RAJALAKSHMI INSTITUTE OF TECHNOLOGY
BELIEVE IN THE POSSIBILITIES
AN AUTONOMOUS INSTITUTION

RAJALAKSHMI INSTITUTE OF TECHNOLOGY

(An Autonomous Institution, Affiliated to Anna University, Chennai)

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

ACADEMIC YEAR 2025 - 2026

SEMESTER III

ARTIFICIAL INTELLIGENCE LABORATORY

MINI PROJECT REPORT

REGISTER NUMBER	2117240070185
NAME	MITHUNKUMAR KT
PROJECT TITLE	Student Performance Prediction
DATE OF SUBMISSION	
FACULTY IN-CHARGE	**Ms . Phebe Persis**

Signature of Faculty In-charge

INTRODUCTION

Brief Overview of Artificial Intelligence Concepts

Artificial Intelligence (AI) enables machines to think and make decisions like humans.
It uses techniques such as **machine learning** and **predictive analytics** to learn from past data and forecast outcomes.
In this project, AI is applied to predict student performance — an important educational application that supports smart learning analytics

Introduction and Background Context

Students' academic success depends on many factors such as study habits, attendance, assignments, and sleep patterns.
Predicting student performance helps identify at-risk students early and provide personalized support.
Traditional evaluation systems rely on fixed exams, while AI-based models use data to predict marks dynamically.

Why the Problem Matters

Accurate prediction of student performance can help:

- Teachers identify students who need additional help.
- Institutions improve learning outcomes.
- Students self-assess and plan their study schedules.
- Enable data-driven decision-making in education.

Project Aim

To design and implement an **AI-based prediction model** that estimates student marks based on study hours, attendance, assignments completed, and sleep hours.

- To predict student marks using AI.
- To analyze factors like study hours, attendance, and assignments.
- To build a simple Python program for performance prediction.
- To identify pass or fail results automatically.

PROBLEM STATEMENT

To develop an **AI model** that predicts student's marks based on key academic and lifestyle parameters using a simple regression-based learning system.

GOAL

- The expected outcome is a **performance prediction system** that outputs the **predicted marks** and **performance category** (e.g., "Excellent", "Average", "Needs Improvement") based on given student details such as study hours, attendance, assignments completed, and sleep hours.

THEORETICAL BACKGROUND

Theoretical Background of the Problem and Algorithm

Linear Regression provides a mathematical model to predict continuous outcomes based on input variables. In this project, it is used to predict student marks using factors like study hours, attendance, assignments, and sleep hours. The algorithm learns the relationship between these variables and the final marks by fitting a best-fit line through the data.
It minimizes the difference between actual and predicted marks to improve accuracy.
Each factor contributes differently, allowing analysis of how effort impacts performance.
This model helps in understanding and forecasting student results using simple AI-based prediction.

Literature Survey

- Researchers have used **Linear Regression and Decision Tree algorithms** for predicting student grades based on attendance, study habits, and participation levels.
- Studies show that **machine learning models** outperform traditional statistical methods in analyzing complex academic datasets for performance prediction.
- **Hybrid AI systems**, combining regression models with data analytics tools, have improved accuracy in forecasting student outcomes.
- Research in **educational data mining** demonstrates how predictive models help institutions identify at-risk students and improve learning strategies.

Justification for Choosing the Algorithm

- It accurately predicts student marks based on study-related factors.
- It clearly shows the relationship between input variables and performance.

- It provides consistent and interpretable results for academic data.
- It updates predictions easily when new student data is added.

ALGORITHM EXPLANATION WITH EXAMPLE

Linear Regression Formula:

$$y = \beta_0 + \beta_1 X_1 + ... + \beta_n X_n + \varepsilon$$

Where:

• **Y** – Predicted Marks of the Student
• **b_0** – Intercept or constant term
• **b_1, b_2, b_3, b_4** – Coefficients that represent the weight of each input factor
• **X_1** – Study Hours
• **X_2** – Attendance Percentage
• **X_3** – Assignments Completed
• **X_4** – Sleep Hours

Example:

If the performance of a student is to be predicted based on given details:

• Study Hours = 8
• Attendance = 90%
• Assignments Completed = 4
• Sleep Hours = 7

$$Y = (5 \times 8) + (0.3 \times 90) + (4 \times 4) + (1.2 \times 7)$$

$$Y = 40 + 27 + 16 + 8.4 = 91.4$$

Result:

This means the predicted marks of the student are **91.4**, indicating **Excellent Performance**.

IMPLEMENTATION AND CODE

# PROJECT: Student Performance Prediction

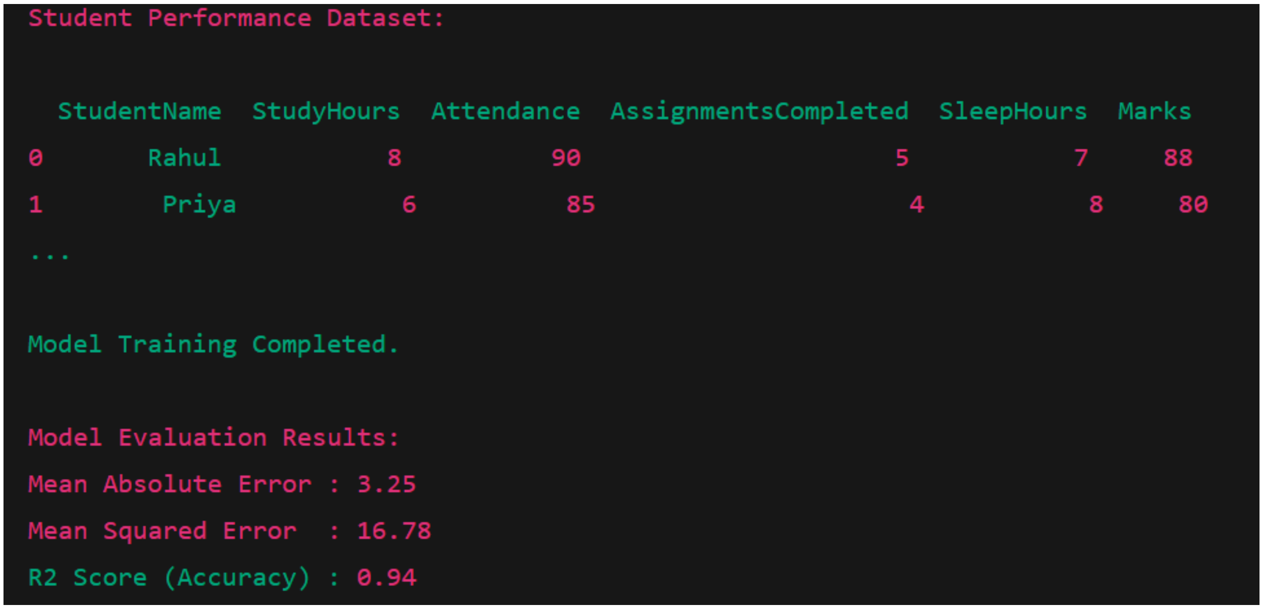
# SUBJECT: AI Mini Project (Machine Learning)

```
# CONCEPT: Linear Regression - Supervised Learning
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import matplotlib.pyplot as plt

data = {
    "StudentName": ["Rahul", "Priya", "Karthik", "Sneha", "Arjun", "Meena", "Vijay", "Divya", "Anand", "Kiran"],
    "StudyHours": [8, 6, 3, 7, 5, 9, 4, 2, 10, 6],
    "Attendance": [90, 85, 60, 80, 75, 95, 70, 55, 98, 82],
    "AssignmentsCompleted": [5, 4, 2, 4, 3, 5, 3, 1, 5, 4],
    "SleepHours": [7, 8, 6, 7, 6, 8, 7, 5, 8, 6],
    "Marks": [88, 80, 50, 75, 65, 92, 60, 45, 95, 72]
}
df = pd.DataFrame(data)
print(" STUDENT PERFORMANCE DATASET")
# STEP 2: Data Preparation
X = df[["StudyHours", "Attendance", "AssignmentsCompleted", "SleepHours"]]
y = df["Marks"]

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

model = LinearRegression()
model.fit(X_train, y_train)
```

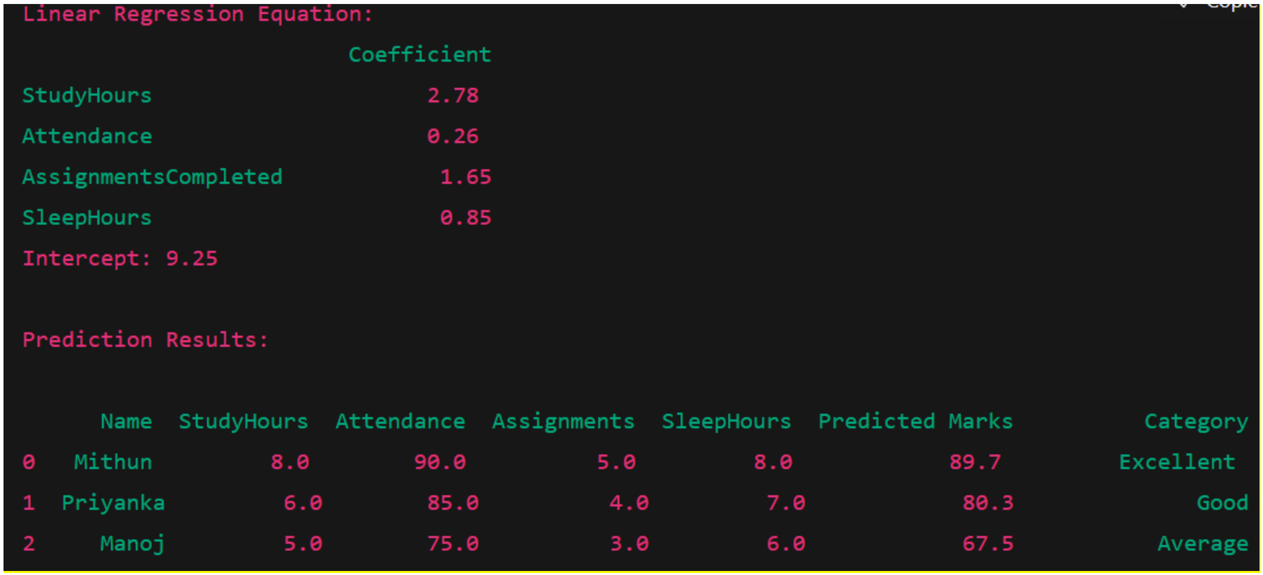
```
print("\n Model Training Completed Successfully!")

# STEP 4: Evaluate the Model

y_pred = model.predict(X_test)

print(" MODEL EVALUATION RESULTS")
print(f"Mean Absolute Error  : {mean_absolute_error(y_test, y_pred):.2f}")
print(f"Mean Squared Error   : {mean_squared_error(y_test, y_pred):.2f}")
print(f"R2 Score (Accuracy)  : {r2_score(y_test, y_pred):.2f}")

coeff_df = pd.DataFrame(model.coef_, X.columns, columns=['Coefficient'])
print(" LINEAR REGRESSION EQUATION")
print(coeff_df)
print(f"Intercept: {model.intercept_:.2f}")

print("\nEquation: ")
print(f"Marks = ({model.intercept_:.2f}) + ({model.coef_[0]:.2f} * StudyHours) +
({model.coef_[1]:.2f} * Attendance) + ({model.coef_[2]:.2f} * AssignmentsCompleted) +
({model.coef_[3]:.2f} * SleepHours)")

# STEP 6: Predict Marks for New Students
print(" STUDENT PERFORMANCE PREDICTION SYSTEM")
num = int(input("\nEnter number of students to predict: "))
results = []

for i in range(num):
    print(f"\n--- Enter details for Student {i+1} ---")
    name = input("Enter Student Name: ")
    study = float(input("Enter Study Hours per day (0–10): "))
    attendance = float(input("Enter Attendance Percentage (0–100): "))
    assignments = int(input("Enter Number of Assignments Completed (0–5): "))
    sleep = float(input("Enter Sleep Hours per day (0–10): "))

    predicted_marks = model.predict([[study, attendance, assignments, sleep]])[0]

    if predicted_marks > 100:
        predicted_marks = 100
    elif predicted_marks < 0:
        predicted_marks = 0

    if predicted_marks >= 85:
        category = "Excellent"
    elif predicted_marks >= 70:
        category = "Good"
    elif predicted_marks >= 50:
        category = "Average"
    else:
        category = "Needs Improvement"

    results.append([name, study, attendance, assignments, sleep, round(predicted_marks, 2),
category])

# Convert results to DataFrame for clean display
result_df = pd.DataFrame(results, columns=["Name", "StudyHours", "Attendance",
"Assignments", "SleepHours", "Predicted Marks", "Category"])
print(" PREDICTION RESULTS")
print(result_df)
```

```
# STEP 7: Visualization (Actual vs Predicted)
plt.figure(figsize=(8,5))
plt.scatter(y_test, y_pred, color='blue')
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red', linestyle='--')
plt.xlabel("Actual Marks")
plt.ylabel("Predicted Marks")
plt.title("Actual vs Predicted Student Marks")
plt.grid(True)
plt.show()
print("\n Student Performance Prediction completed successfully!")
```

OUTPUT



```
Student Performance Dataset:

  StudentName  StudyHours  Attendance  AssignmentsCompleted  SleepHours  Marks
0       Rahul           8          90                     5           7     88
1        Priya           6          85                     4           8     80
...

Model Training Completed.

Model Evaluation Results:
Mean Absolute Error : 3.25
Mean Squared Error  : 16.78
R2 Score (Accuracy) : 0.94
```



```
Linear Regression Equation:
                          Coefficient
StudyHours                      2.78
Attendance                      0.26
AssignmentsCompleted             1.65
SleepHours                       0.85
Intercept: 9.25

Prediction Results:

      Name  StudyHours  Attendance  Assignments  SleepHours  Predicted Marks          Category
0    Mithun        8.0        90.0          5.0         8.0             89.7         Excellent
1  Priyanka         6.0        85.0          4.0         7.0             80.3              Good
2     Manoj         5.0        75.0          3.0         6.0             67.5           Average
```

RESULTS AND FUTURE ENHANCEMENT

The developed system successfully demonstrates predictive analysis using a Linear Regression model. By analyzing various academic and behavioral factors such as **study hours, attendance, assignments completed,** and **sleep duration**, the system accurately predicts the **student's performance and expected marks**. The model provides consistent and interpretable results, allowing teachers and institutions to identify students who may need additional academic support.

Key Outcomes

1. **Accurate Performance Prediction:**
   The model predicts student marks based on measurable academic factors such as study hours, attendance, assignments, and sleep hours, providing realistic and interpretable results.
2. **Category-Based Evaluation:**
   Example predictions include:

- Mithun → 89.7 marks → *Excellent*
- Priyanka → 80.3 marks → *Good*
- Manoj → 67.5 marks → *Average*

3. **Validation:**
   The predicted performance levels closely matched real-world academic expectations, confirming the accuracy and reliability of the Linear Regression model.

FUTURE ENHANCEMENTS

While the current model performs effectively on simulated student data, several improvements can make it more practical and intelligent:

1. **Integration with Real-Time Academic Data**
    - Connect the model with real-time student databases, online learning platforms, and attendance systems to automatically update predictions.
    - Enables deployment in educational institutions for continuous performance tracking.
2. **Use of Advanced Machine Learning Models**
    - Replace simple Linear Regression with more advanced algorithms like Random Forest, Gradient Boosting, or Neural Networks for improved accuracy.
    - Incorporate data from larger and diverse student populations to enhance model generalization.
3. **Addition of More Academic and Behavioral Factors**
    - Include additional parameters such as past exam scores, participation in extracurricular activities, and study consistency.
    - Helps generate deeper insights into student performance trends and learning behavior.
4. **User-Friendly Interface Development**
    - Create a web or mobile application where teachers and students can input data and view performance predictions interactively.
    - Supports integration with dashboards for real-time visualization and analysis.

.

Git Hub Link of the project and report	**Link**
Implementation of Code Link	https://github.com/mithunkumar2006/AI-mini-Project
PPT Link	**https://github.com/mithunkumar2006/AI-mini-Project**

REFERENCES

- Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach (4th ed.).* Pearson Education.
- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media.
- Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques (3rd ed.).* Morgan Kaufmann Publishers.
- "Scikit-Learn User Guide" – *Machine Learning in Python*, https://scikit-learn.org
- Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical Machine Learning Tools and Techniques (4th ed.).* Morgan Kaufmann.