and appropriately store data in order to guarantee data integrity, confidentiality, and availability. To safeguard data from unauthorised access and breaches, it is essential to adopt effective data storage measures, including data backup, encryption, and access controls.

* + 1. Data Documentation:

Thorough documentation of the gathered data, encompassing data sources, data quality, data transformation, and data labelling, is crucial for ensuring transparency, repeatability, and future reference. The documentation should encompass metadata, data dictionaries, and any other pertinent information that can facilitate comprehension and use of the data in subsequent instances.

# Data Pre-processing:

Data preparation is a key step in the construction of a "student performance prediction" system. It entails cleaning, converting, and preparing the collected data to ensure its accuracy, dependability, and suitability for analysis. Here are some common data pre-processing techniques that can be applied in a student performance prediction.

* + 1. Handling Missing Values:

Missing values are a prevalent problem in real-world datasets, and they can have a negative impact on the precision and dependability of prediction models. Various strategies, such as imputation, deletion, or estimating, can be used to manage missing values. Imputation is the process of replacing missing information with estimated values derived from statistical approaches or machine learning algorithms. Deletion entails the elimination of rows or columns that have missing values, but it must be executed with caution to prevent the loss of crucial information. Estimation entails use methodologies such as regression or machine learning to forecast absent values by leveraging other accessible data.

* + 1. Handling Outliers:

Outliers refer to data points that exhibit substantial deviation from the remaining data and can have a detrimental effect on the accuracy of predictive models. Outliers can be addressed using many techniques, including filtering, transformation, or imputation. Filtering entails the elimination or substitution of outlier values using predetermined thresholds or statistical techniques. Transformation entails the utilisation of mathematical or statistical operations on the data in order to mitigate the influence of outliers. Imputation is the process of substituting abnormal numbers with estimated values derived from statistical techniques or machine learning algorithms.

* + 1. Data Normalisation:

Data normalisation is the procedure of standardising the data to a uniform range in order to prevent favouritism towards specific characteristics caused by variations in their magnitudes. Standard normalisation approaches commonly used are min-max scaling, z-score normalisation, and log transformation.

# Machine Learning Models:

Various machine learning models can be employed for "student performance prediction" depending on the available data, the unique problem at hand, and the intended outcome. Below are many frequently employed machine learning methods for predicting student achievement.

* + 1. Linear Regression:

Linear regression is a straightforward and easily understandable model that can be employed to predict numerical outcomes, such as student performance scores, using one or more input features. The model assumes a linear correlation between the input features and the target variable. It calculates the coefficients of the linear equation by employing least squares optimization.

* + 1. Logistic Regression:

Logistic regression is a statistical model used for binary classification, where it predicts binary outcomes, such as whether a student will pass or fail a course, based on input features. The logistic function is used to estimate the probability of class membership. This method can be expanded to accommodate multi-class classification by employing approaches like one-vs-rest or softmax.

* + 1. Decision Trees:

Decision trees are flexible models that can be employed for both classification and regression applications. The data is divided into partitions depending on the input features, and judgements are made at each node based on the feature values in order to forecast the target variable. Decision trees can be represented graphically and are easily understood, although they are susceptible to overfitting.

* + 1. Artificial Neural Networks (ANN):

Artificial Neural Network (ANN) is a very adaptable and intricate model that may be employed for various purposes, such as categorization, estimation, and forecasting of sequential data. An artificial neural network (ANN) is composed of several layers of interconnected nodes, often known as neurons. The network employs activation functions to incorporate non-linear characteristics into the model. Complex student performance prediction problems can also utilise deep learning architectures, such as Convolutional Neural Networks (CNN) for picture data or Recurrent Neural Networks (RNN) for sequential data.

# Model Evaluation:

Model evaluation is a crucial stage in the machine learning pipeline for "student performance prediction" since it allows us to examine the performance and generalisation capability of the models. Below are several frequently employed evaluation metrics.

* + 1. Mean Squared Error (MSE):

Mean Squared Error (MSE) is a statistical measure used in regression analysis to quantify the average of the squared differences between the predicted and actual values of the target variable. Smaller values of Mean Squared Error (MSE) indicate superior performance.

* + 1. Logistic Regression:

Logistic regression is a statistical model used for binary classification, where it predicts binary outcomes, such as whether a student will pass or fail a course, based on input features. The logistic function is used to estimate the probability of class membership. This method can be expanded to accommodate multi-class classification by employing approaches like one-vs-rest or softmax.

* + 1. Decision Trees:

Decision trees are flexible models that can be employed for both classification and regression applications. The data is divided into partitions depending on the input features, and judgements are made at each node based on the feature values in order to forecast the target variable. Decision trees can be represented graphically and are easily understood, although they are susceptible to overfitting.

* + 1. Artificial Neural Networks (ANN):

Artificial Neural Network (ANN) is a very adaptable and intricate model that may be employed for various purposes, such as categorization, estimation, and forecasting of sequential data. An artificial neural network (ANN) is composed of several layers of interconnected nodes, often known as neurons. The network employs activation functions to incorporate non-linear characteristics into the model. Complex student performance prediction problems can also utilise deep learning architectures, such as Convolutional Neural Networks (CNN) for picture data or Recurrent Neural Networks (RNN) for sequential data.

# Model Evaluation:

Model evaluation is a crucial stage in the machine learning pipeline for "student performance prediction" since it allows us to examine the performance and generalisation capability of the models. Below are several frequently employed evaluation metrics.

* + 1. Mean Squared Error (MSE):

Mean Squared Error (MSE) is a statistical measure used in regression analysis to quantify the average of the squared differences between the predicted and actual values of the target variable. Smaller values of Mean Squared Error (MSE) indicate superior performance.

* + 1. Root Mean Squared Error (RMSE):

RMSE is a modified version of MSE that calculates the square root of the MSE in order to provide a metric that is expressed in the same units as the target variable. Root Mean Square Error (RMSE) is a widely utilised statistic for regression tasks, such as predicting "student performance"

R2 is a statistical measure that quantifies the amount of variability in the target variable that can be accounted for by the model. The range of R2 is from 0 to 1

F1 SCORE:

The F1 Score is a mathematical average that combines precision and recall in a way that achieves a balance between the two. The F1 Score is a widely utilised metric for imbalanced datasets, where both precision and recall hold significance.

* + 1. The Receiver Operating Characteristic (ROC) Curve:

ROC curve is a visual depiction of the relationship between the true positive rate (TPR) and false positive rate (FPR) at various thresholds in binary classification.

F1 SCORE:

The F1 Score is a mathematical average that combines precision and recall in a way that achieves a balance between the two. The F1 Score is a widely utilised metric for imbalanced datasets, where both precision and recall hold significance.

* + 1. Receiver Operating Characteristic (ROC) Curve:

The ROC curve is a visual depiction of the relationship between the true positive rate (TPR) and false positive rate (FPR) at various thresholds in binary classification.

# Model Deployment:

Model deployment is a crucial stage in the machine learning process for "student performance prediction" since it enables the utilisation of the model in practical scenarios.

* + 1. Web Application:

A prevalent method for implementing a machine learning model is by utilising a web application. The model can be included into a web application by utilising web frameworks like Flask, Django, or Node.js. Users can engage with the online application by supplying input data, and the model will produce forecasts based on the given data.

* + 1. REST API:

Another method of deploying a machine learning model involves utilising a REST API. The model can be implemented on a cloud-based platform like AWS or Azure, and the REST API can be utilised to transmit queries to the model and obtain results. This strategy is frequently employed when the model requires access from several applications or services.

* + 1. Mobile Application:

Mobile applications can also incorporate machine learning models. The model can be incorporated into the mobile application through the utilisation of mobile development frameworks like React Native or Flutter. Users can engage with the mobile application by inputting data, and the model will produce predictions based on the supplied data.

# User Interface:

The user interface (UI) is a crucial element of any machine learning application, such as "student performance prediction", as it enables people to engage with the model and comprehend the predictions.

* + 1. Input Data:

The user interface should offer a lucid and instinctive method for users to submit data for the purpose of "student performance prediction". This may encompass input fields for demographic information, academic background, and other pertinent attributes.

* + 1. Model Output:

The user interface should present the predictions produced by the model in a concise and comprehensible manner. This may encompass a concise overview of the anticipated outcome, together with visual representations and graphs to facilitate consumers' comprehension of the elements that influence the forecast.

* + 1. User Feedback:

The user interface should include a mechanism for users to offer feedback regarding the predictions produced by the model. These functionalities may encompass the capacity to identify inaccurate forecasts, offer further data, or solicit further examination.

* + 1. Accessibility:

The user interface should be intentionally built to prioritise accessibility, ensuring that individuals with disabilities, such as visual or motor impairments, can effectively utilise it. Some such features that could be included are voice commands, screen readers, and high-contrast colour schemes.

* + 1. User Experience (UX):

The user interface (UI) should be meticulously crafted to prioritise the user experience, guaranteeing its usability and comprehensibility. These qualities may encompass elements like explicit labelling, user-friendly navigation, and adaptable design for various screen sizes and devices.

# Monitoring and Maintenance

Monitoring and maintenance are essential components of any machine learning system, including the "student performance prediction" system. Below are few factors to consider when monitoring and maintaining the system:

* + 1. Data Quality:

The precision and dependability of the forecasts produced by the model rely on the calibre of the input data. Hence, it is crucial to consistently oversee the quality of the input data and swiftly resolve any identified flaws.

* + 1. Model Performance:

Regular monitoring of the machine learning model's performance is necessary to ensure the ongoing generation of accurate and dependable predictions. This may involve tracking various measures, such as accuracy, precision, recall, and F1 score.

* + 1. Security:

The machine learning system should be constructed with a focus on security to prevent unauthorised access to critical data or model parameters. This may involve the implementation of access controls, encryption of data, and the utilisation of secure communication protocols.

* + 1. Scalability:

As the user base and data volume grow, it may become imperative to expand the machine learning system in order to maintain optimal performance. This could involve implementing the system on a cloud-based platform that has the capability to automatically adjust resources according to demand.

3.1.1 Documentation:

Ensuring that the documentation for the machine learning system is kept current is crucial. This documentation should include details on the data sources, model design, and deployment process. This can enhance the system's maintainability and provide seamless updates and modifications as required.

# Data Security and Privacy:

Ensuring data security and privacy are crucial factors to consider in the context of "student performance prediction."

* + 1. Data Encryption:

It is imperative to encrypt all student data in order to thwart illegal access. Encryption is the process of transforming data into a code that can only be accessed by approved individuals who possess the decryption key.

* + 1. Access Control:

Only authorised workers should have access to student data. One way to accomplish this is by establishing role-based access control, which involves assigning distinct roles and permissions to individuals depending on their job duties.

* + 1. Secure Storage:

It is imperative to store all student data in highly secure databases and servers, which are safeguarded by firewalls and additional security protocols.

* + 1. Regular Audits:

Periodic audits should be carried out to verify compliance with data security and privacy rules.

* + 1. Privacy Policies:

All stakeholders should be given a privacy policy that is clear and straightforward. The policy should unambiguously articulate the methods by which student data will be gathered, analysed, and employed.

* + 1. Documentation:

Ensuring that the documentation for the machine learning system is kept current is crucial. This documentation should include details on the data sources, model design, and deployment process. This can enhance the system's maintainability and provide seamless updates and modifications as required.

# Data Security and Privacy:

Ensuring data security and privacy are crucial factors to consider in the context of "student performance prediction."

* + 1. Data Encryption:

It is imperative to encrypt all student data in order to thwart illegal access. Encryption is the process of transforming data into a code that can only be accessed by approved individuals who possess the decryption key.

* + 1. Access Control:

Only authorised workers should have access to student data. One way to accomplish this is by establishing role-based access control, which involves assigning distinct roles and permissions to individuals depending on their job duties.

* + 1. Secure Storage:

It is imperative to store all student data in highly secure databases and servers, which are safeguarded by firewalls and additional security protocols.

* + 1. Regular Audits:

Periodic audits should be carried out to verify compliance with data security and privacy rules.

* + 1. Privacy Policies:

All stakeholders should be given a privacy policy that is clear and straightforward. The policy should unambiguously articulate the methods by which student data will be gathered, analysed, and employed.

CHAPTER IV METHODOLOGY

Predicting student performance is a challenging endeavour that necessitates a clearly defined process to guarantee precise outcomes. The initial phase of this process involves establishing the problem and articulating the project's objectives in a concise and unambiguous manner. After the problem has been clearly identified, the process of gathering and preparing data can commence. Relevant data can be gathered from diverse sources and preprocessed by performing tasks such as data cleansing, eliminating duplication, and addressing any missing values. The subsequent stage involves identifying the pertinent traits or variables that have the ability to influence student performance. Feature engineering entails the process of changing and selecting the most pertinent features for the predictive model. Subsequently, exploratory data analysis can be performed to acquire a deeper understanding of the data, detect regularities and tendencies, and represent the data visually through graphs and charts. After the data has been prepared, a suitable machine learning model can be chosen and trained using the data. Subsequently, the model's performance can be assessed and refined until its accuracy reaches a suitable level. Ultimately, the model can be implemented in a live operational setting to forecast student achievement. Continuously monitoring the model's performance and making appropriate adjustments through retraining or fine-tuning is crucial to maintain its accuracy and relevance.

In order to guarantee the precision of the student performance prediction model, it is essential to meticulously choose and preprocess the data. The data must be pertinent and current, and any inaccuracies or discrepancies should be rectified. Feature selection is crucial as it aids in identifying the most critical aspects that lead to student achievement. Exploratory data analysis can be employed to visually comprehend and gain a deeper understanding of the data, hence refining the model and enhancing its accuracy.

* + - 1. DEFINE THE PROBLEM:

Provide a concise and unambiguous description of the problem and the specific goals of the project. For instance, the goal could be to forecast student grades by considering a range of academic and non-academic variables.

# Data collection and preprocessing:

Gather pertinent data from diverse sources such as academic records, attendance records, and student surveys. Perform data preprocessing by eliminating duplicates, cleansing the data, and addressing any instances of missing values.

# Feature engineering:

Determine the pertinent characteristics or factors that have the potential to influence student achievement. These characteristics may encompass socio- economic background, prior academic achievement, and attendance records. Feature engineering encompasses the process of modifying and choosing the most pertinent features for the predictive model.

# Exploratory data analysis (EDA):

Perform exploratory data analysis (EDA) to extract valuable information from the data, detect recurring patterns and trends, and present the data visually through the use of graphs and charts.

# Machine learning model selection:

Choose a suitable machine learning model by considering the issue statement, the available data, and the project's purpose. Machine learning algorithms suitable for predicting student performance include linear regression, decision trees, and neural networks.

# Model training and testing:

Divide the data into separate sets for training and testing. Proceed to train the chosen machine learning model using the training set, and evaluate its performance using the testing set. Assess the model's performance by utilising suitable metrics, including accuracy, precision, recall, and F1 score.

# Model tuning and validation:

Optimise the model by fine-tuning its hyperparameters, then assess the model's performance using a distinct validation set. This process entails iteratively refining the model until its performance reaches a suitable level.

# Model deployment:

After the model has been trained and validated, it can be implemented in a production setting to forecast student performance.

# Model monitoring and maintenance:

Continuously assess the model's performance over time and adjust or refine the model as needed to maintain its accuracy and relevance.

Essentially, a clearly defined process is essential for constructing a precise model that predicts student success. This methodology encompasses several steps: problem definition, data collection and preprocessing, feature selection, exploratory data analysis, machine learning model selection, model training and testing, model fine-tuning and validation, model deployment, and model monitoring and maintenance. By adhering to this process, we may construct a precise model that can aid in forecasting student performance and discerning the components that contribute to academic achievement.

CHAPTER V CODING AND TESTING

import pandas as pd import seaborn as sb

import matplotlib.pyplot as plt import time as t

import sklearn.utils as u

import sklearn.preprocessing as pp import sklearn.tree as tr

import sklearn.ensemble as es import sklearn.metrics as m import sklearn.linear\_model as lm

import sklearn.neural\_network as nn import numpy as np

#import random as rnd import warnings as w w.filterwarnings('ignore') ch

= 0

while(ch != 10):

print("1.Marks Class Count Graph\t2.Marks Class Semester-wise Graph\n3.Marks Class Gender-wise Graph\t4.Marks Class Nationality- wise Graph\n5.Marks Class Grade-wise Graph\t6.Marks Class Section- wise Graph\n7.Marks Class Topic-wise Graph\t8.Marks Class Stage-wise Graph\n9.Marks Class Absent Days-wise\t10.No Graph\n")

ch = int(input("Enter Choice: ")) if (ch == 1):

print("Loading Graph \n")

t.sleep(1)

print("\tMarks Class Count Graph")

axes = sb.countplot(x='Class', data=data, order=['L', 'M', 'H']) plt.show()

elif (ch == 2):

print("Loading Graph \n") t.sleep(1)

print("\tMarks Class Semester-wise Graph") fig, axesarr = plt.subplots(1, figsize=(10, 6))

sb.countplot(x='Semester', hue='Class', data=data, hue\_order=['L', 'M', 'H'], axes=axesarr)

plt.show() elif (ch == 3):

print("Loading Graph..\n") t.sleep(1)

print("\tMarks Class Gender-wise Graph") fig, axesarr = plt.subplots(1, figsize=(10, 6))

sb.countplot(x='gender', hue='Class', data=data, order=['M', 'F'], hue\_order=['L', 'M', 'H'], axes=axesarr)

plt.show() elif (ch == 4):

print("Loading Graph..\n") t.sleep(1)

print("\tMarks Class Nationality-wise Graph") fig, axesarr = plt.subplots(1, figsize=(10, 6))

sb.countplot(x='NationalITy', hue='Class', data=data, hue\_order=['L', 'M', 'H'], axes=axesarr)

plt.show() elif (ch == 5):

print("Loading Graph: \n") t.sleep(1)

print("\tMarks Class Grade-wise Graph") fig, axesarr = plt.subplots(1, figsize=(10, 6))

sb.countplot(x='GradeID', hue='Class', data=data, order=['G-02', 'G- 04', 'G-05', 'G-06', 'G-07', 'G-08', 'G-09', 'G-10', 'G-11', 'G-12'], hue\_order

= ['L', 'M', 'H'], axes=axesarr) plt.show()

elif (ch ==6):

print("Loading Graph..\n") t.sleep(1)

print("\tMarks Class Section-wise Graph") fig, axesarr = plt.subplots(1, figsize=(10, 6))

sb.countplot(x='SectionID', hue='Class', data=data, hue\_order = ['L', 'M', 'H'], axes=axesarr)

plt.show() elif (ch == 7):

print("Loading Graph..\n") t.sleep(1)

print("\tMarks Class Topic-wise Graph") fig, axesarr = plt.subplots(1, figsize=(10, 6))

sb.countplot(x='Topic', hue='Class', data=data, hue\_order = ['L', 'M', 'H'], axes=axesarr)

plt.show() elif (ch == 8):

print("Loading Graph..\n") t.sleep(1)

print("\tMarks Class Stage-wise Graph") fig, axesarr = plt.subplots(1, figsize=(10, 6))

sb.countplot(x='StageID', hue='Class', data=data, hue\_order = ['L', 'M', 'H'], axes=axesarr)

plt.show() elif (ch == 9):

print("Loading Graph..\n") t.sleep(1)

print("\tMarks Class Absent Days-wise Graph") fig, axesarr = plt.subplots(1, figsize=(10, 6))

sb.countplot(x='StudentAbsenceDays', hue='Class', data=data, hue\_order = ['L', 'M', 'H'], axes=axesarr)

plt.show() if(ch == 10):

print("Exiting..\n") t.sleep(1)

#cor = data.corr() #print(cor)

data = data.drop("gender", axis=1) data = data.drop("StageID", axis=1) data = data.drop("GradeID", axis=1)

data = data.drop("NationalITy", axis=1) data = data.drop("PlaceofBirth", axis=1) data = data.drop("SectionID", axis=1) data

= data.drop("Topic", axis=1)

data = data.drop("Semester", axis=1) data = data.drop("Relation", axis=1)

data = data.drop("ParentschoolSatisfaction", axis=1) data = data.drop("ParentAnsweringSurvey", axis=1) #data = data.drop("VisITedResources", axis=1)

data = data.drop("AnnouncementsView", axis=1) u.shuffle(data)

countD = 0

countP = 0

countL = 0

countR = 0

countN = 0

gradeID\_dict = {"G-01" : 1,

"G-02" : 2,

"G-03" : 3,

"G-04" : 4,

"G-05" : 5,

"G-06" : 6,

"G-07" : 7,

"G-08" : 8,

"G-09" : 9,

"G-10" : 10,

"G-11" : 11,

"G-12" : 12}

data = data.replace({"GradeID" : gradeID\_dict}) #sig = []

for column in data.columns:

if data[column].dtype == type(object): le

= pp.LabelEncoder()

data[column] = le.fit\_transform(data[column]) ind = int(len(data) \* 0.70)

feats = data.values[:, 0:4] lbls = data.values[:,4] feats\_Train = feats[0:ind]

feats\_Test = feats[(ind+1):len(feats)] lbls\_Train = lbls[0:ind]

lbls\_Test = lbls[(ind+1):len(lbls)] modelD

= tr.DecisionTreeClassifier() modelD.fit(feats\_Train, lbls\_Train) lbls\_predD = modelD.predict(feats\_Test) for a,b in zip(lbls\_Test, lbls\_predD):

if(a==b): countD

+= 1

accD = (countD/len(lbls\_Test))

print("\nAccuracy measures using Decision Tree:") print(m.classification\_report(lbls\_Test, lbls\_predD),"\n") print("\nAccuracy using Decision Tree: ", str(round(accD, 3)))

t.sleep(1)

modelR = es.RandomForestClassifier()

modelR.fit(feats\_Train, lbls\_Train) lbls\_predR = modelR.predict(feats\_Test) for a,b in zip(lbls\_Test, lbls\_predR):

if(a==b): countR

+= 1

print("\nAccuracy Measures for Random Forest Classifier: \n") #print("\nConfusion Matrix: \n", m.confusion\_matrix(lbls\_Test, lbls\_predR))

print("\n", m.classification\_report(lbls\_Test,lbls\_predR)) accR = countR/len(lbls\_Test)

print("\nAccuracy using Random Forest: ", str(round(accR, 3))) t.sleep(1)

modelP = lm.Perceptron() modelP.fit(feats\_Train, lbls\_Train) lbls\_predP = modelP.predict(feats\_Test) for a,b in zip(lbls\_Test, lbls\_predP):

if a == b: countP

+= 1

accP = countP/len(lbls\_Test)

print("\nAccuracy measures using Linear Model Perceptron:") print(m.classification\_report(lbls\_Test, lbls\_predP),"\n") print("\nAccuracy using Linear Model Perceptron: ", str(round(accP, 3)), "\n")

t.sleep(1)

modelL = lm.LogisticRegression() modelL.fit(feats\_Train, lbls\_Train) lbls\_predL = modelL.predict(feats\_Test) for a,b in zip(lbls\_Test, lbls\_predL):

if a == b: countL

+= 1

accL = countL/len(lbls\_Test)

print("\nAccuracy measures using Linear Model Logistic Regression:") print(m.classification\_report(lbls\_Test, lbls\_predL),"\n") print("\nAccuracy using Linear Model Logistic Regression: ", str(round(accP, 3)), "\n") t.sleep(1)

modelN = nn.MLPClassifier(activation="logistic") modelN.fit(feats\_Train, lbls\_Train)

lbls\_predN = modelN.predict(feats\_Test) for a,b in zip(lbls\_Test, lbls\_predN):

#sig.append(1/(1+ np.exp(-b))) if a==b:

countN += 1

#print("\nAverage value of Sigmoid Function: ", str(round(np.average(sig), 3)))

print("\nAccuracy measures using MLP Classifier:") print(m.classification\_report(lbls\_Test, lbls\_predN),"\n") accN = countN/len(lbls\_Test)

print("\nAccuracy using Neural Network MLP Classifier: ", str(round(accN, 3)), "\n")

choice = input("Do you want to test specific input (y or n): ") if(choice.lower()=="y"):

gen = input("Enter Gender (M or F): ") if (gen.upper() == "M"):

gen = 1

elif (gen.upper() == "F"): gen = 0

nat = input("Enter Nationality: ") pob = input("Place of Birth: ")

sta = input("Enter Stage ID(Lower level, Middle school, High school):

")

if (sta == "Lower level"): sta = 2

elif(sta == "Middle school"): sta = 1

elif (sta == "High school"): sta = 0

gra = input("Grade ID as (G-<grade>): ") if(gra

== "G-02"):

gra = 2

elif (gra == "G-04"): gra = 4

elif (gra == "G-05"): gra = 5

elif (gra == "G-06"): gra = 6

elif (gra == "G-07"): gra = 7

elif (gra == "G-08"): gra = 8

elif (gra == "G-09"): gra = 9

elif (gra == "G-10"): gra = 10

elif (gra == "G-11"): gra = 11

elif (gra == "G-12"): gra = 12

sec = input("Enter Section: ") top = input("Enter Topic: ")

sem = input("Enter Semester (F or S): ") if (sem.upper() == "F"):

sem = 0

elif (sem.upper() == "S"): sem = 1

rel = input("Enter Relation (Father or Mum): ") if (rel == "Father"):

rel = 0

elif (rel == "Mum"): rel = 1

rai = int(input("Enter raised hands: "))

res = int(input("Enter Visited Resources: "))

ann = int(input("Enter announcements viewed: ")) dis = int(input("Enter no. of Discussions: "))

sur = input("Enter Parent Answered Survey (Y or N): ") if (sur.upper() == "Y"):

sur = 1

elif (sur.upper() == "N"): sur = 0

sat = input("Enter Parent School Satisfaction (Good or Bad): ") if (sat == "Good"):

sat = 1

elif (sat == "Bad"): sat = 0

absc = input("Enter No. of Abscenes(Under-7 or Above-7): ") if (absc == "Under-7"):

absc = 1

elif (absc == "Above-7"): absc = 0

arr = np.array([rai, res, dis, absc])

#arr = np.array([gen, rnd.randint(0, 30), rnd.randint(0, 30), sta, gra, rnd.randint(0, 30), rnd.randint(0, 30), sem, rel, rai, res, ann, dis, sur, sat, absc])

predD = modelD.predict(arr.reshape(1, -1)) predR = modelR.predict(arr.reshape(1, -1)) predP = modelP.predict(arr.reshape(1, -1)) predL = modelL.predict(arr.reshape(1, -1)) predN = modelN.predict(arr.reshape(1, -1)) if (predD == 0):

predD = "H" elif (predD == 1):

predD = "M" elif (predD == 2):

predD = "L" if (predR == 0):

predR = "H" elif (predR == 1):

predR = "M" elif (predR == 2):

predR = "L" if (predP == 0):

predP = "H" elif (predP == 1):

predP = "M" elif (predP == 2):

predP = "L" if (predL == 0):

predL = "H" elif (predL == 1):

predL = "M" elif (predL == 2):

predL = "L" if (predN == 0):

predN = "H" elif (predN == 1):

predN = "M" elif (predN == 2):

predN = "L" t.sleep(1)

print("\nUsing Decision Tree Classifier: ", predD)

t.sleep(1)

print("Using Random Forest Classifier: ", predR) t.sleep(1)

print("Using Linear Model Perceptron: ", predP) t.sleep(1)

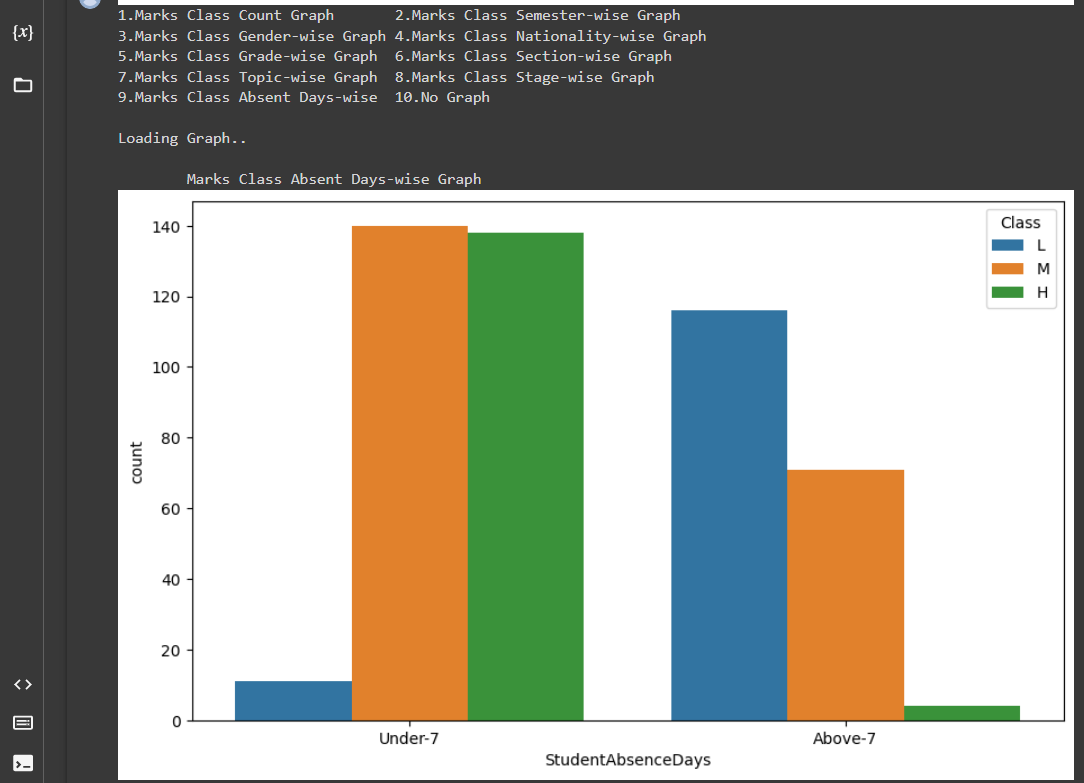
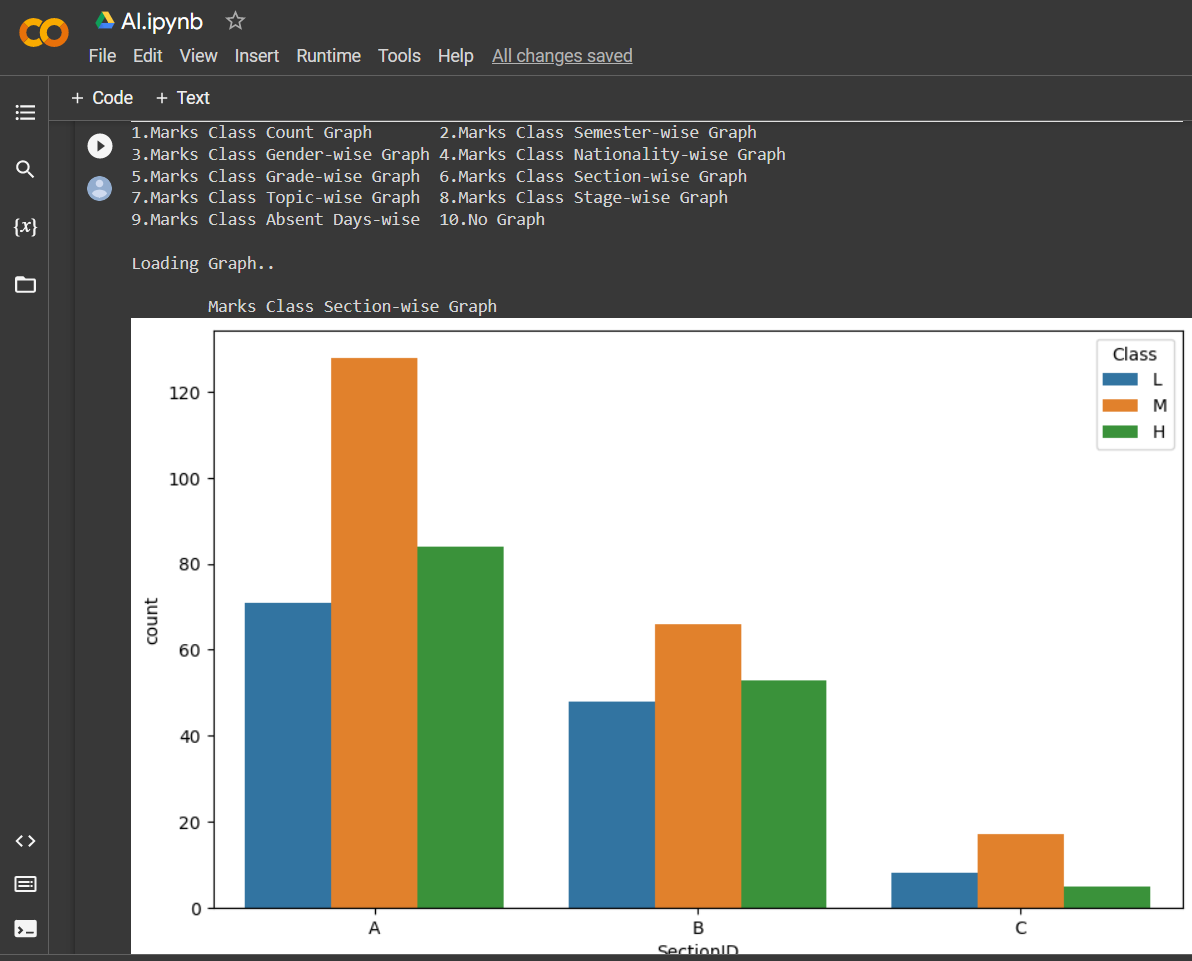
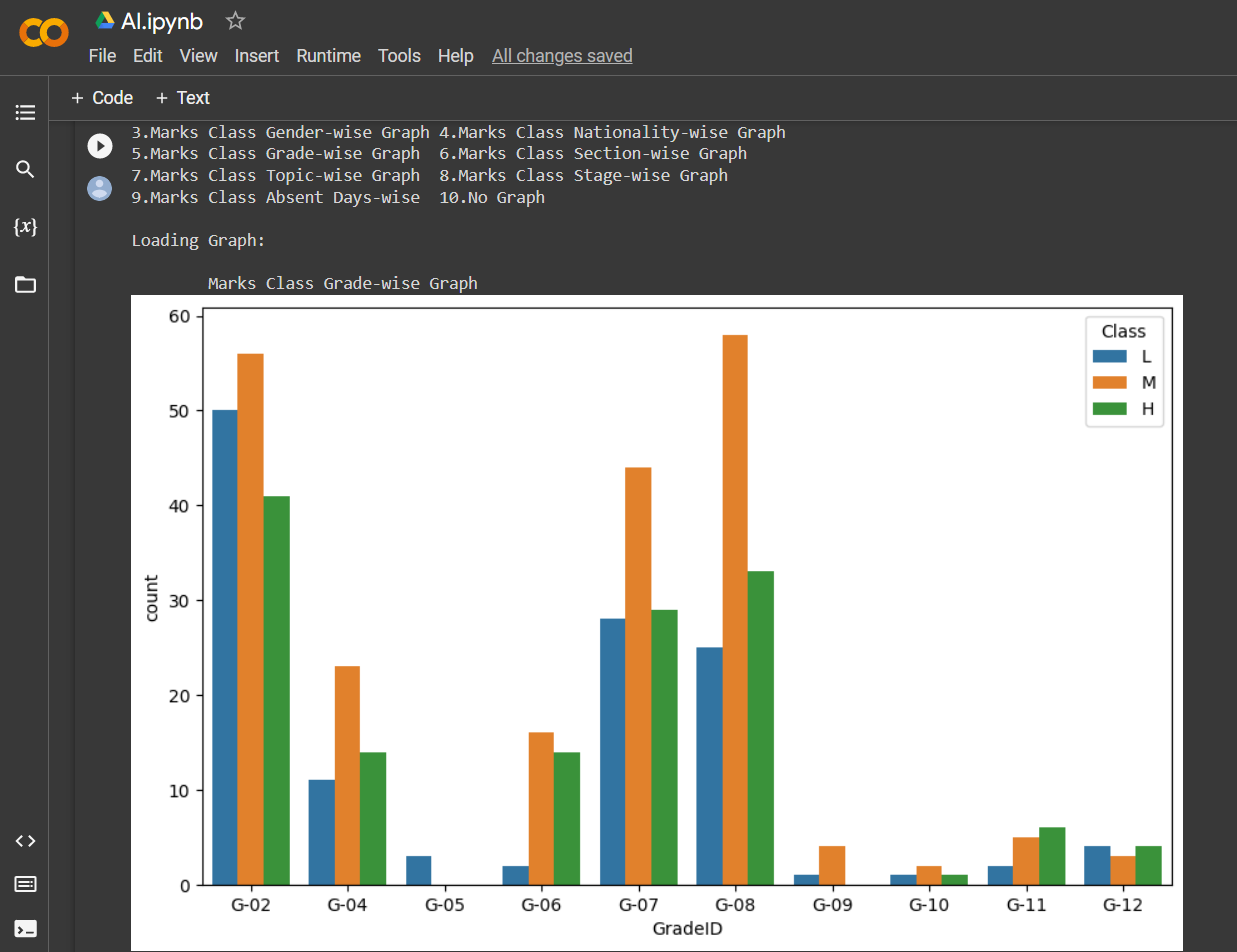
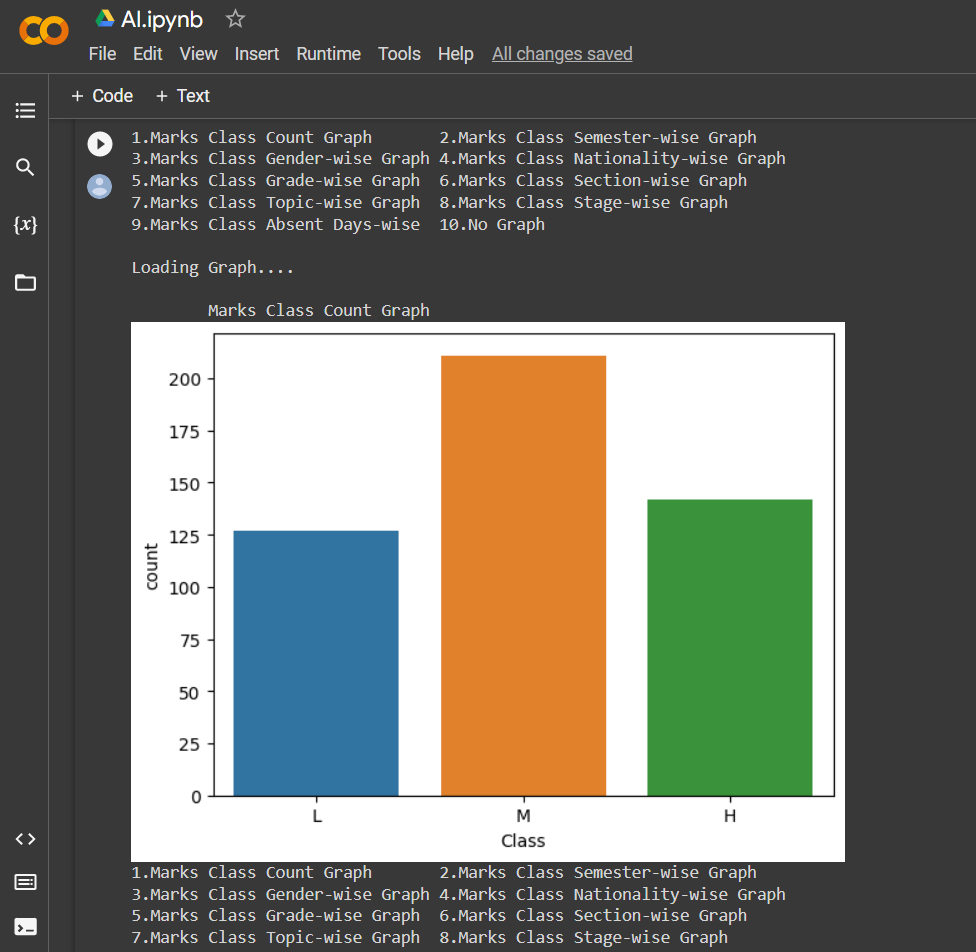
print("Using Linear Model Logisitic Regression: ", predL) t.sleep(1)

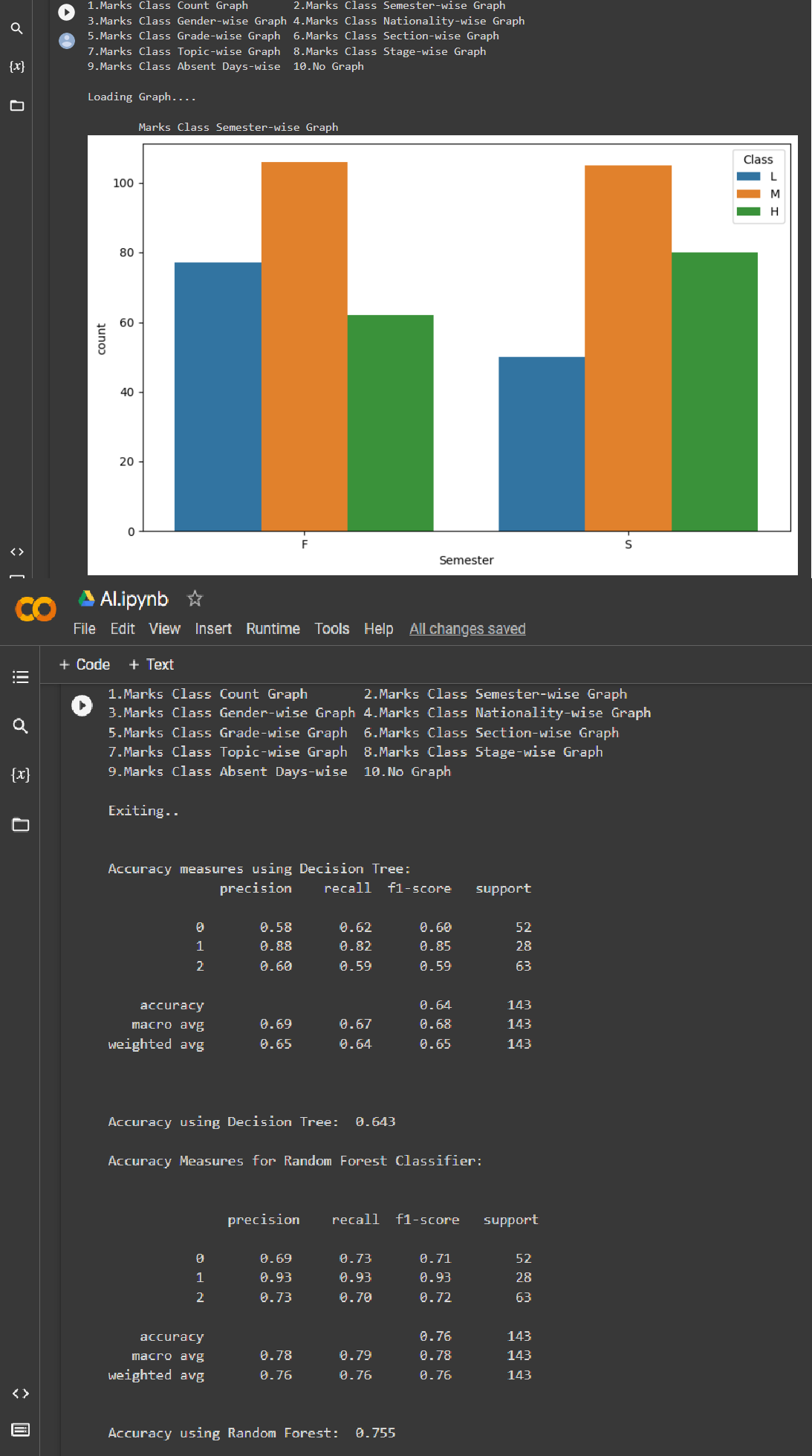
print("Using Neural Network MLP Classifier: ", predN) print("\nExiting...")

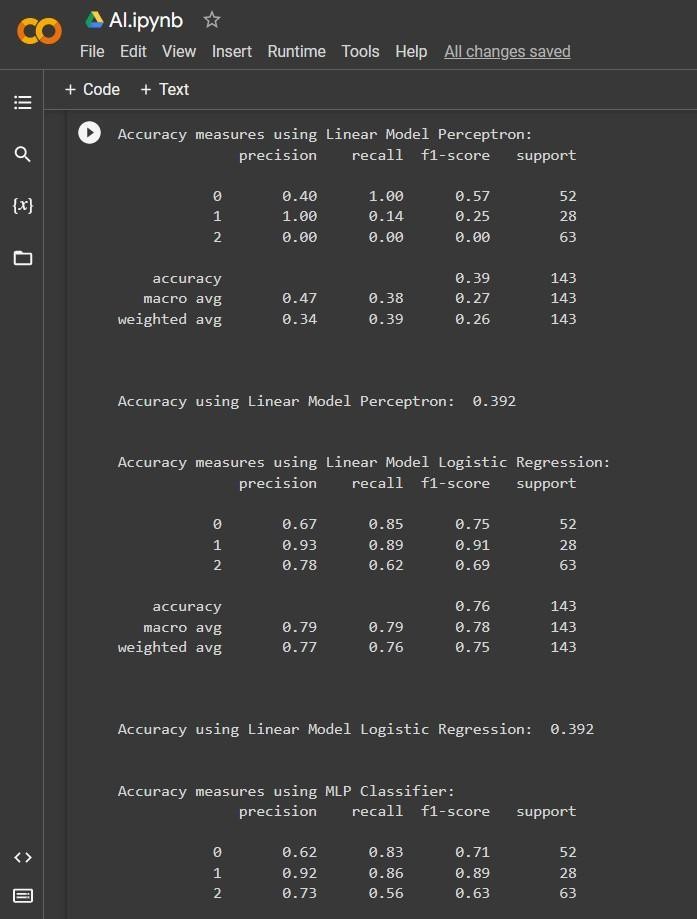
t.sleep(1) else:

print("Exiting..") t.sleep(1)

RESULTS







CHAPTER VI

CONCLUSION AND FUTURE ENHANCEMENT

# CONCLUSION:

To summarise, a student performance prediction model is a powerful tool that enables educators and administrators to identify kids who are at risk of academic underachievement. Through the implementation of a clearly defined technique encompassing data collection, preprocessing, feature selection, model selection, training, testing, and validation, we are able to construct precise models that yield significant insights into the determinants of academic achievement.

Nevertheless, it is crucial to recognise the ethical ramifications of employing such a model and to guarantee its responsible and transparent usage. It is imperative to exercise caution in order to prevent biases and ensure the confidentiality and safety of student data.

In summary, the creation of a student performance prediction model is a continuous endeavour that necessitates constant monitoring and refinement. By persistently investigating novel methodologies and strategies, we can construct models that are more precise, efficient, and morally sound, ultimately contributing to the enhancement of student outcomes and the advancement of academic achievement.

FUTURE ENHANCEMENT:

There are multiple methods by which a student performance prediction model can be augmented in the future. Some possible areas for enhancement encompass:

* 1. Incorporating more diverse and relevant data:

Current models currently utilise academic and demographic data, but the inclusion of supplementary data such as social media activity, extracurricular activities, and mental health indicators could offer important insights into the elements that contribute to academic achievement.

* 1. Using more advanced machine learning models:

Deep learning and reinforcement learning models have demonstrated encouraging outcomes in forecasting student performance, and future investigation of these models could potentially result in even higher levels of precision.

* 1. Addressing ethical concerns:

It is crucial for the model to be transparent, neutral, and to prioritise student privacy in order to achieve success. Further study and development are required to effectively address these challenges.

* 1. Developing personalised interventions:

After a student has been recognised as at-risk, the model can be utilised to create tailored treatments that target their individual needs and difficulties. This may encompass suggestions for supplementary resources, tutoring, or counselling services.

* 1. Improving data quality and availability:

Acquiring data that is both of superior quality and covers a wide range of information is crucial for ensuring the precision of the model. It is imperative to enhance data gathering procedures and enhance the accessibility of data to academics and practitioners.

In summary, the creation of a student performance prediction model is a continuous endeavour that necessitates constant ingenuity and cooperation among academics, educators, and policymakers. Through ongoing enhancements to the model and diligent adherence to ethical and responsible practices, we can contribute to the enhancement of student outcomes and the advancement of academic achievement for all students.

**REFERENCES**

1. J. Xu, K. H. Moon, and M. Van Der Schaar, “A Machine Learning Approach for Tracking and Predicting Student Performance in Degree Programs,” IEEE J. Sel. Top. Signal Process., vol. 11, no. 5, pp. 742–753, 2017.
2. K. P. Shaleena and S. Paul, “Data mining techniques for predicting student performance,” in ICETECH 2015 - 2015 IEEE International Conference on Engineering and Technology, 2015, no. March, pp. 0–2.
3. A. M. Shahiri, W. Husain, and N. A. Rashid, “A Review on Predicting Student’s Performance Using Data Mining Techniques,” in Procedia Computer Science, 2015.
4. Y. Meier, J. Xu, O. Atan, and M. Van Der Schaar, “Predicting grades,” IEEE Trans. Signal Process., vol. 64, no. 4, pp. 959–972, 2016.
5. P. Guleria, N. Thakur, and M. Sood, “Predicting student performance using decision tree classifiers and information gain,” Proc. 2014 3rd Int. Conf. Parallel, Distrib. Grid

Comput. PDGC 2014, pp. 126–129, 2015.

1. P. M. Arsad, N. Buniyamin, and J. L. A. Manan, “A neural network students’ performance

prediction model (NNSPPM),” 2013 IEEE Int. Conf. Smart Instrumentation, Meas. Appl. ICSIMA 2013, no. July 2006, pp. 26–27, 2013.

1. K. F. Li, D. Rusk, and F. Song, “Predicting student academic performance,” Proc. - 2013 7th Int. Conf. Complex, Intelligent, Softw. Intensive Syst. CISIS 2013, pp. 27–33, 2013.
2. G. Gray, C. McGuinness, and P. Owende, “An application of classification models to predict learner progression in tertiary education,” in Souvenir of the 2014 IEEE International Advance Computing Conference, IACC 2014, 2014.
3. N. Buniyamin, U. Bin Mat, and P. M. Arshad, “Educational data mining for prediction and classification of engineering students achievement,” 2015 IEEE 7th Int. Conf. Eng. Educ. ICEED 2015, pp. 49–53, 2016.
4. Z. . Alharbi, J. . Cornford, L. . Dolder, and B. . De La Iglesia, “Using data mining techniques to predict students at risk of poor performance,” Proc. 2016 SAI Comput. Conf. SAI 2016, pp. 523–531, 2016.