Student Performance Prediction using Machine Learning

```
# Import libraries
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix

# Step 1: Load Dataset
data = pd.read_csv("student.csv")

# Step 2: Handle Missing Values
data['Attendance'].fillna(data['Attendance'].mean(), inplace=True)
data['Previous_Marks'].fillna(data['Previous_Marks'].mean(), inplace=True)
data['Study_Hours'].fillna(data['Study_Hours'].mean(), inplace=True)

data['Gender'].fillna(data['Gender'].mode()[0], inplace=True)
data['Assignments'].fillna(data['Assignments'].mode()[0], inplace=True)

# Step 3: Convert Categorical Data
le = LabelEncoder()
data['Gender'] = le.fit_transform(data['Gender'])
data['Assignments'] = le.fit_transform(data['Assignments'])
data['Result'] = le.fit_transform(data['Result'])

# Step 4: Normalize Features
scaler = StandardScaler()
features = ['Study_Hours', 'Attendance', 'Previous_Marks']
data[features] = scaler.fit_transform(data[features])

# Step 5: Split Data
X = data.drop('Result', axis=1)
y = data['Result']

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# Step 6: Train Model
model = LogisticRegression()
model.fit(X_train, y_train)

# Step 7: Test Model
y_pred = model.predict(X_test)

# Step 8: Evaluate Model
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
```