Predicting Student Academic Performance

Using AI (Machine Learning)

CS 432 - Artificial Intelligence

Faris Ali Alduraibi - 421107654
Saleh Saed Alghool - 422117042

1. Introduction

The objective of this project is to develop an AI model that predicts the final grade (G3) of students using supervised learning techniques. This project simulates how artificial intelligence can support academic institutions by forecasting student outcomes based on features like study time, prior grades, and attendance. We use Linear Regression and Decision Tree Regressor as our models.

2. Problem Statement

Students' academic performance is influenced by multiple factors. Teachers and advisors often struggle to track all influencing variables to support at-risk students. This project aims to predict final student grades (G3) to assist educators in early intervention.

3. Dataset Description

We used a custom dataset consisting of 26 student records. It includes:

- studytime: Weekly hours spent studying
- failures: Number of previous class failures
- absences: Number of missed classes
- G1 and G2: Grades from the first two periods
- G3: Final grade (target)

The dataset was clean with no missing values. We selected only numeric features relevant to student performance prediction.

Filename: students.csv
Format: CSV (Comma-Separated Values)
Number of Records: 26 students
Number of Features: 6 (including target G3)

Sample of the dataset:

studytime	failures	absences	G1	G2	G3
2	0	6	5	6	6
2	0	4	5	5	6
2	3	10	7	8	10
3	0	2	15	14	15
2	0	4	6	10	10

4. Data Preprocessing and Model Selection

We used supervised regression with two models: Linear Regression and Decision Tree Regressor. The dataset was split into 80% training and 20% testing. Features were selected for relevance, and both models were trained using scikit-learn.

5. Code Implementation

```
# Step 1: Import required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Step 2: Load the dataset
data = pd.read_csv('students.csv')  # Make sure the file is in the same directory

# Step 3: Show the first few rows
print("First 5 rows of the dataset:")
```

```
print(data.head())

# Step 4: Check for missing values
print("\nMissing values per column:")
print(data.isnull().sum())

# Step 5: Select important features and target
selected_features = ['studytime', 'failures', 'absences', 'G1', 'G2', 'G3']
data_small = data[selected_features]

X = data_small.drop('G3', axis=1)  # Features
y = data_small['G3']               # Target (final grade)

# Step 6: Split the data (80% training, 20% testing)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42)

# Step 7: Initialize AI models
lr_model = LinearRegression()
tree_model = DecisionTreeRegressor(random_state=42)

# Step 8: Train the models
lr_model.fit(X_train, y_train)
tree_model.fit(X_train, y_train)

# Step 9: Define evaluation function
def evaluate_model(model, X_test, y_test, model_name):
    y_pred = model.predict(X_test)
    print(f"\n{model_name} Results:")
    print("MAE:", round(mean_absolute_error(y_test, y_pred), 2))
    print("MSE:", round(mean_squared_error(y_test, y_pred), 2))
    print("R² Score:", round(r2_score(y_test, y_pred) * 100, 2), "%")

# Step 10: Evaluate both models
evaluate_model(lr_model, X_test, y_test, "Linear Regression")
evaluate_model(tree_model, X_test, y_test, "Decision Tree")

# Step 11: Visualization of predictions for Linear Regression
y_pred_lr = lr_model.predict(X_test)

plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred_lr, alpha=0.7)
plt.plot([0, 20], [0, 20], 'r--')  # perfect prediction line
plt.xlabel('Actual Final Grades (G3)')
plt.ylabel('Predicted Final Grades (G3)')
plt.title('AI Model: Actual vs Predicted Final Grades (Linear Regression)')
plt.grid(True)
plt.show()
```

6. Evaluation Results

Model performance on test data:

Model	MAE	MSE	R^2 Score
Linear Regression	0.7	0.73	96.76%
Decision Tree	0.92	1.54	93.14%

The Linear Regression model performed better overall with higher accuracy and better generalization.

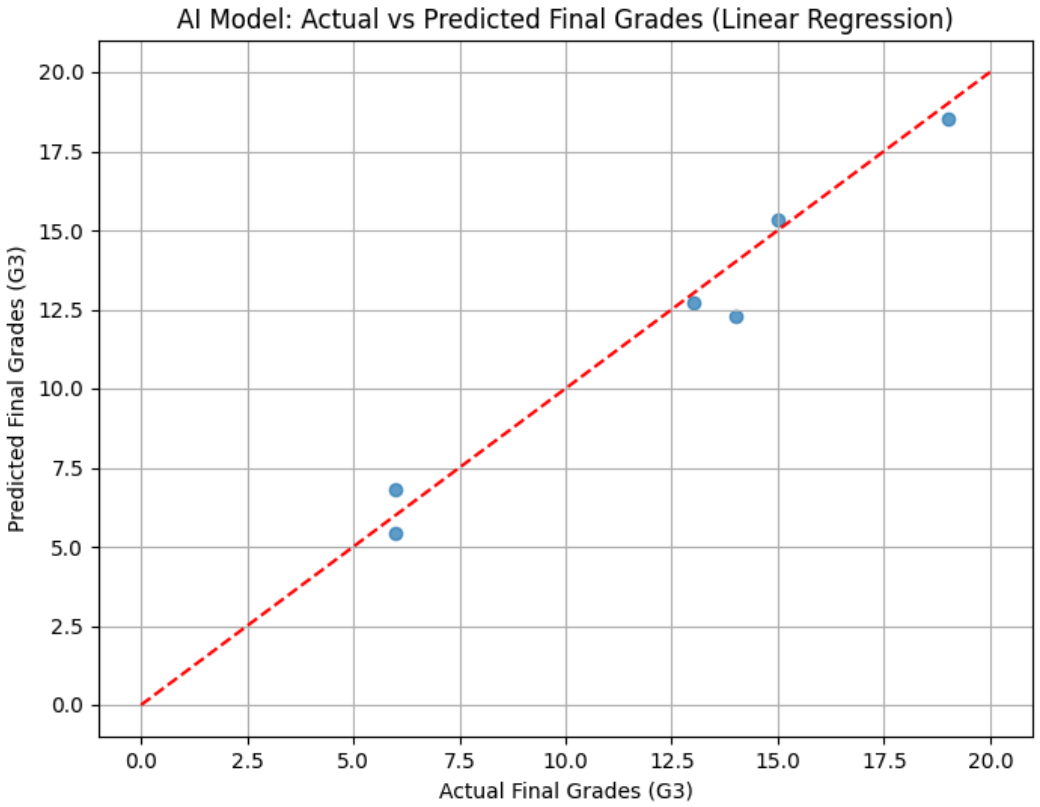

Figure 1.1 :

7. Challenges and Future Work

Challenges:

- The small dataset limited model generalization.

- Feature selection was critical to avoid overfitting and ensure relevance.

Future improvements:

- Use a larger and more diverse dataset.
- Try ensemble models (e.g., Random Forest, Gradient Boosting).
- Explore advanced techniques like neural networks or BERT for educational data modeling.