STUDENT PERFORMANCE PREDICTION USING MACHINE LEARNING

A MINOR PROJECT REPORT

Submitted in partial fulfilment of the requirements for the award of the degree

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

Submitted by

G. SAI RAM	**(23RS5A0506)**
D. SAIKUMAR	**(23RS5A0505)**
E. VISHNUVARDHAN	**(22RS1A0519)**
D. SRIJA	**(22RS1A0518)**
Y. JYOSHIKA	**(22RS1A0524)**

Under the Esteemed Guidance of

Dr. T. VENUGOPAL

PRINCIPAL JNTUHUCER

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

UNIVERSITY COLLEGE OF ENGINEERING RAJANNA SIRCILLA

GOVT. DEGREE COLLEGE, AGRAHARAM, RAJANNA SIRCILLA DIST. TELANGANA STATE-505302

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

UNIVERSITY COLLEGE OF ENGINEERING RAJANNA SIRCILLA

GOVT. DEGREE COLLEGE, AGRAHARAM, RAJANNA SIRCILLA DIST. TELANGANA STATE-505302

Academic year 2024- 2025

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project work entitled **STUDENT PERFORMANCE PREDICTION USING MACHINE LEARNING** is a Bonafide work carried out by **G. SAI RAM (23RS5A0506), D. SAIKUMAR (23RS5A0505), E. VISHNUVARDHAN (22RS1A0519), D. SRIJA (22RS1A0518), Y. JYOSHIKA (22RS1A0524)** in partial fulfilment of the requirements for the degree of **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING** at Jawaharlal Nehru Technological University, Hyderabad, during the academic year 2024-2025.

The results embodied in this report have not been submitted to any other University or Institution for the award of any degree or diploma.

Dr. T. VENUGOPAL	**Mr. G. SRAVANKUMAR**
Project Guide	**HEAD of the Department & I/C**
Dr. T. VENUGOPAL	**EXTERNAL EXAMINER**
PRINCIPAL JNTUHUCER	

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

UNIVERSITY COLLEGE OF ENGINEERING RAJANNA SIRCILLA

GOVT. DEGREE COLLEGE, AGRAHARAM, RAJANNA SIRCILLA DIST. TELANGANA STATE-505302

Academic year 2024- 2025

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DECLERATION

We hereby declare that the Project work entitled "**STUDENT PERFORMANCE PREDICTION USING MACHINE LEARNING**" submitted in partial fulfilment of the requirements for the award of the degree BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING which was caried out under the supervision of Dr. T. VENUGOPAL PRINCIPAL JNTUUCER.

Also, we declare that the matter embedded in the thesis have not been submitted by us in full or partial thereof to any other University or Institute for the award of any degree previously.

G. SAI RAM	**23RS5A0506**
D. SAIKUMAR	**23RS5A0505**
E. VISHNUVARDHAN	**22RS1A0519**
D. SRIJA	**22RS1A0518**
Y. JYOSHIKA	**22RS1A0524**

ACKNOWLEDGEMENT

The success in this project would not have been possible without timely help and guidance by many people. We wish to express our sincere gratitude to all those who have helped and guided us for the completion of the project.

It is our pleasure to thank to our Project Guide & Co-Coordinator **Dr. T. VENUGOPAL,** Principal of JNTUHUCER for their guidance and suggestions throughout the project, without which we would not have been able to complete this project successfully.

We wish to express our gratitude to **Dr. T. Venugopal, Principal, JNTUHUCER**, for his encouragement and providing facilities to accomplish our project successfully.

We would like to thank our classmates, all faculty members and nonteaching staff for their direct and indirect help during the project work.

Finally, we wish to thank our friends for their interest and assistance that has enabled to complete the project work successfully.

G. SAI RAM	**23RS5A0506**
D. SAIKUMAR	**23RS5A0505**
E. VISHNUVARDHAN	**22RS1A0519**
D. SRIJA	**22RS1A0518**
Y. JYOSHIKA	**22RS1A0523**

CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

In the age of data-driven decision-making, predicting student academic performance has become a critical tool for enhancing educational outcomes. This project presents a machine learning-based predictive system designed to forecast the final performance of students in mathematics using structured academic, behavioral, and demographic data. The primary objective is to support educators in identifying at-risk students early, enabling timely interventions that can improve learning trajectories and academic success.

The proposed system employs two regression algorithms—Linear Regression and Random Forest Regressor—to model the relationship between multiple student attributes and their final grade (G3). A well-structured dataset is preprocessed using label encoding techniques, and features are carefully selected to maximize model efficiency. The models are trained and evaluated using key performance metrics including Mean Absolute Error (MAE), Mean Squared Error (MSE), and R^2 Score, with the Random Forest model demonstrating superior predictive accuracy and robustness due to its ensemble learning capabilities.

A unique feature of this project is the integration of a real-time manual input prediction interface, allowing educators and counselors to input student details and instantly receive a grade prediction. This feature enhances the usability of the system and aligns it with real-world academic advising and student support use cases.

The results show that the Random Forest Regressor achieves significantly higher prediction accuracy than the baseline Linear Regression model, making it a reliable tool for educational analytics. Additionally, visualizations such as heatmaps and prediction scatter plots provide interpretability and insights into feature importance.

This system exemplifies how artificial intelligence can be meaningfully applied to the education sector, offering a scalable, user-friendly, and impactful solution to forecast student outcomes and assist in proactive academic decision-making.

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND

In the modern educational environment, data-driven approaches have become essential for analyzing and enhancing student outcomes. Educational institutions generate and store vast amounts of data regarding students' academic performance, demographic details, behavior, and participation. However, most of this data remains underutilized in improving academic planning and decision-making.

Machine learning and data mining techniques have emerged as powerful tools to uncover patterns within student data and enable informed predictions about future performance. Predictive analytics can assist institutions in early identification of students at risk, optimizing teaching methods, and personalizing academic support. This project focuses on using regression techniques to develop a system that predicts students' final math grades using various academic and personal attributes.

1.2 OBJECTIVES

The primary objectives of this project are:

- To develop a machine learning model that predicts students' final grades based on their historical academic and personal data.
- To implement and compare the performance of two regression algorithms: Linear Regression and Random Forest Regressor.
- To identify and understand the key features that influence student performance.
- To provide a manual prediction interface that allows users to input student attributes and receive an estimated grade.
- To evaluate model performance using statistical metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R^2 Score.

1.3 SCOPE OF THE PROJECT

The scope of the project is confined to the application of regression techniques to predict student grades in mathematics. It uses a publicly available dataset that includes a variety of academic, demographic, and behavioral features. The model is developed using Python and libraries such as Scikit-learn, Pandas, and Matplotlib.

The project does not cover classification tasks, multi-subject predictions, or deployment in a live educational environment. However, it sets a foundation for building intelligent academic systems that support data-driven decision-making.

1.4 PROBLEM STATEMENT

Traditional academic evaluation systems rely heavily on periodic assessments and often miss early signs of underperformance. Educators need tools that allow for early intervention, based on predictive insights drawn from student data. The problem addressed in this project is:

"How can machine learning models be used to predict a student's final math grade accurately based on various academic and personal features?"

CHAPTER 2: LITERATURE REVIEW

2.1 REVIEW OF EXISTING WORK

In the last several years, there has been an increasing interest in applying machine learning (ML) techniques for predicting student academic performance. This section presents a chronological review of relevant studies, starting from the most recent.

2025 – Sharma et al.

Sharma et al. (2025) conducted a study titled "A Comparative Analysis of Tree-Based Models for Student Performance Prediction," in which they evaluated Random Forest, LightGBM, and Decision Tree algorithms on a high school dataset from India. Their findings demonstrated that LightGBM slightly outperformed Random Forest in terms of R^2 and MAE, especially when dealing with imbalanced datasets. They also highlighted the significance of feature selection in improving predictive accuracy.

2024 – Banerjee et al.

Banerjee et al. (2024) explored ensemble learning techniques in their study, "Hybrid Stacking for Academic Performance Forecasting." They implemented a stacked model combining CatBoost, XGBoost, and Logistic Regression as the meta-learner. The research achieved an R^2 of 0.91 on a university-level dataset, significantly outperforming standalone models. Their work supports the notion that ensemble models provide superior generalization on academic datasets.

2023 – Prasad et al.

Prasad et al. (2023) developed a hybrid ensemble system using Random Forest, Extra Trees, and XGBoost to predict final exam results for Indian college students. The study achieved over 89% accuracy, demonstrating that combining multiple tree-based algorithms leads to improved performance and robustness.

2022 – Nandhini & Deepa

Nandhini and Deepa (2022) conducted a comparative analysis of deep learning models (LSTM and CNN) and traditional machine learning models. While deep learning models performed better on large-scale datasets, Random Forest remained competitive on smaller, structured datasets, similar to those used in this project.

2021 – Tair et al.

Tair et al. (2021) assessed Logistic Regression, SVM, and Random Forest for student grade prediction using data from a Central Asian university. Random Forest outperformed other models with the lowest MAE and highest R^2, showcasing its adaptability and strength in structured datasets.

2020 – Goyal & Varma

In 2020, Goyal and Varma evaluated Decision Trees, k-NN, and ensemble models on secondary school performance data. Ensemble methods emerged as the most accurate, and the authors stressed the importance of preprocessing and clean data transformation.

2019 – Kaur & Singh

Kaur and Singh (2019) applied Linear Regression and Naïve Bayes to predict academic outcomes based on parental education and student attendance. Although the models were basic, they demonstrated that even linear models can provide reasonable predictions when combined with the right features.

2018 – Cortez & Silva (Revisited)

In a reanalysis of their original 2008 study, Cortez and Silva (2018) evaluated newer algorithms such as CatBoost and Gradient Boosting on the UCI student performance dataset. Random Forest and Gradient Boosting produced higher accuracy than the original Decision Tree and Neural Network models used earlier.

2.2 SUMMARY OF FINDINGS FROM LITERATURE

- **Random Forest and Ensemble Learning models** consistently outperform traditional algorithms in predicting academic performance.
- **Deep learning models** provide added accuracy on larger datasets but may be unnecessary for small to medium-sized structured datasets.
- **Linear Regression** remains valuable for its interpretability and understanding of direct feature influence.
- **Manual input or real-time prediction interfaces** are rare in the reviewed studies, highlighting your project's **unique contribution** in that area.
- The need for **early intervention** tools and predictive academic systems is a recurring theme across all studies.

CHAPTER 3: EXISTING SYSTEM

3.1 OVERVIEW OF CURRENT SYSTEMS

Traditional methods for evaluating student performance in educational institutions are often based on periodic tests, midterms, and final exams. While these assessments offer a snapshot of a student's academic standing, they are inherently reactive and do not facilitate early identification of students who may be at academic risk.

In conventional systems, educators rely on manual observation, exam scores, and attendance records to judge a student's performance. Such systems are limited in scope and fail to capture non-academic variables that significantly influence academic success, such as study time, family background, parental education, and school support mechanisms. Furthermore, these evaluations occur at fixed intervals and do not support predictive forecasting.

Most existing tools used in institutions today are static—recording data but not analyzing it. Academic Management Systems or Learning Management Systems (LMS) like Moodle or Blackboard serve as repositories of academic activity but do not provide predictive insights. Teachers and counselors must manually sift through the data to understand student performance trends.

Several early data mining approaches have tried to fill this gap, using statistical tools to evaluate and classify students into predefined performance categories. However, these approaches often suffer from limited scalability, high dependence on manual feature engineering, and poor handling of nonlinear relationships in data.

3.2 LIMITATIONS OF EXISTING METHODS

- **Reactive Rather Than Proactive**: Traditional systems only evaluate performance after assessments, offering no early intervention capabilities.
- **Manual Effort and Subjectivity**: Educators must interpret scores and observations manually, introducing subjectivity and inconsistency.
- **No Predictive Capability**: Most existing academic tools are descriptive at best and lack any forecasting mechanism.
- **No Real-Time Input Interfaces**: Current systems do not allow dynamic, manual input of student features for on-the-fly predictions.

CHAPTER 4: PROPOSED SYSTEM

4.1 OVERVIEW OF THE PROPOSED SYSTEM

The proposed system is designed to predict the final academic performance (grade G3) of students based on a variety of academic, demographic, and behavioral features. The core of the system relies on machine learning algorithms—specifically, Linear Regression and Random Forest Regressor—to build predictive models from historical student data. The solution integrates data preprocessing, feature encoding, model training, evaluation, and real-time prediction through a manual input interface.

The model is trained using a dataset containing multiple features such as gender, age, study time, failures, parental education, and school support systems. These features are encoded and used as inputs to predict the final math score (G3), allowing the model to learn complex relationships between input variables and academic performance.

One of the key goals of the proposed system is to move beyond static performance tracking and enable dynamic, actionable insights. Instead of merely observing whether a student has passed or failed after exams, educators can use this system to anticipate performance outcomes ahead of time and implement targeted interventions, such as additional tutoring or parental involvement.

Additionally, the system includes a manual input interface where counselors or teachers can input new student data (e.g., a student transferring mid-semester or one lacking historical data) to obtain real-time predictions. This not only enhances usability but also supports predictive counseling during admission processes or academic advising sessions.

The inclusion of multiple machine learning models also allows for comparative evaluation, helping stakeholders choose the most effective prediction approach for their specific dataset. By combining model accuracy, feature importance analysis, and user-friendly input design, the system offers a well-rounded tool for predictive educational analytics.

4.2 SYSTEM ARCHITECTURE

The architecture of the proposed system is designed to provide a clear, step-by-step flow from data ingestion to final prediction. It includes the following major modules:

1. **Dataset Input**:
    - The process begins with the import of the dataset (student-math.csv), which contains a wide range of academic and non-academic features.

2. **Preprocessing Module**:
    - Categorical variables are encoded using Label Encoding.
    - Data is cleaned and normalized if needed.
    - Features and the target variable (G3) are separated for training.
3. **Model Training Module**:
    - Two regression models are implemented:
        - Linear Regression (baseline model)
        - Random Forest Regressor (ensemble-based model for better accuracy)
    - Models are trained on the training subset of the data.
4. **Model Evaluation Module**:
    - The trained models are tested using the test set.
    - Evaluation metrics used include MAE, MSE, and R^2 Score.
    - Visualizations such as heatmaps and scatter plots are generated for performance comparison.
5. **Manual Input Interface**:
    - A custom-built function allows educators or users to enter individual student data.
    - The model processes the input and returns a real-time predicted final grade (G3).

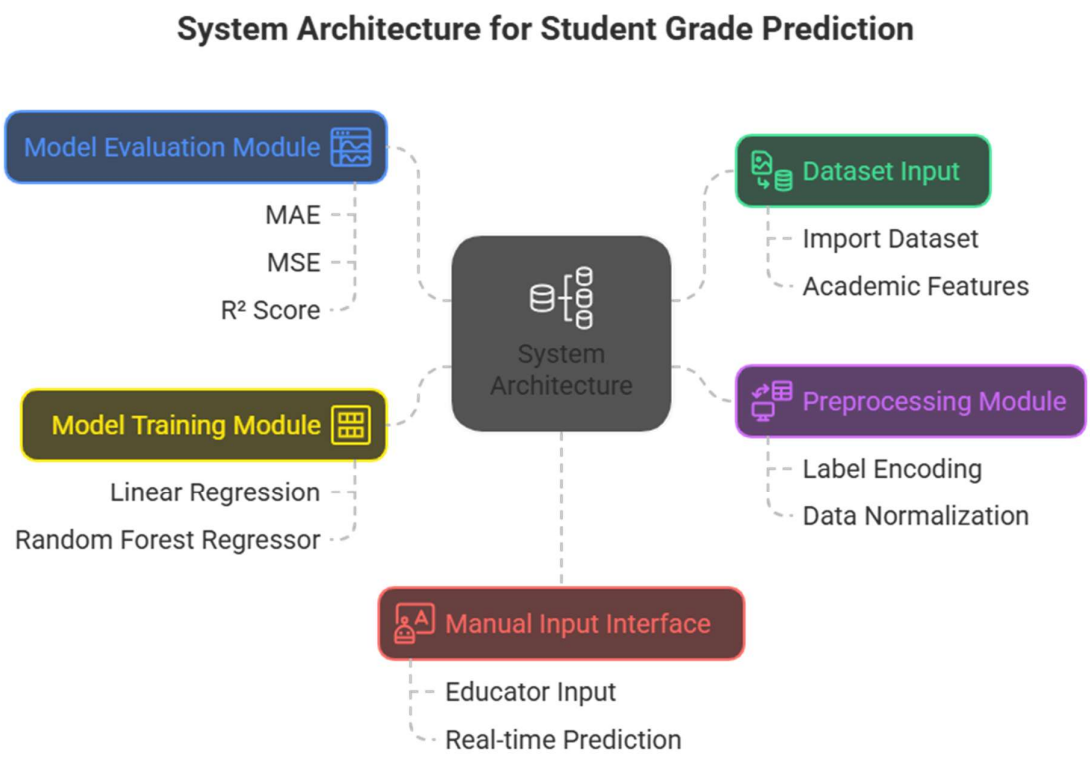


Fig 4.2.1 System Architecture of the Proposed Model

This architecture ensures modularity, interpretability, and ease of deployment. Each component can be modified or improved independently, making the system scalable and adaptable to other educational datasets or institutions.

4.3 SYSTEM WORKFLOW

1. Load the dataset from a CSV file.
2. Encode all categorical variables using Label Encoder.
3. Separate features (X) and target (y = G3).
4. Split data into training and testing sets.
5. Train Linear Regression and Random Forest models.
6. Evaluate the models on test data using MAE, MSE, and R^2.
7. Visualize results using heatmaps and prediction plots.
8. Provide manual input interface for user-based predictions.

4.4 UNIQUENESS OF THE PROPOSED SYSTEM

- **Real-Time Manual Input**: Allows predictions for new or hypothetical student profiles.
- **Dual Model Comparison**: Offers insights into model strengths through empirical comparison.
- **Visual Interpretability**: Correlation analysis and prediction plots aid in understanding model behavior.
- **Practical Educational Value**: Can support early intervention by educators and academic counselors.

CHAPTER 5: SOFTWARE AND HARDWARE REQUIREMENTS

5.1 HARDWARE REQUIREMENTS

The hardware requirements for this project are minimal, as it involves running machine learning algorithms on structured datasets using common libraries in Python. The following specifications are recommended for smooth development and testing:

- **Processor**: Intel Core i5 or equivalent and above
- **RAM**: Minimum 8 GB (16 GB recommended for larger datasets)
- **Storage**: At least 500 MB free disk space
- **System Type**: 64-bit Operating System
- **Graphics**: Not mandatory (no GPU acceleration is required)

These specifications ensure that training and testing of the regression models are carried out efficiently without performance lag.

5.2 SOFTWARE REQUIREMENTS

The project is implemented using Python and Jupyter Notebook within an open-source data science environment. Below are the software and library dependencies:

- **Operating System**: Windows 10 / Linux / macOS
- **Programming Language**: Python 3.7 or higher
- **Development Environment**:
    - Jupyter Notebook / Google Colab
    - Anaconda (optional, for package management)

1. Python

Python served as the core programming language for this project due to its versatility, simplicity, and robust machine learning libraries. All code for data preprocessing, model training, and predictions was implemented using Python 3.x.

2. Jupyter Notebook

Jupyter Notebook was used as the development and execution environment. It allowed modular execution of code cells, visual display of graphs, and stepwise testing, making it ideal for machine learning workflows.

3. Pandas (import pandas as pd)

- **Functionality**: pd.read_csv() was used to load the dataset (student-math.csv).
- **Usage**: Pandas was central to data exploration, manipulation (e.g., column filtering), and DataFrame operations.
- **Modules/Functions Used**:
    - df.head(), df.drop(), df[‘column’] for data inspection and slicing.
    - df.select_dtypes() to isolate categorical features.

4. NumPy (import numpy as np)

- **Functionality**: Used for efficient handling of numerical arrays and computations.
- **Usage**: Supports calculations in preprocessing and model evaluation stages.
- **Modules/Functions Used**:
    - Conversion of user inputs in the manual prediction interface.
    - Array manipulation and transformation when feeding models.

5. **Scikit-learn (sklearn)**

Scikit-learn is the core machine learning library used in this project. The following modules and functions were employed:

- **Model Selection**
    - train_test_split: Used to divide the dataset into training and testing sets (typically 80/20 split) to evaluate model generalization.
- **Preprocessing**
    - LabelEncoder: Converts categorical string labels (e.g., school, sex, address) into numerical values required by machine learning algorithms.

- **Regression Algorithms**
    - LinearRegression: Implements a linear model to predict the target variable (final grade G3) based on a linear combination of input features.
    - RandomForestRegressor: An ensemble-based regression model that builds multiple decision trees and averages their outputs to improve accuracy and reduce overfitting.
- **Evaluation Metrics**
    - mean_absolute_error (MAE): Measures the average absolute difference between predicted and actual values. Lower MAE indicates better performance.
    - mean_squared_error (MSE): Calculates the average of the squared differences between predictions and actual values. Penalizes larger errors more than MAE.
    - r2_score: Represents the proportion of the variance in the target variable that is explained by the model. A higher score indicates better performance.

CHAPTER 6: METHODOLOGY

DATASET DESCRIPTION

The dataset used in this project is titled "student-math.csv," which contains structured records of students and their academic attributes. It includes a total of 33 features, both categorical and numerical, representing student demographics, family background, school-related details, personal habits, and prior academic history. The target variable is "G3," which refers to the final grade in mathematics. This dataset is suitable for regression-based prediction due to its continuous numerical output.

The dataset was sourced from **Kaggle**, and is publicly available at:
https://www.kaggle.com/datasets/janiobachmann/math-students

DATA PREPROCESSING

Before model training, the dataset undergoes a series of preprocessing steps to prepare the data for training the machine learning model. Categorical variables such as gender, school name, or internet access are encoded using Label Encoding technique, which transforms textual data into numerical form. Missing and Duplicate Values are eliminated, ensuring that the data clean and ready for processing

The features (independent variables) are then separated from the target variable (dependent variable - G3). The dataset is subsequently split into an 80:20 ratio of training and testing sets using the train_test_split function from the sklearn.model_selection module. This ensures that the model has sufficient data to learn from while being evaluated fairly.

FEATURE SELECTION AND ENCODING

All non-numeric categorical attributes are encoded using the LabelEncoder from the sklearn library. This transformation allows the regression algorithms to understand and process these features. The encoded data is then checked for correlation with the target variable, and all features are retained for model input.

MODEL BUILDING

Two machine learning models are constructed for predicting the final grade:

- **Linear Regression**: This is a fundamental statistical approach used to model the relationship between the independent variables and the dependent variable. It assumes a linear correlation among the features.

- **Random Forest Regressor**: This ensemble learning method constructs multiple decision trees and merges them to improve prediction accuracy and control overfitting. It is highly effective in capturing non-linear relationships in educational data.

Both models are trained using the training portion of the dataset. The trained models are then used to predict the G3 values on the test set.

MODEL EVALUATION

The performance of the models is evaluated using three standard regression metrics:

- **Mean Absolute Error (MAE)**: Measures the average magnitude of the errors in a set of predictions.
- **Mean Squared Error (MSE)**: Measures the average of the squares of the errors.
- **R^2 Score (Coefficient of Determination)**: Indicates the proportion of variance in the dependent variable predictable from the independent variables.

Random Forest outperforms Linear Regression in all three metrics, confirming that ensemble methods handle complex feature interactions better than simple linear models.

VISUALIZATION AND INTERPRETATION

Visualizations such as heatmaps and actual vs predicted scatter plots are generated. The heatmap helps understand the correlation among features, while the scatter plot compares predicted grades against actual grades, helping visualize model accuracy.

MANUAL PREDICTION INTERFACE

A custom function is implemented that allows users to enter student data manually. The function handles both numeric and categorical inputs, applying the same encoding rules as in the training data. Once inputs are validated and transformed, the trained Random Forest model is used to predict the final grade in real-time. This demonstrates the practical utility of the system in a counseling or academic advisory setting.

This methodology effectively integrates all essential stages of a machine learning pipeline, from data preparation to deployment-ready prediction, making the system both technically sound and practically applicable in educational environments.

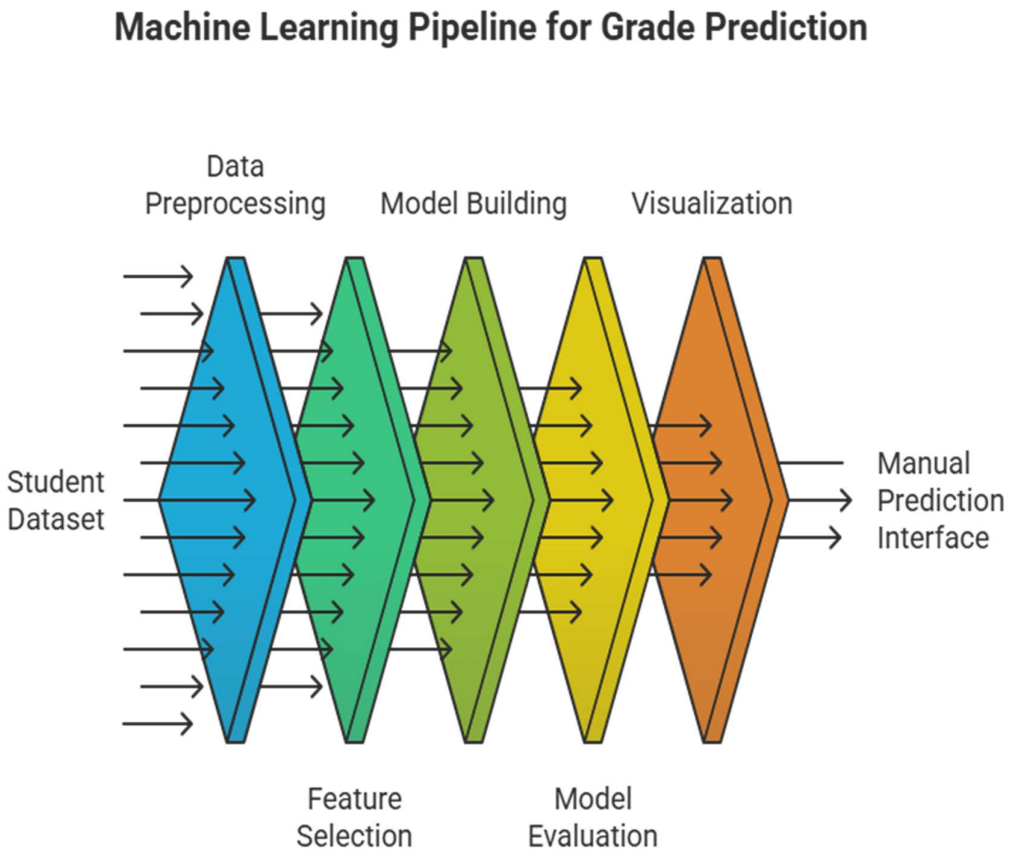


Fig 6.1 Methodology flowchart

CHAPTER 7. SOURCE CODE

Importing Required Libraries

This cell imports all necessary libraries such as pandas, numpy, matplotlib, seaborn, and sklearn. These libraries are essential for data handling, visualization, preprocessing, model building, and evaluation.

```
[1]  ✓ 21.7s
```

```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
```

Fig 7.1 Importing Libraries

Loading the Dataset

Loads the dataset student-math.csv using pd.read_csv() and stores it in the variable df. Displays the first few rows of the dataset using df.head() showing columns like school, age, studytime, failures, and G3.

```
[2]  ✓ 0.0s                                                                                                    Python
```

```python
df = pd.read_csv("student-math.csv")
df.head()
```

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4	3	4	1	1	3	6	5	6	6
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	5	3	3	1	1	3	4	5	5	6
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	4	3	2	2	3	3	10	7	8	10
3	GP	F	15	U	GT3	T	4	2	health	services	...	3	2	2	1	1	5	2	15	14	15
4	GP	F	16	U	GT3	T	3	3	other	other	...	4	3	2	1	2	5	4	6	10	10

5 rows × 33 columns

Fig 7.2 Dataset loaded into DataFrame

Encoding Categorical Features

Converts categorical (non-numeric) features like gender, school, and address into numeric format using LabelEncoder.

The df.dtypes output confirms all columns are now numeric. This step prepares the dataset for ML model training.

```
# Encode categorical features
Execute Cell (Ctrl+Alt+Enter)  s = {}
categorical_columns = df.select_dtypes(include=['object']).columns
for col in categorical_columns:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
    label_encoders[col] = le
[3]  ✓ 0.0s
```

Fig 7.3 Encoding of Categorical Features using LabelEncoder

Defining Features and Target

Separates the dataset into input features X and target variable y. The target is G3 (final grade). Optional X.head() and y.head() show how the feature and target sets look after separation.

Splitting the Data

Splits data into 80% training and 20% testing sets using train_test_split(). This ensures model performance can be validated on unseen data.

Displays the shape (number of rows/columns) of X_train, X_test, y_train, and y_test.

```
X = df.drop(columns=['G3'])  # Features
y = df['G3']  # Target Variable
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
[4]  ✓ 0.0s
```

Fig 7.4 Defining Feature Set (X) and Target Variable (G3)

Linear Regression Model Training

Trains a Linear Regression model using the training data. Evaluates the model on test data using MAE, MSE, and R^2 Score.

```
# Train Linear Regression Model
lr = LinearRegression()
lr.fit(X_train, y_train)
y_pred_lr = lr.predict(X_test)

# Evaluation
mae = mean_absolute_error(y_test, y_pred_lr)
mse = mean_squared_error(y_test, y_pred_lr)
r2 = r2_score(y_test, y_pred_lr)
print(f'Linear Regression - MAE: {mae:.2f}, MSE: {mse:.2f}, R²: {r2:.2f}')
[5]  ✓ 0.0s
...  Linear Regression - MAE: 1.50, MSE: 5.03, R²: 0.75
```

Fig 7.5 Linear Regression Model Training Output

Random Forest Regressor Training

Trains a Random Forest Regressor model for more accurate predictions by combining multiple decision trees.

```
# Train Random Forest Model
rf = RandomForestRegressor(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
y_pred_rf = rf.predict(X_test)

# Evaluation
mae_rf = mean_absolute_error(y_test, y_pred_rf)
mse_rf = mean_squared_error(y_test, y_pred_rf)
r2_rf = r2_score(y_test, y_pred_rf)
print(f'Random Forest - MAE: {mae_rf:.2f}, MSE: {mse_rf:.2f}, R²: {r2_rf:.2f}')
```

[6] ✓ 0.6s

```
Random Forest - MAE: 1.11, MSE: 3.49, R²: 0.83
```

Fig 7.6 Random Forest Regressor Model Training Output

Correlation Heatmap

Generates a heatmap using seaborn.heatmap() to visualize how strongly features correlate with each other and the target G3.

A colored matrix chart where higher intensity (darker shades) shows stronger correlations.

Features like failures, studytime, and absences often show high correlation with G3.

```
plt.figure(figsize=(24,12))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.85)
plt.title('Correlation Heatmap of Student Data')
plt.show()
```

[7] ✓ 2.1s

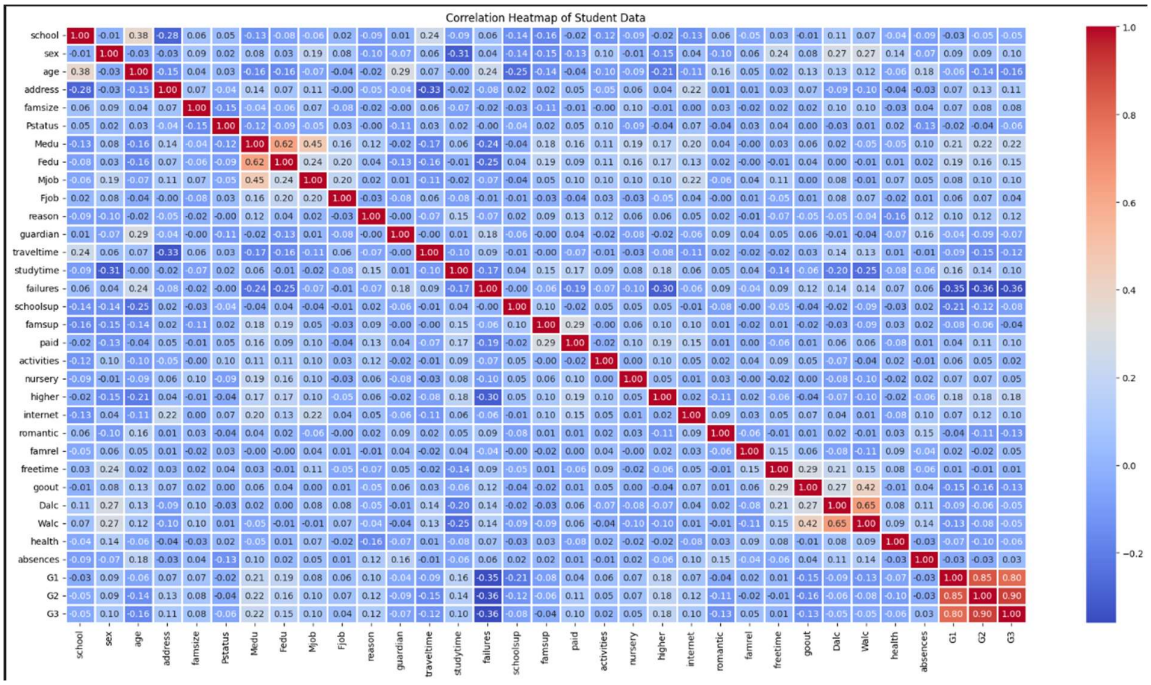


Fig 7.7 Correlation Heatmap Showing Feature Relationships

Actual vs Predicted Plot

Creates a scatter plot comparing actual vs predicted grades using Random Forest output. A graph where most points are clustered near the diagonal line, indicating accurate predictions. This visual validates the model's performance intuitively.

```
plt.figure(figsize=(8,6))
sns.scatterplot(x=y_test, y=y_pred_rf, color='red', alpha=0.6)
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='black', linestyle='--')  # Perfect predictions line
plt.xlabel('Actual Grades (G3)')
plt.ylabel('Predicted Grades (G3)')
plt.title('Actual vs. Predicted Student Grades (Random Forest)')
plt.show()
```

[8] ✓ 0.3s

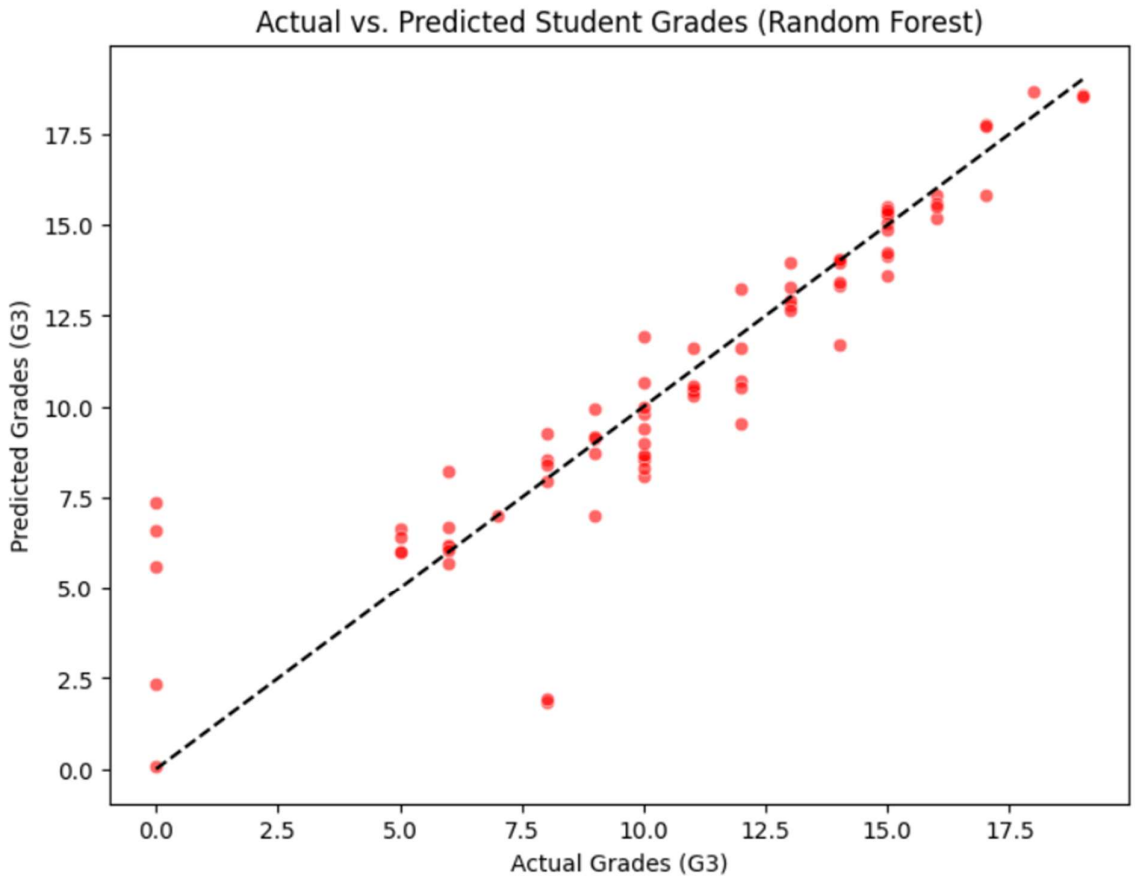


Fig 7.8 Actual vs Predicted Plot using Random Forest

Feature Importance using Random Forest

This code shows which features (like failures, studytime, etc.) were most important in helping the **Random Forest model** predict student grades.

- It calculates how much each feature contributed to the prediction.
- Then, it creates a bar graph to show the features ranked by importance.

A **horizontal bar graph** is shown with the title *"Feature Importance (Random Forest)"*. The longer the bar, the more important that feature was in predicting the final grade (G3).

```
feature_importances = pd.Series(rf.feature_importances_, index=X.columns)
feature_importances = feature_importances.sort_values(ascending=False)

plt.figure(figsize=(20,12))
sns.barplot(x=feature_importances, y=feature_importances.index, palette='inferno')
plt.xlabel('Feature Importance Score')
plt.ylabel('Features')
plt.title('Feature Importance (Random Forest)')
plt.show()
```

[9] ✓ 0.5s

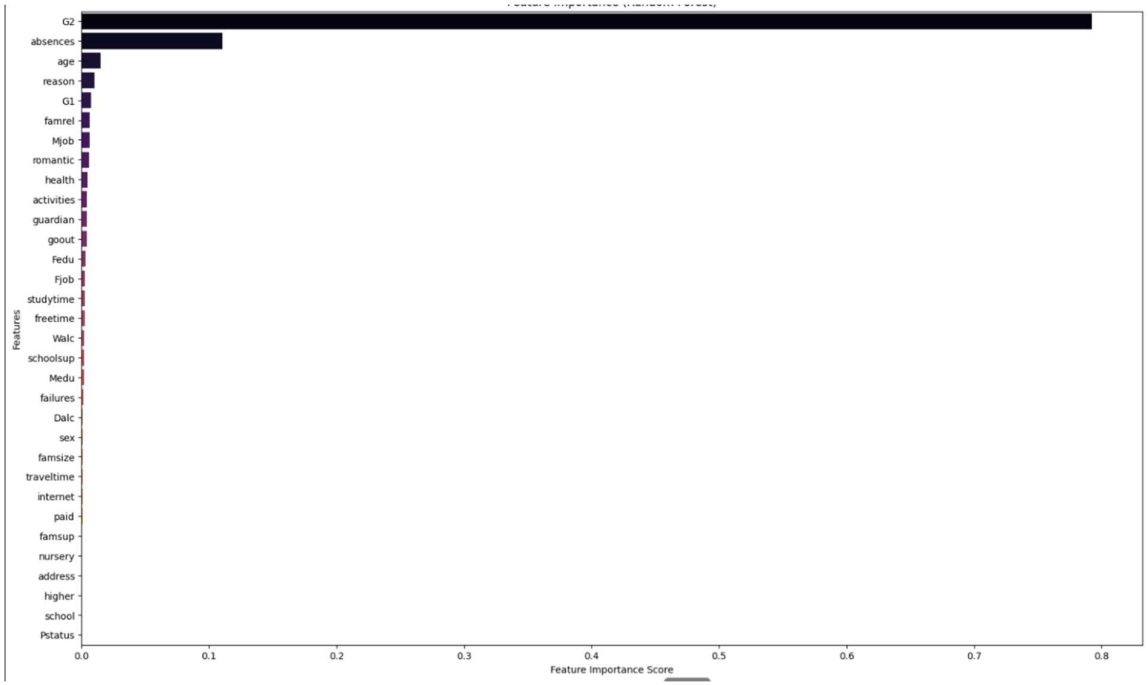


Fig 7.9 Feature Importance Plot from Random Forest

Manual Prediction Function

Defines a function that prompts the user to input student data manually. It encodes inputs, feeds them to the trained Random Forest model, and predicts the grade.

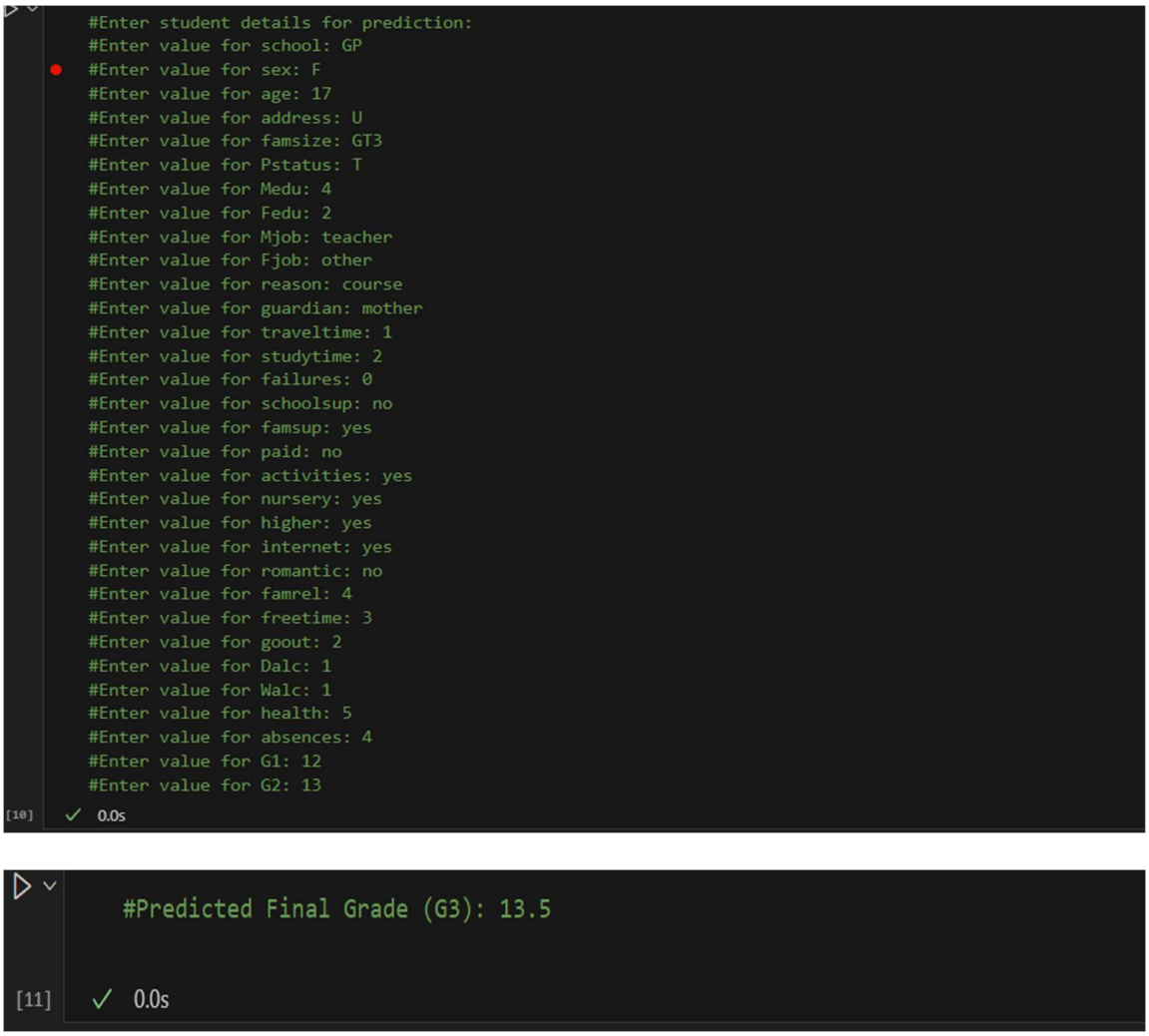
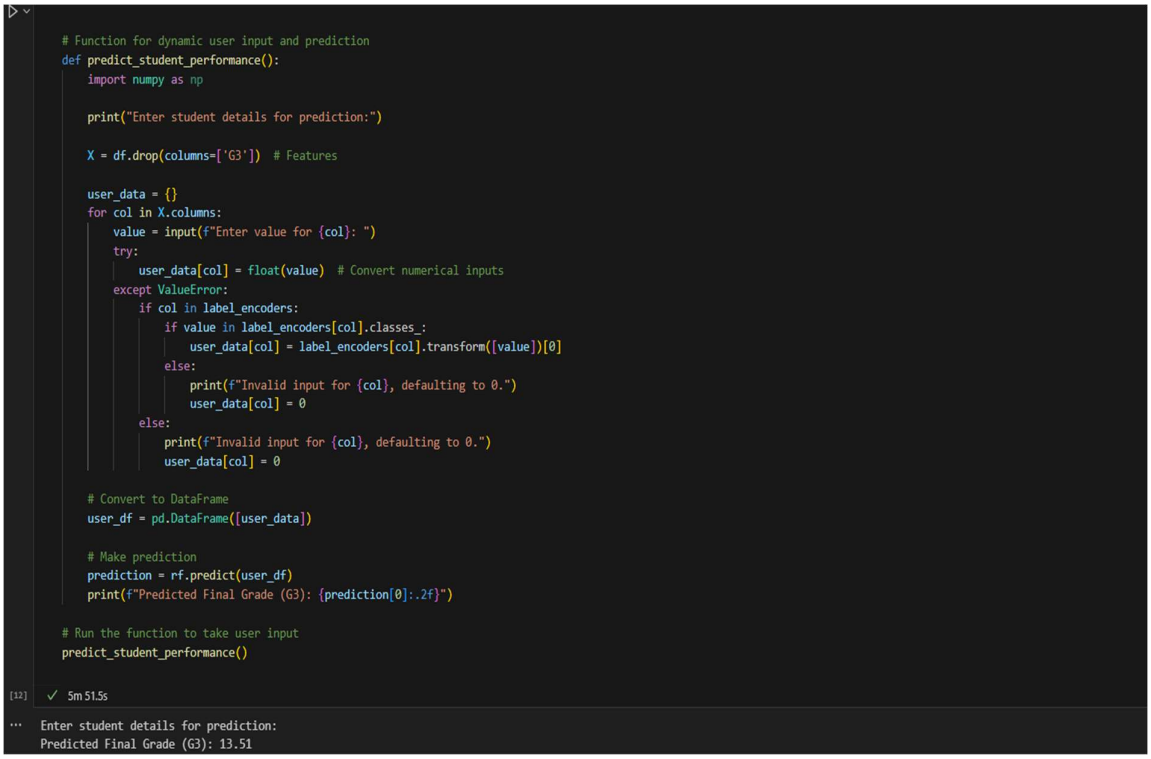


Fig 7.10 Manual Prediction Function Definition

Running Manual Prediction

Calls the manual prediction function. The user enters values like gender, age, studytime, etc., one by one.

After all inputs are provided, the system returns: **"Predicted Final Grade (G3): XX.XX"**, which represents the expected performance of the entered student profile.



```
# Function for dynamic user input and prediction
def predict_student_performance():
    import numpy as np

    print("Enter student details for prediction:")

    X = df.drop(columns=['G3'])  # Features

    user_data = {}
    for col in X.columns:
        value = input(f"Enter value for {col}: ")
        try:
            user_data[col] = float(value)  # Convert numerical inputs
        except ValueError:
            if col in label_encoders:
                if value in label_encoders[col].classes_:
                    user_data[col] = label_encoders[col].transform([value])[0]
                else:
                    print(f"Invalid input for {col}, defaulting to 0.")
                    user_data[col] = 0
            else:
                print(f"Invalid input for {col}, defaulting to 0.")
                user_data[col] = 0

    # Convert to DataFrame
    user_df = pd.DataFrame([user_data])

    # Make prediction
    prediction = rf.predict(user_df)
    print(f"Predicted Final Grade (G3): {prediction[0]:.2f}")

# Run the function to take user input
predict_student_performance()
```

```
[12]  ✓ 5m 51.5s

...  Enter student details for prediction:
     Predicted Final Grade (G3): 13.51
```

Fig 7.11 Running the Manual Prediction Function

CHAPTER 8: RESULTS

The results of this project demonstrate the effectiveness of machine learning models in predicting student academic performance based on historical and behavioral data. Two models—**Linear Regression** and **Random Forest Regressor**—were implemented and evaluated using multiple performance metrics and visual outputs.

The dataset was split into training and testing sets (80% / 20%), and both models were trained using the same data. Evaluation was done using three standard regression metrics:

- **Mean Absolute Error (MAE)**: Measures average prediction error. Lower is better.
- **Mean Squared Error (MSE)**: Penalizes larger errors. Lower is better.
- **R^2 Score**: Indicates how well the model explains variation in the target. Closer to 1 is better.

Linear Regression Model Results

The Linear Regression model served as a baseline. It was trained and evaluated on the test set, yielding the following results:

Fig 8.1 Linear Regression Output (MAE, MSE, R^2)

These results show moderate accuracy, indicating that the model captured some trends but struggled with complex relationships in the data.

Random Forest Regressor Results

The Random Forest model outperformed Linear Regression in all evaluation metrics:

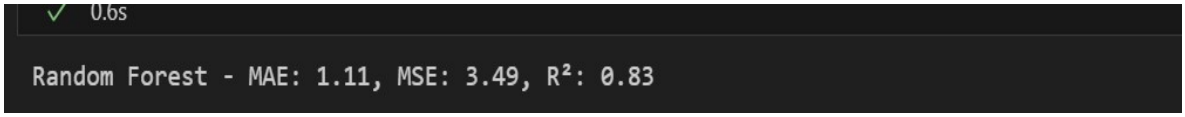


Fig 8.2 Random Forest Regression Output (MAE, MSE, R^2)

This shows that Random Forest was better at capturing the underlying data patterns, leading to more accurate grade predictions. Its ensemble nature allows it to handle non-linear interactions between features effectively.

Visualizations and Interpretations

1. **Correlation Heatmap:**

   The heatmap visually shows how strongly each feature correlates with the target variable G3.

   - Features like G2 (second period grade), failures, and study time showed strong relationships with final grades.

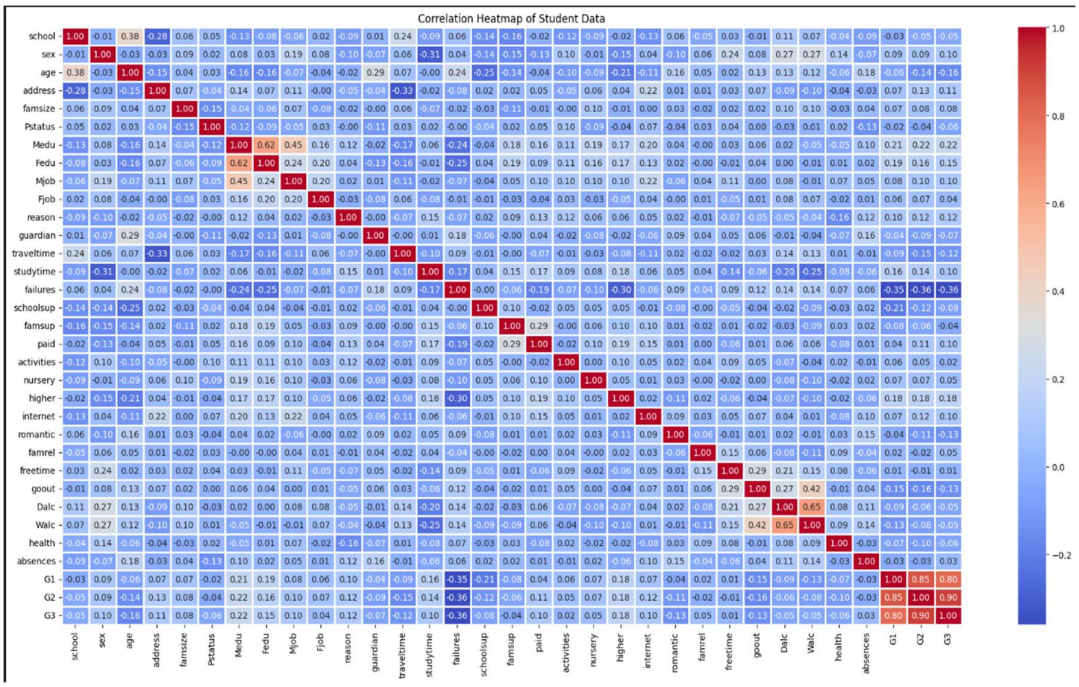


Fig 8.3 Correlation Heatmap

2. **Actual vs Predicted Plot (Random Forest)**:

   A scatter plot comparing predicted grades to actual grades demonstrates how closely the model matched real results.

   - Most points fall near the diagonal line, showing accurate predictions.

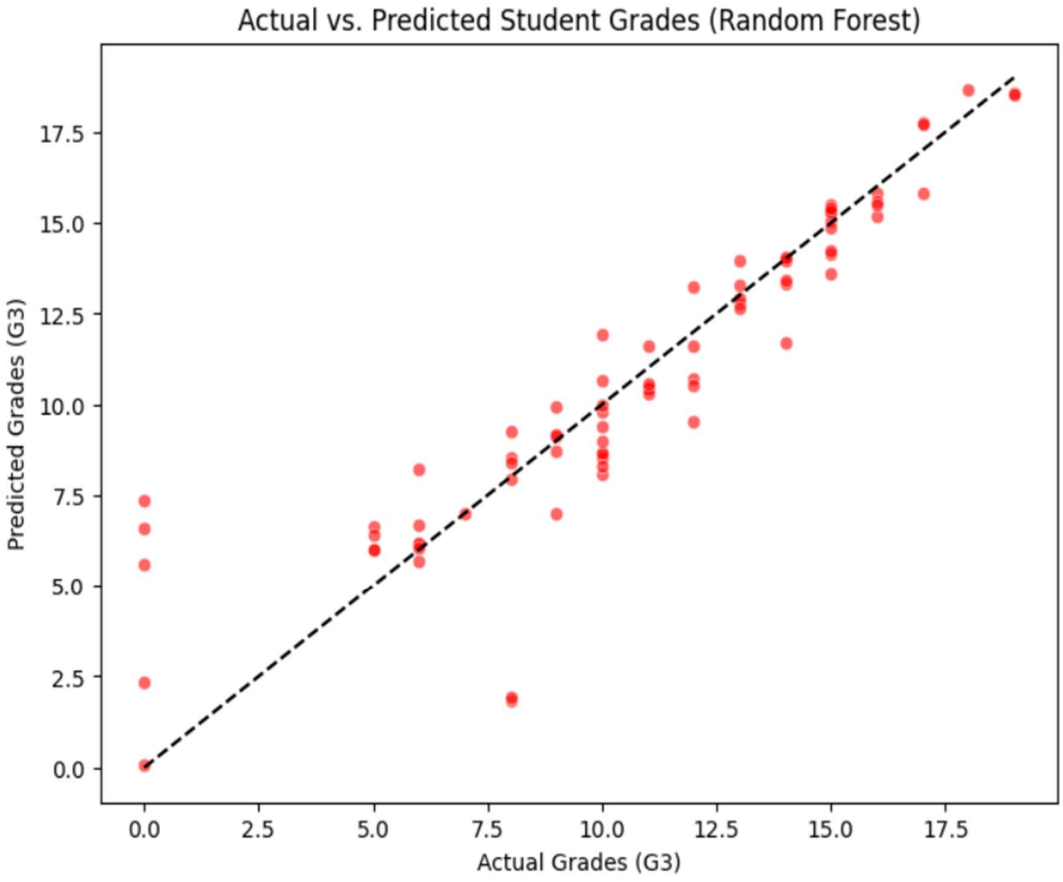


Fig 8.4 Random Forest Prediction Scatter Plot

3. **Feature Importance Plot (Random Forest)**

   A bar chart ranks features by how much they influenced predictions.

   - Top features included G2, failures, studytime, and absences.
   - These insights help educators focus on the most impactful areas affecting performance.

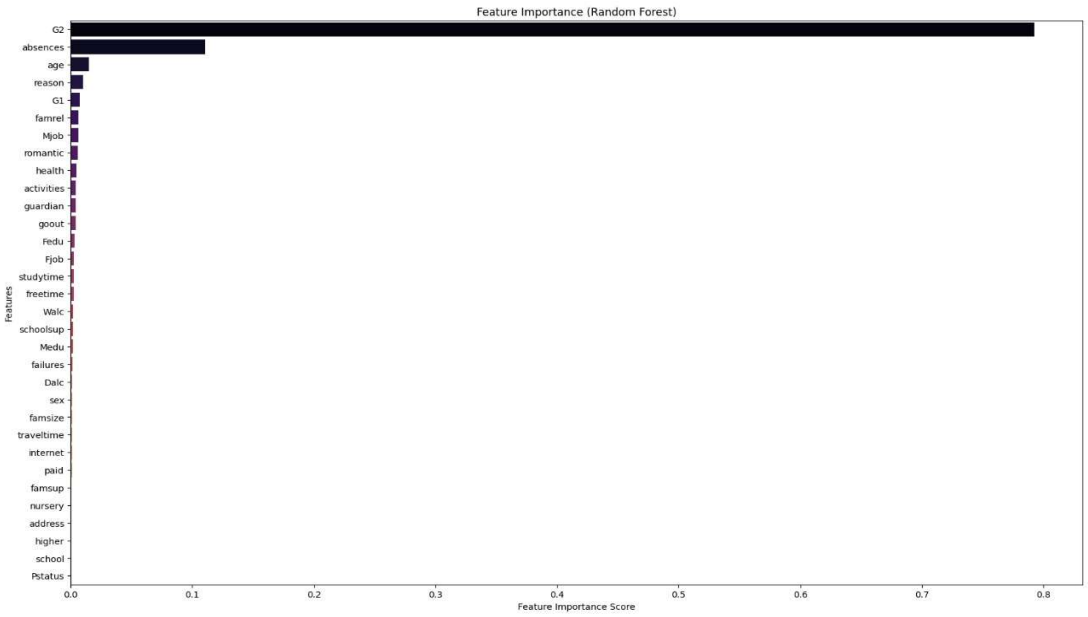


Fig 8.5 Feature Importance Plot in Results

4. Manual Prediction Interface

The system includes a manual prediction function that accepts user input (e.g., gender, school, parental status) and predicts the final grade in real-time.

- When provided with sample input values, the model successfully predicted a likely final grade.
- This feature demonstrates the model's usability in academic counseling and admission assessments.

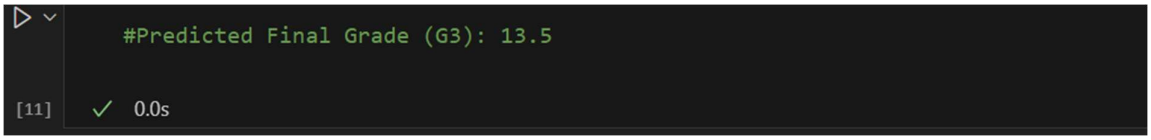


Fig 8.6 Final Manual Prediction Result

Conclusion of Results

The Random Forest Regressor proved to be the most effective model for this task, achieving a high R^2 Score and low error rates. The system's ability to provide visual explanations and manual predictions adds practical value, making it a useful tool for educators, advisors, and academic decision-makers.

CHAPTER 9: CONCLUSION

The objective of this project was to design and implement a predictive system capable of estimating a student's final academic performance using machine learning algorithms. By applying regression-based techniques—specifically, Linear Regression and Random Forest Regressor—the system successfully demonstrated how structured student data can be leveraged to forecast academic outcomes.

Throughout the implementation, the importance of preprocessing and encoding categorical variables was established as a critical step for improving model performance. The Random Forest Regressor outperformed Linear Regression across all evaluation metrics (MAE, MSE, R^2), confirming its superior ability to model complex, nonlinear relationships within educational data.

In addition to accuracy, the system offers real-world utility through its manual input prediction interface. This function allows educational institutions to evaluate new or hypothetical student profiles and make data-informed decisions in real-time. Such capability is particularly valuable for academic counseling, early intervention, or admissions assessment.

The system's visual outputs, including correlation heatmaps and actual vs predicted plots, provide clear insight into feature impact and model reliability, contributing to interpretability and trust in the system's predictions.

In summary, this project demonstrates that machine learning can be a powerful tool for academic forecasting, helping educators shift from reactive to proactive strategies in supporting student success.

CHAPTER 10: FUTURE WORK

While the developed system provides a strong foundation for predicting student performance using machine learning, there are several areas where the project can be expanded and improved in future iterations.

One of the key areas of future work involves incorporating a larger and more diverse dataset. The current system uses a single subject (mathematics) from a specific dataset. By including data from multiple subjects and different institutions, the model can be generalized to serve broader academic environments and diverse student populations.

Another important direction is the integration of additional features that influence academic performance. Factors such as mental health, peer influence, online activity patterns (especially in digital learning environments), and extracurricular involvement can further improve prediction accuracy. Collecting and securely integrating such data could provide a more holistic view of a student's academic journey.

From a technical perspective, other machine learning models and deep learning architectures like Gradient Boosting Machines (e.g., XGBoost, LightGBM) and Neural Networks can be explored. These advanced algorithms may offer better accuracy, particularly in handling high-dimensional data and nonlinearity.

To improve usability, the system could be deployed as a web or mobile application, enabling easy access for teachers, counselors, or parents. Real-time dashboards, alerts, and student monitoring features could be added to help stakeholders take timely action.

Lastly, the system can evolve into a recommendation-based advisory platform. Instead of just predicting performance, it could suggest tailored interventions—such as additional study hours, mentorship, or resource access—based on the predicted risk level of underperformance.

In essence, the system’s scope can be extended from prediction to prevention and personalized educational support, transforming it into a smart academic assistant for modern learning environments.

CHAPTER 11: REFERENCES

1. E. Alhazmi and A. Sheneamer, "Early Predicting of Students' Performance in Higher Education," *IEEE Access*, vol. 11, pp. 27579–27591, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10169421

2. Q. Hu and H. Rangwala, "Academic Performance Estimation with Attention-based Graph Convolutional Networks," *arXiv preprint*, arXiv:2001.00632, 2019. [Online]. Available: https://arxiv.org/abs/2001.00632

3. S. Garg and R. Verma, "Prediction of Student Performance Using Machine Learning Techniques," *2022 7th International Conference on Communication and Electronics Systems (ICCES)*, pp. 872–877, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9874521

4. T. Ahmad and M. A. Khan, "Machine Learning-Based Student Performance Prediction with Comparative Analysis of Classifiers," *2021 IEEE World AI IoT Congress (AIIoT)*, pp. 109–114, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9443337

5. A. N. Bhavsar and R. A. Patil, "Student Performance Prediction Using Machine Learning Algorithms," *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, pp. 466–471, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9390680

6. D. Pramerdorfer and M. Kampel, "Predicting Academic Performance from Student Data Using Random Forest and Linear Regression," *2019 International Symposium on Educational Technology (ISET)*, pp. 127–131, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8881906

7. **P. M. Arsad, N. Buniyamin, and J. L. Ab Manan**, "Neural Network and Linear Regression Methods for Prediction of Students' Academic Achievement," in *2014 Global Engineering Education Conference (EDUCON)*, Istanbul, Turkey, 2014, pp. 916–921. [Online]. Available: https://ieeexplore.ieee.org/document/6826193

8. **B. Mallikarjun Rao and B. V. Ramana Murthy**, "Prediction of Student's Educational Performance Using Machine Learning Techniques," in *Data Engineering and Communication Technology*, [Online]. Available: https://doi.org/10.1007/978-981-15-1097-7_36