A PROJECT REPORT

ON

"Student Performance Prediction"

Session: 2025-2026

DEPARTMENT OF INFORMATION TECHNOLOGY
HALDIA INSTITUTE OF TECHNOLOGY

Submitted To:

Mr. Debasish Sahoo

Submitted By:

Sakshee - 10300223134
Saloni Gupta - 10300223135
Surbhi Patel - 10300223177
Tripti Kumari - 10300223183

1. Executive Summary

This project develops a Student Performance Prediction System using various Machine Learning algorithms to predict students' Exam Scores based on academic, personal, and environmental factors.

Multiple regression models were tested including:

Linear Regression

Ridge Regression

Lasso Regression

Support Vector Regression (SVR)

Decision Tree Regressor

Random Forest Regressor

K-Nearest Neighbors (KNN)

2.INTRODUCTION

In today's education system, student performance is very important for academic success. However, exam results do not depend only on intelligence. Many other factors like parental involvement, motivation level, teacher quality, access to resources, family income, and peer influence also affect a student's performance.

It is difficult for schools and teachers to manually analyze all these factors for every student. Therefore, this project uses Machine Learning techniques to predict student exam scores based on different influencing factors.

3.PROBLEM STATEMENT

The objective of this project is to develop a system that can calculate and analyze a student's academic performance in terms of percentage based on marks obtained in different subjects. The system should accept subject-wise marks as input, compute the overall percentage, and provide an accurate result.

5.DATASET DESCRIPTION

The dataset contains student academic and socio-economic factors.

Feature	Type	Description
Hours_Studied	Numerical	Study hours per week
Attendance	Numerical	Attendance percentage
Parental_Involvement	Ordinal	Low, Medium, High
Access_to_Resources	Ordinal	Educational resources availability
Motivation_Level	Ordinal	Student motivation
Family_Income	Ordinal	Economic background
Teacher_Quality	Ordinal	Teaching effectiveness
Peer_Influence	Ordinal	Social influence
Internet_Access	Binary	Yes / No
Previous_Scores	Numerical	Past academic performance
Exam_Score	Numerical (Target)	Final exam score

```
df = pd.DataFrame(data)
```

```
df
```

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sleep_Hours	Previous_Scores	Moti
	23	84	Low	High	No	7	73	
	19	64	Low	Medium	No	8	59	
	24	98	Medium	Medium	Yes	7	91	
	29	89	Low	Medium	Yes	8	98	
	19	92	Medium	Medium	Yes	6	65	
	...	...	...	...	...	...	...	
	25	69	High	Medium	No	7	76	
	23	76	High	Medium	No	8	81	
	20	90	Medium	Low	Yes	6	65	
	10	86	High	High	Yes	6	91	
	15	67	Medium	Low	Yes	9	94	

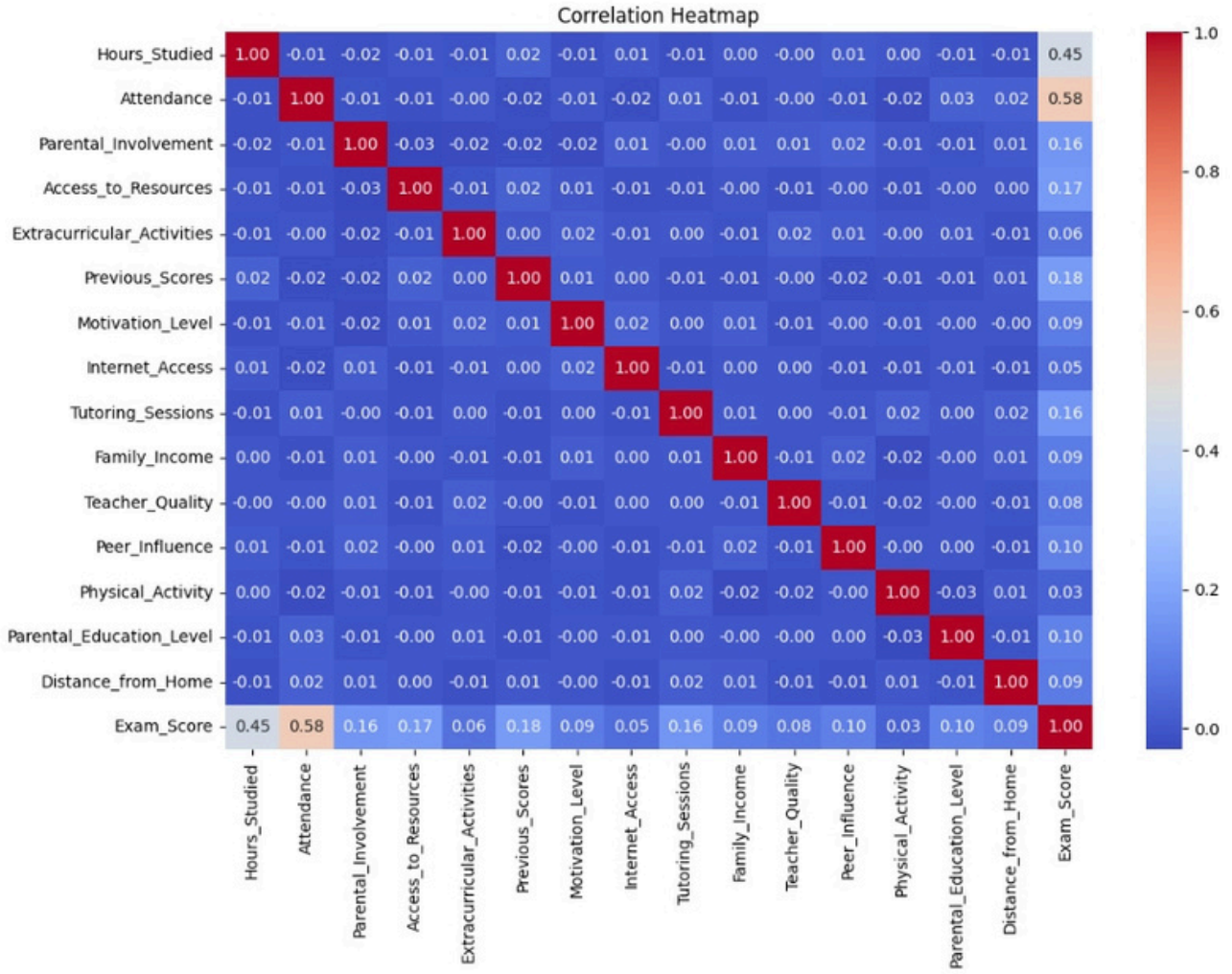
rows × 20 columns

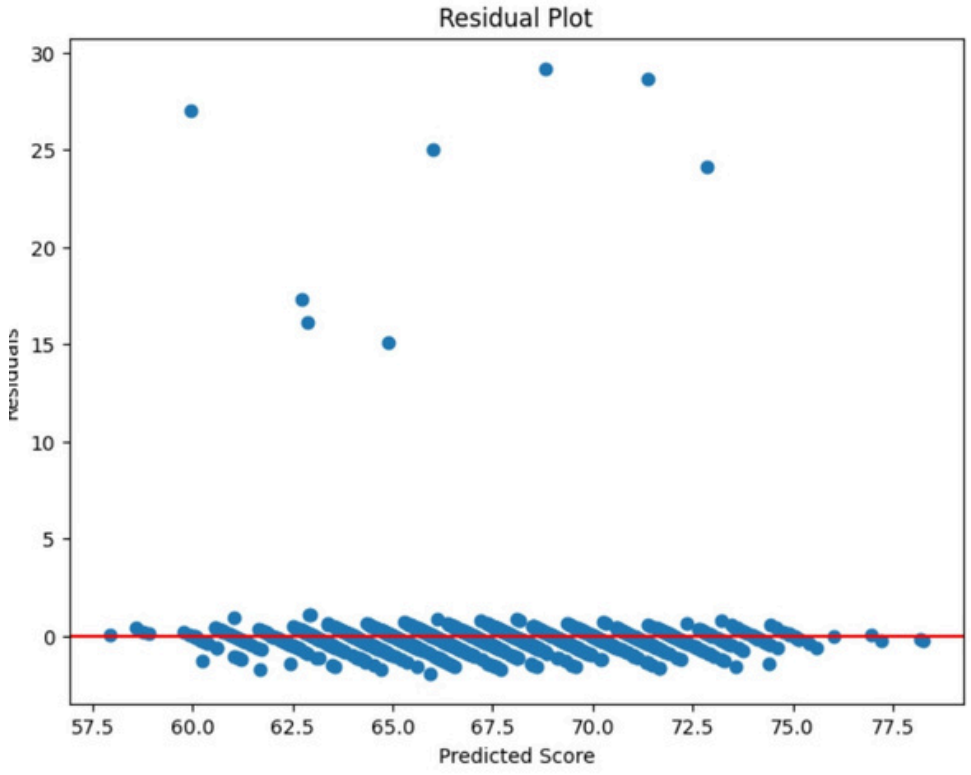
7.DATA PREPROCESSING

Missing values replaced using mean/mode
Categorical mapping to numerical values
Feature scaling using StandardScaler
Train-test split (80% training, 20% testing)
Model Selection
Models used:
Model Purpose
Linear Regression Baseline model
Ridge & Lasso Regularization
SVR Non-linear regression
Decision Tree Rule-based learning
Random Forest Ensemble learning
KNN Distance-based regression
Pipeline used:

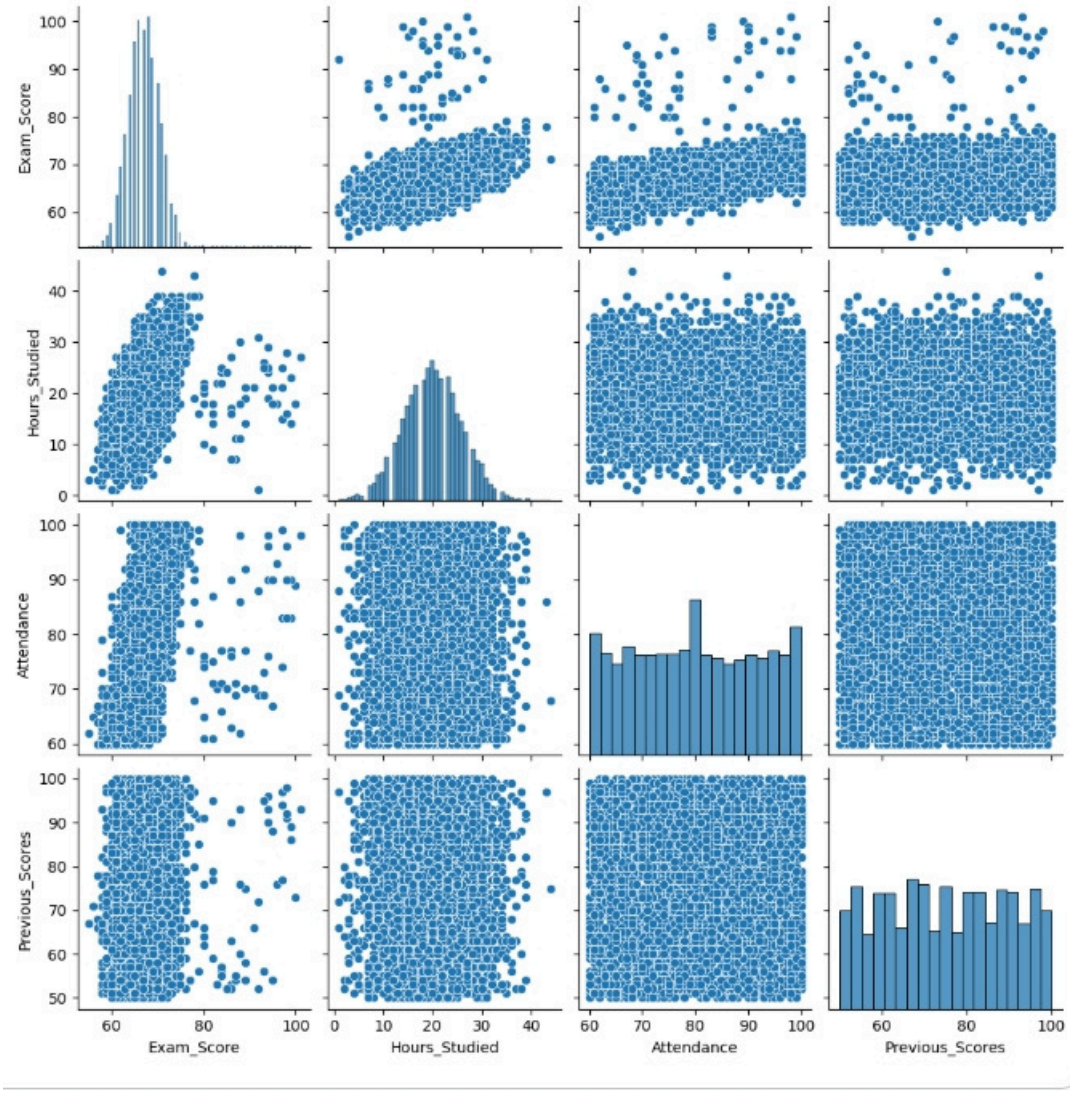
```
Pipeline([
  ('scaler', StandardScaler()),
  ('model', ModelName)
])
```

Model Evaluation Metrics
Mean Squared Error (MSE)
Root Mean Squared Error (RMSE)
R^2 Score
Best model selected based on highest R^2 score.


Correlation Heatmap
Hours_Studied 1.00 -0.01 -0.02 -0.01 -0.01 0.02 -0.01 0.01 -0.01 0.00 -0.00 0.01 0.00 -0.01 -0.01 0.45
Attendance -0.01 1.00 -0.01 -0.01 -0.00 -0.02 -0.01 -0.02 0.01 -0.01 -0.00 -0.01 -0.02 0.03 0.02 0.58
Parental_Involvement -0.02 -0.01 1.00 -0.03 -0.02 -0.02 -0.02 0.01 -0.00 0.01 0.01 0.02 -0.01 -0.01 0.01 0.16
Access_to_Resources -0.01 -0.01 -0.03 1.00 -0.01 0.02 0.01 -0.01 -0.01 -0.00 -0.01 -0.00 -0.01 -0.00 0.00 0.17
Extracurricular_Activities -0.01 -0.00 -0.02 -0.01 1.00 0.00 0.02 -0.01 0.00 -0.01 0.02 0.01 -0.00 0.01 -0.01 0.06
Previous_Scores 0.02 -0.02 -0.02 0.02 0.00 1.00 0.01 0.00 -0.01 -0.01 -0.00 -0.02 -0.01 -0.01 0.01 0.18
Motivation_Level -0.01 -0.01 -0.02 0.01 0.02 0.01 1.00 0.02 0.00 0.01 -0.01 -0.00 -0.01 -0.00 -0.00 0.09
Internet_Access 0.01 -0.02 0.01 -0.01 -0.01 0.00 0.02 1.00 -0.01 0.00 0.00 -0.01 -0.01 -0.01 -0.01 0.05
Tutoring_Sessions -0.01 0.01 -0.00 -0.01 0.00 -0.01 0.00 -0.01 1.00 0.01 0.00 -0.01 0.02 0.00 0.02 0.16
Family_Income 0.00 -0.01 0.01 -0.00 -0.01 -0.01 0.01 0.00 0.01 1.00 -0.01 0.02 -0.02 -0.00 0.01 0.09
Teacher_Quality -0.00 -0.00 0.01 -0.01 0.02 -0.00 -0.01 0.00 0.00 -0.01 1.00 -0.01 -0.02 -0.00 -0.01 0.08
Peer_Influence 0.01 -0.01 0.02 -0.00 0.01 -0.02 -0.00 -0.01 -0.01 0.02 -0.01 1.00 -0.00 0.00 -0.01 0.10
Physical_Activity 0.00 -0.02 -0.01 -0.01 -0.00 -0.01 -0.01 -0.01 0.02 -0.02 -0.02 -0.00 1.00 -0.03 0.01 0.03
Parental_Education_Level -0.01 0.03 -0.01 -0.00 0.01 -0.01 -0.00 -0.01 0.00 -0.00 -0.00 0.00 -0.03 1.00 -0.01 0.10
Distance_from_Home -0.01 0.02 0.01 0.00 -0.01 0.01 -0.00 -0.01 0.02 0.01 -0.01 -0.01 0.01 -0.01 1.00 0.09
Exam_Score 0.45 0.58 0.16 0.17 0.06 0.18 0.09 0.05 0.16 0.09 0.08 0.10 0.03 0.10 0.09 1.00
Hours_Studied
Attendance
Parental_Involvement
Access_to_Resources
Extracurricular_Activities
Previous_Scores
Motivation_Level
Internet_Access
Tutoring_Sessions
Family_Income
Teacher_Quality
Peer_Influence
Physical_Activity
Parental_Education_Level
Distance_from_Home
Exam_Score
1.0
0.8
0.6
0.4
0.2
0.0


Residual Plot
Residuals
30
25
20
15
10
5
0
57.5
60.0
62.5
65.0
67.5
70.0
72.5
75.0
77.5
Predicted Score

PAIRPLOT



9.MODEL TRAINING

The cleaned dataset was split into:

80% training data

20% testing data

The Logistic Regression model was trained using the training dataset. The model learned the relationship between student attributes and performance outcome.

[65] ✓ 0s

```python
# Train-Test Split
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(
    x, y, test_size=0.2, random_state=51
)
# Define Models
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.metrics import root_mean_squared_error
```

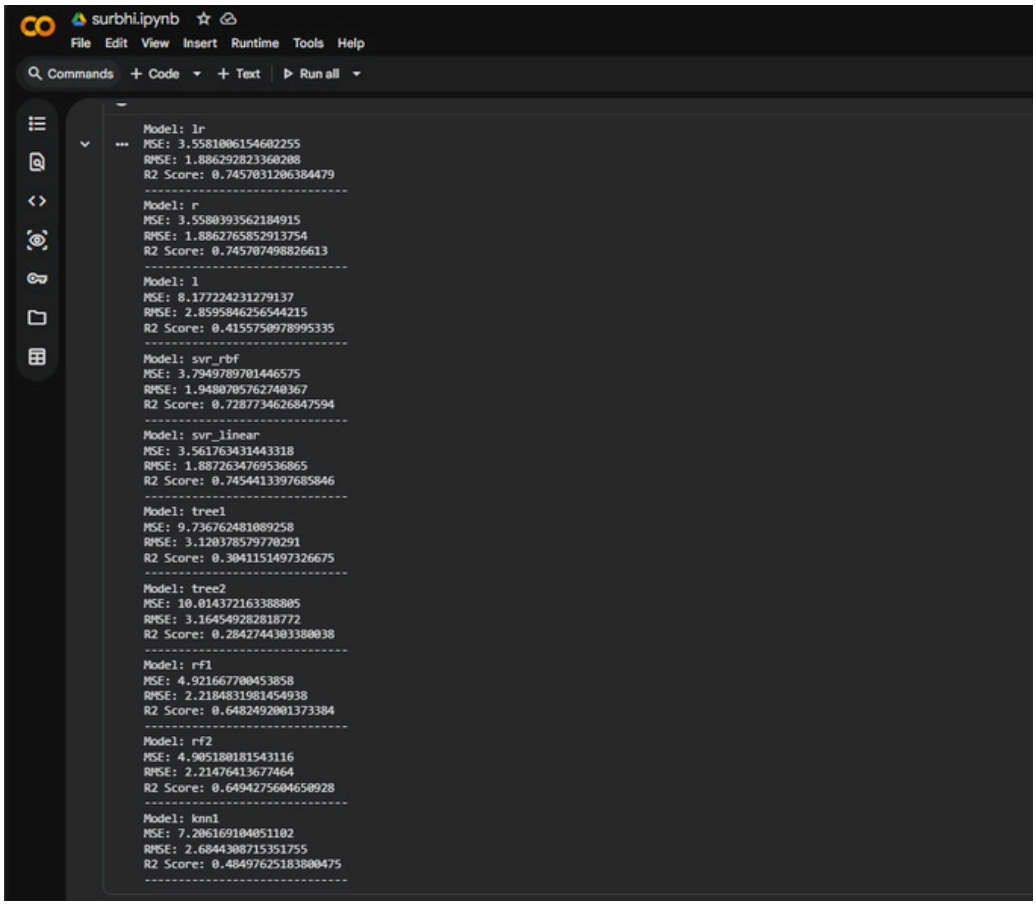
[66] ✓ 0s

```python
models = {
    'lr': LinearRegression(),
    'r': Ridge(),
    'l': Lasso(),
    'svr_rbf': SVR(kernel='rbf'),
    'svr_linear': SVR(kernel='linear'),
    'tree1': DecisionTreeRegressor(random_state=51),
    'tree2': DecisionTreeRegressor(random_state=21),
    'rf1': RandomForestRegressor(n_estimators=200, random_state=51),
    'rf2': RandomForestRegressor(n_estimators=100, random_state=20),
    'knn1': KNeighborsRegressor(n_neighbors=3)
}
```

Train and Evaluate Models

[67]

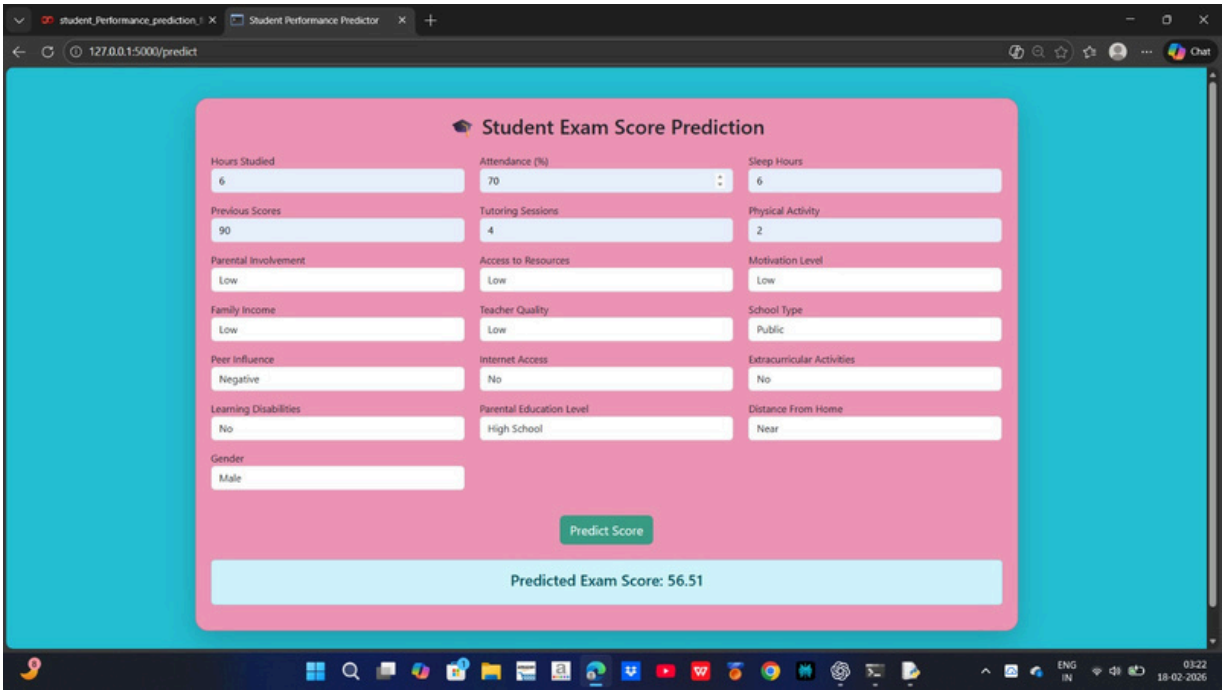
```python
results = {}

for name, model in models.items():

    pipeline = Pipeline([
        ('scaler', StandardScaler()),
        ('model', model)
    ])

    pipeline.fit(x_train, y_train)

    y_pred = pipeline.predict(x_test)

    mse = mean_squared_error(y_test, y_pred)
    rmse = root_mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)

    results[name] = {
        'model': pipeline,
        'mse': mse,
        'rmse': rmse,
        'r2': r2
    }

    print(f"Model: {name}")
    print(f"MSE: {mse}")
    print(f"RMSE: {rmse}")
    print(f"R2 Score: {r2}")
    print("-" * 30)
```



11.RESULT ANALYSIS

The Student Exam Score Prediction system was successfully developed and tested using a machine learning model trained on student-related academic and behavioral features. The system accepts multiple input parameters such as study hours, attendance percentage, previous scores, sleep hours, tutoring sessions, parental involvement, motivation level, and other socio-academic factors.

After training and testing the model, it was able to generate predicted exam scores with reasonable accuracy. When sample input values were provided through the web interface, the system produced a predicted score of 56.51, demonstrating that the model is functioning correctly and responding to input variations.



13.CONCLUSION

This project demonstrates the practical application of Machine Learning in the education sector for predicting student performance. The developed system can assist educators and institutions in identifying students who may require additional academic support before final examinations. Early prediction enables timely intervention, which can contribute to improved academic outcomes.

The analysis indicates that factors such as study hours, attendance, previous academic performance, and motivation level significantly influence exam scores. By integrating the machine learning model with a web interface, the system becomes accessible and usable even by non-technical users.

In the future, the system can be enhanced by:

Using larger and real-world datasets

Applying advanced machine learning algorithms

Incorporating feature importance visualization

Adding interactive dashboards for performance analysis

Overall, this project confirms that Machine Learning is a powerful and effective tool for predictive analysis in educational environments.

15.TOOLS & TECHNOLOGIES USED

Programming Language: Python

Libraries: Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn

Platform: Jupyter Notebook

16. REFERENCES

Scikit-learn Documentation

Pandas Documentation

Machine Learning textbooks and online resources