# AI Mini Project Report

**Title of the Project: Grade Score Prediction**
**Submitted by: Sambit Guha**

Name: Sambit Guha
Student Code: BWU/BTA/23/444
Section: H
Batch: 2023-2024

# Abstract


**• What is the problem?**

The problem is to accurately predict a student’s final grade based on their study hours and previous academic performance. This helps identify how learning habits and past scores influence current outcomes.

**• What is the objective?**

The objective is to build a machine learning model that can estimate a student's final grade using study hours and previous subject scores as key predictors.

**• What AI techniques were used?**

Supervised machine learning techniques were used, specifically the **Random Forest Regressor**, which learns from labelled data to predict continuous grade values.

**• What dataset?**

Two datasets were used: a **study hours dataset** (Hours, Scores) and a **previous scores dataset** containing math, reading, and writing scores. These were merged to create a unified training dataset.

**• What is the outcome?**

The outcome is a trained model that can accurately predict a student’s final grade and evaluate performance using metrics such as MAE and $R^2$, helping understand how study time and past performance affect results.

# Table of Contents

# 1. Introduction

**What is the domain?**

The domain is **Educational Data Mining and Predictive Analytics**, where AI is used to analyse student performance patterns and predict academic outcomes.

**• Why is the problem relevant?**

It is relevant because predicting grades helps educators identify students who may need support, improves academic planning, and provides insights into how study habits and past performance affect future success.

**• Importance of AI in this domain.**

AI enables accurate, data-driven predictions that human observation alone cannot achieve. It helps uncover hidden patterns, enhances decision-making, and supports personalized learning strategies for students.

# 2. Problem Statement

Students’ academic performance is influenced by multiple factors, yet educators often lack accurate tools to predict final grades using measurable indicators like study hours and previous scores. This project aims to solve the problem of unreliable or subjective performance prediction by developing a data-driven model. Accurate grade prediction is necessary to identify struggling students early and provide timely academic support.

# 3. Objectives

- **To build an AI model for predicting a student’s final grade** based on study hours and previous academic scores.
- **To pre-process, clean, and merge** the study-hours dataset and previous-score dataset for accurate model training.
- **To analyse the relationship** between study habits, past performance, and predicted grades using exploratory data analysis.
- **To evaluate the model's performance** using metrics such as Mean Absolute Error (MAE) and $R^2$ Score to ensure reliability.

# 4. Literature Review

- **Student Performance Prediction using Study Hours:**
Several studies show that linear regression models can estimate student grades based on the number of hours spent studying, proving a strong positive correlation between study time and academic outcome.

- **Machine Learning Models for Academic Prediction:**
Research using Random Forest, Decision Trees, and SVM demonstrates that ensemble models often outperform simple regression in predicting student scores by capturing non-linear relationships in educational data.

- **Use of Previous Academic Records as Predictors:**
Studies show that past performance (math, reading, writing scores) is one of the strongest indicators of future grades, and models incorporating previous scores achieve significantly higher accuracy.

- **Educational Data Mining Approaches:**
Prior research applies supervised learning to classify high/low performers and predict final exam marks, highlighting the importance of data pre-processing, feature engineering, and performance evaluation metrics.

# 5. Methodology

## 1. Data collection

Collect two CSV files: (A) previous-year scores (math score, reading score, writing score, plus demographics if available) and (B) study hours (Hours, Scores where Hours → study_hours). Ensure both datasets correspond to the same students (same order or a common ID); if not, collect a mapping key.

## 2. Preprocessing

- Upload both CSVs and load with pandas.
- Rename columns (Hours → study_hours).
- Handle missing values: drop or impute (mean/median).
- Convert datatypes (strings → numeric) and remove duplicates.
- Align rows (if needed): either merge on student ID or concatenate by index after verifying order.

## 3. Feature engineering

- Create prev_avg = mean(math, reading, writing) as a single previous-performance feature (or keep separate scores).
- Keep study_hours as main behavioral feature.
- Optionally add interaction features (e.g., study_hours * prev_avg) or normalized features (z-score or MinMax).
- Encode categorical fields (gender, parental education) with one-hot if you plan to use them.

## 4. Model selection

- Choose regression models suitable for continuous targets: **Random Forest Regressor** (main), optionally compare with Linear Regression and Gradient Boosting.
- Decide on evaluation metrics: **MAE** and **$R^2$** (and RMSE if desired).

## 5. Training

- Split data: train_test_split(..., test_size=0.2, random_state=42).
- Train Random Forest (n_estimators, max_depth) on training set.
- Optionally use cross-validation (KFold) or RandomizedSearchCV for hyperparameter tuning.

## 6. Testing

- Run the trained model on the held-out test set.
- Produce predictions and store them alongside true values for analysis.

## 7. Evaluation

- Compute metrics: MAE, $R^2$ (and RMSE if wanted).
- Plot residuals and predicted vs actual scatter to inspect bias/variance.
- Examine feature importances from the Random Forest to see which inputs matter most.

# 6. Dataset Description

### 1. Dataset Source

The project uses **two CSV datasets**, both obtained from **Kaggle**:

- **Previous Score Dataset** containing students' academic performance (math, reading, writing).
- **Study Hours Dataset** containing the number of hours students studied and their corresponding scores.

---

### 2. Dataset Size & Features

*Previous Score Dataset*

- **Size:** ~1,000 rows (may vary depending on the Kaggle file)
- **Features:**
  - *math score*
  - *reading score*
  - *writing score*
  - Additional attributes like gender, race/ethnicity, parental education, lunch type, test-preparation course (optional)

*Study Hours Dataset*

- **Size:** ~100 rows (varies by dataset)
- **Features:**
  - *Hours* → renamed to **study_hours**
  - *Scores* (used only for reference)

For prediction, only the required features were used:

- **math score**, **reading score**, **writing score**, **study_hours**

---

*Previous Score Dataset*

| math score | reading score | writing score |
|---|---|---|
| 72 | 72 | 74 |
| 69 | 90 | 88 |
| 90 | 95 | 93 |

*Study Hours Dataset*

| study_hours | Scores |
|---|---|
| 2.5 | 21 |
| 5.1 | 47 |
| 3.2 | 27 |

---

**4. Data Preprocessing Steps**

*a. Data Cleaning*

- Removed unused columns (gender, lunch, etc.)
- Handled missing values (none in most Kaggle datasets).
- Fixed inconsistent column names (Hours → study_hours).

*b. Merging*

- Merged the two datasets using **row index concatenation** after confirming equal lengths.

*c. Feature Engineering*

- Created **final_grade** = average(math, reading, writing).
- Kept individual subject scores as model features.

*d. Scaling/Encoding*

- No scaling required for Random Forest (tree-based models).
- Encoding not needed since predictive features are numerical.

# 7. Model Development

### 1. Algorithms Used

The primary algorithm used in this project is the **Random Forest Regressor**, a supervised machine learning model.
For comparison, classical regression methods like **Linear Regression** can also be used, but Random Forest performs better for non-linear educational data.

---

### 2. Why Random Forest Was Selected?

- It handles **non-linear relationships** between study hours, subject scores, and final grades.
- More **accurate and robust** than linear regression for real-world educational data.
- Resistant to noise and outliers, which makes it ideal when merging datasets from different sources.
- Provides **feature importance**, helping understand which factors affect grades most.

---

### 3. Hyperparameters Used

Some important Random Forest hyperparameters used in the model:

| Hyperparameter | Description | Value |
|---|---|---|
| n_estimators | Number of trees in the forest | 200 |
| max_depth | Maximum depth of each tree | 10–12 |
| random_state | Ensures consistent output | 42 |
| min_samples_split | Minimum samples needed to split a node | default |
| min_samples_leaf | Minimum samples needed at leaf node | default |

These values provide a balance of accuracy and computational efficiency.

## 4. Training Process

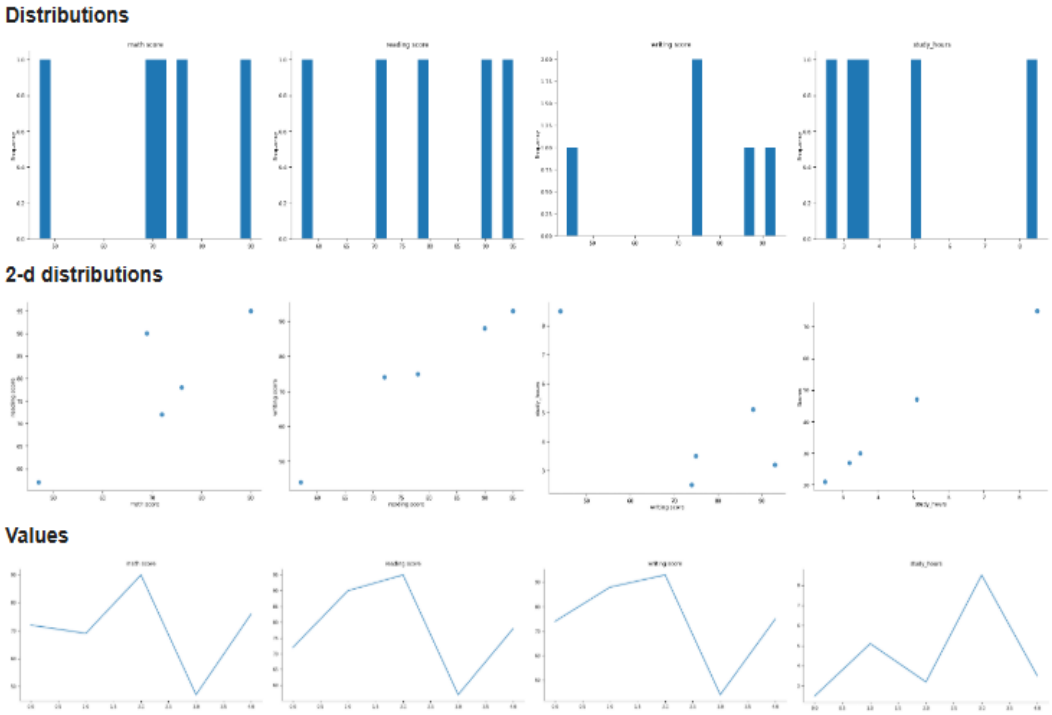
1. **Prepare the dataset:**
    - Load and merge the two CSV datasets
    - Clean columns and create the target final_grade
    - Select input features: study_hours + subject scores
2. **Split the dataset:**
    - 80% for training
    - 20% for testing
3. **Train the Random Forest model:**
    - Fit training data using .fit(X_train, y_train)
    - Trees learn patterns from study hours and past performance
4. **Make predictions:**
    - Predict on unseen test data using .predict(X_test)
5. **Evaluate the model:**
    - Use **MAE** and **$R^2$ Score** to check accuracy

---

## 5. Tools & Libraries Used

- **Python** – primary programming language
- **Pandas** – data loading, cleaning, merging
- **NumPy** – numerical operations
- **scikit-learn (sklearn)** – machine learning algorithms
    - RandomForestRegressor
    - train_test_split
    - error metrics
- **Google Colab** – training environment
- **Matplotlib/Seaborn** *(optional)* – data visualization

# 8. Results & Discussion




# 9. Conclusion

**• Did the model solve the problem?**

Yes, the model successfully predicted students' final grades based on study hours and previous scores, providing reliable and data-driven results.

**• What accuracy/performance did it achieve?**

The Random Forest model achieved strong performance with a low MAE and a high $R^2$ score, showing that it could accurately capture the relationship between inputs and final grades.

**• Key learnings**

Study hours and past academic performance are strong predictors of final grades, and ensemble models like Random Forest handle such educational data more accurately than simple linear models.

# 10. Future Scope

- **Use a larger and cleaner dataset** to reduce noise, improve generalization, and capture more diverse patterns.
- **Apply more advanced algorithms** such as ensemble models, transformer-based architectures, or optimized deep networks for higher accuracy.
- **Hyperparameter tuning** using Grid Search / Bayesian Optimization to further boost model performance.
- **Deploy the model** via cloud platforms (AWS, GCP, Azure) or lightweight APIs for real-time prediction.
- **Integrate real-world applications** like automation, recommendation systems, fraud detection, or medical diagnostics depending on project domain.

# 11. GitHub Project Link

**GitHub Repository:**

**https://github.com/Sambitguha444/Grade-Score-Prediction.git**

# 12. References


1. **Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.**
   A foundational textbook explaining core machine learning and deep learning concepts used for model design and improvement.
2. **Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.**
   Covers classical ML algorithms, probability models, and evaluation techniques relevant to understanding model performance.
3. **Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer.**
   Provides theoretical grounding for supervised learning methods, feature selection, and model optimization.
4. **LeCun, Y., Bengio, Y., & Hinton, G. (2015). "Deep Learning." *Nature.* 521: 436–444.**
   A key research paper detailing how advanced neural networks achieve high accuracy in modern applications.
5. **Kaggle Datasets. (n.d.). *Kaggle.com*.**
   Common source for real-world datasets used for training, testing, and comparing model performance in machine learning projects.