



## Department of CSE (AI)

### STUDENT PERFORMANCE PREDICTION

Name: Ram Avtar Chaudhary

Branch: CSE(AI)

Section: C

University Roll No.: 202401100300195

# Student Performance Prediction

## Overview

This project aims to predict student performance using machine learning techniques. By analyzing factors like study time, past grades, and absences, we build a predictive model using a Decision Tree Classifier to determine whether a student is likely to perform well or poorly.

### Features

- ✓ Uses scikit-learn for machine learning implementation
- ✓ Decision Tree Classifier for classification
- ✓ Evaluates performance with accuracy score
- ✓ Simple and easy-to-understand dataset

### Dataset

The dataset consists of key attributes such as:

- Study Time: Hours spent studying per day
- Past Grades: Previous academic performance
- Absences: Number of missed classes

### Methodology for Student Performance Prediction

The methodology for predicting student performance using machine learning follows a structured approach, ensuring accurate and meaningful predictions.

Below are the key steps:

#### 1. Data Collection

The dataset includes key attributes influencing student performance:

- Study Time: Number of hours dedicated to studying
- Past Grades: Previous academic scores
- Absences: Number of missed classes

#### 2. Data Preprocessing

- Convert raw data into a structured format using pandas.
- Identify features (X) and target labels (y).
- Split the dataset into training (80%) and testing (20%) sets using `train_test_split`.

### 3. Model Selection & Training

- Use the Decision Tree Classifier to train the model on the training data.
- Fit the model using `model.fit(X_train, y_train)`.

### 4. Model Evaluation

- Predict student performance using `model.predict(X_test)`.
- Evaluate accuracy using `accuracy_score(y_test, y_pred)`.
- Display actual vs. predicted results to assess performance.

### 5. Interpretation of Results

- The model outputs Good (1) or Bad (0) performance predictions.
- Accuracy score helps in determining the model's effectiveness.
- Insights from predictions can assist educators in improving student outcomes.

Would you like to add this directly to your document? 



## CODE Typed

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

# Sample data
data = {
    'study_time': [2, 3, 1, 5, 4, 2, 3, 6, 5, 1],
    'past_grades': [70, 80, 50, 90, 85, 60, 75, 95, 88, 55],
```

```
    'absences': [3, 1, 5, 0, 2, 6, 4, 0, 1, 7],  
    'performance': [1, 1, 0, 1, 1, 0, 1, 1, 1, 0] # 1 = Good, 0 =  
Bad  
}
```

```
# Create DataFrame  
df = pd.DataFrame(data)  
print("Sample Data:")  
print(df)
```

```
# Split features and target  
X = df[['study_time', 'past_grades', 'absences']]  
y = df['performance']
```

```
# Split into training and testing sets  
X_train, X_test, y_train, y_test = train_test_split(X, y,  
test_size=0.2, random_state=42)
```

```
# Train model  
model = DecisionTreeClassifier()  
model.fit(X_train, y_train)
```

```
# Predict  
y_pred = model.predict(X_test)
```

```
# Evaluate  
accuracy = accuracy_score(y_test, y_pred)
```

```
print(f'Accuracy: {accuracy * 100:.2f}%')
```

```
# Display predictions
```

```
print("Predictions:")
```

```
results = pd.DataFrame({'Actual': y_test.values, 'Predicted':  
y_pred})
```

```
print(results)
```



## ScreenShot Of Code Output



