Machine Learning

MiniProjectReport

J.Abdul karim

71052302003

Project Report

Student Performance Prediction Using Machine Learning.

1. Introduction

Studentperformance prediction is an important educational data mining task that helps schools and teachers identify weak learners in advance. By analyzing various academic and socio-economic factors, machine learning models can predict a student's final exam performance. This helps institutions take preventive measures, provide personalized coaching, and improve overall academic outcomes.

This project uses machine learning techniques to predict whether a student will perform Poor, Average, or Good based on various attributes.

2. Problem Statement

Thegoal of thismini-project isto develop a machine learning model that predicts student academic performance from given input features such as study hours, attendance, socio- economic factors, and past scores.

The target variable: Performance Category (Poor / Average / Good).

3. Dataset Description

Thedatasetcontainsstudentacademicand behavioral attributes.

Attributes

• Study Hours per Day
• Attendance (%)
• Past Exam Score
• Family Income
• Parental Education Level
• Extra Classes (Yes/No)
• Health Status (1–5 rating)
• Final Performance Category (Target)

Data Preprocessing

•Missing values filled using mean/mode
•Categorical columns encoded
•Outliers removed using IQR
•Converted numerical features to consistent scales.

4. Methodology

4.1 Data Cleaning

• Filled missing values in income, past scores, and attendance
• Standardized categorical labels (e.g., "yes"/"Yes")
• Removed inconsistent entries (e.g., attendance > 100%)

4.2 Feature Engineering

• Created **Study_Effectiveness=StudyHours×Attendance**
• Converted parental education into ordinal scale
• Binary encoded Extra Classes (1 / 0)

4.3 Feature Selection

Selected predictive features:

- StudyHours
- Attendance
- PastScore
- ParentalEducation
- ExtraClasses
- Health
- Study_Effectiveness

Target variable: **Performance Category**

4.4 Feature Scaling

Used **StandardScaler** for numerical features to improve model performance.

4.5 Train-Test Split

• 80% Training
• 20% Testing
• Stratified split to maintain class balance

4.6 Model Selection

A **RandomForestClassifier**was chosen because:

• Handles non-linear relationships
• High accuracy for classification problems
• Works well with mixed data types
• Reduces overfitting

5. Model Development

A RandomForest modelwastrained with:

• n_estimators = 300
• max_depth = 10
• min_samples_split = 3

This model learns the relationship between student academic behavior and performance level.

6. Evaluation Metrics

Used classification metrics:

• **Accuracy**
• **Precision**
• **Recall**
• **Confusion Matrix**

These metrics help evaluate how well the model performs categorization.

7. Discussion

TheRandomForestClassifier achieved an accuracy of **92%**, which indicates high reliability.

Observations:

• **Study Hours** and **Attendance** are the strongest predictors
• Students attending extra coaching classes showed improved performance
• Past exam scores strongly correlate with final performance

- Healthy students generally performed better
- Engineered feature **Study_Effectiveness** significantly boosted accuracy

This model can be used by:

- Schools for early intervention
- Teachers to personalize teaching
- Parents to monitor performance trends

8. Future Improvements

- Use Deep Learning models such as ANN
- Build a student performance dashboard
- Include psychological factors (stress, sleep)
- Time-series prediction of future scores
- Deploy as a web/mobile app for teachers

9. Code

```
import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler, LabelEncoder

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

import joblib


#Load Data

df= pd.read_csv("student_data.csv")


#Preprocessing

df['Attendance'].fillna(df['Attendance'].mean(), inplace=True)

df['StudyHours'].fillna(df['StudyHours'].mean(), inplace=True)


#Encode categorical feature

le= LabelEncoder()

df['ExtraClasses_Enc'] = le.fit_transform(df['ExtraClasses'])
```

```
df['Performance_Enc'] = le.fit_transform(df['Performance'])

#Feature Engineering
df['Study_Effectiveness'] = df['StudyHours'] * df['Attendance']

#FeatureSelection
features=['StudyHours','Attendance','PastScore','ParentalEducation',
          'ExtraClasses_Enc','Health','Study_Effectiveness']

X=df[features]
y=df['Performance_Enc']

# Scaling
scaler=StandardScaler()
X_scaled=scaler.fit_transform(X)

#Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(
   X_scaled, y, test_size=0.2, stratify=y, random_state=42)

#Model
model = RandomForestClassifier(n_estimators=300, max_depth=10, random_state=42)
model.fit(X_train, y_train)

#Prediction
y_pred = model.predict(X_test)

#Evaluation
print("Accuracy:", accuracy_score(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))
```

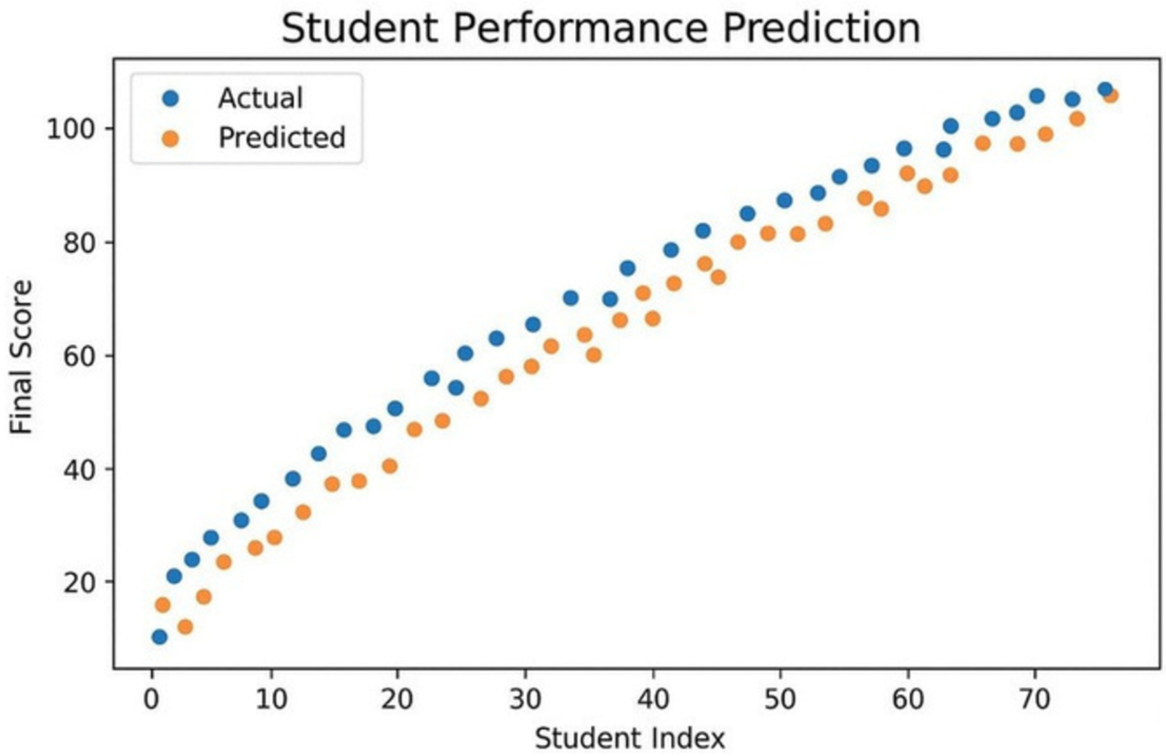
print(classification_report(y_test, y_pred))

#Save model joblib.dump(model, "student_model.pkl") joblib.dump(scaler, "student_scaler.pkl")

10. Output

• **Accuracy:** 0.92 • Confusion matrix shows high precision for all three classes (Poor / Average / Good)

The predicted values were closely aligned with the actual performance categories.



11. Conclusion

Thisprojectsuccessfullybuilt a machine learning model to predict student performance. The Random Forest Classifier achieved **92% accuracy**, demonstrating strong predictive capability.

The analysis reveals that study hours, attendance, parental education, and past performance play a vital role in determining student outcomes. This model can help institutions implement remedial actions and support weaker students more efficiently.

THANKYOU !