

```
# Task 1: Student Pass/Fail Prediction
# Step 1: Install and Import Required Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, ConfusionMatrixDisplay

# Step 2: Create Sample Dataset
data = {
    'Study Hours': [2, 3, 4, 5, 1, 6, 7, 2.5, 3.5, 8, 4.5, 1.5, 9, 6.5, 2],
    'Attendance': [60, 65, 70, 80, 50, 90, 95, 55, 75, 96, 85, 40, 98, 93, 52],
    'Pass': [0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0]
}
df = pd.DataFrame(data)

# Step 3: Data Exploration
print("\nDataset Preview:")
print(df.head())
print("\nChecking for missing values:")
print(df.isnull().sum())
# Visualizing Study Hours vs Attendance
plt.figure(figsize=(8, 5))
sns.scatterplot(x='Study Hours', y='Attendance', hue='Pass', data=df, palette='Set1')
plt.title('Study Hours vs Attendance (Pass/Fail)')
plt.show()
```

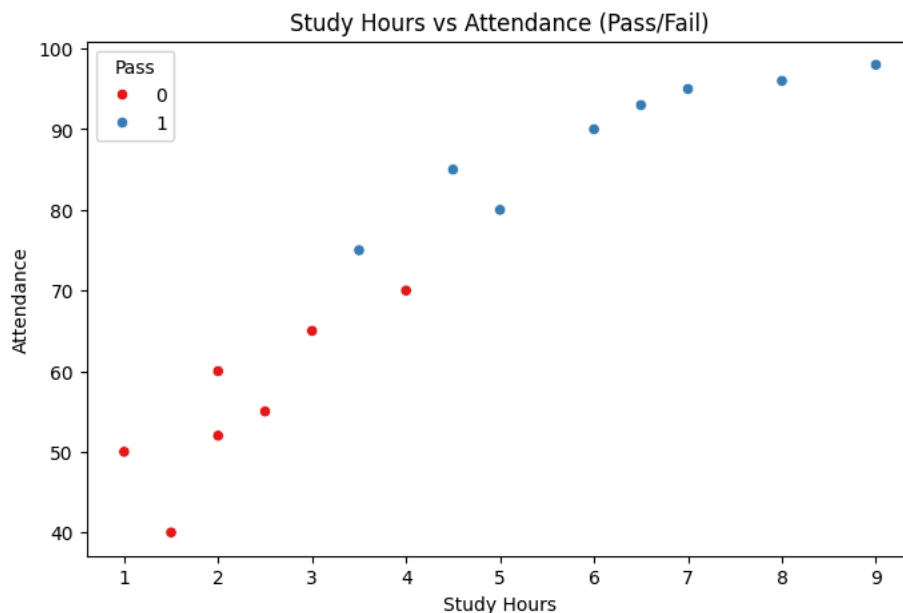


Dataset Preview:

	Study Hours	Attendance	Pass
0	2.0	60	0
1	3.0	65	0
2	4.0	70	0
3	5.0	80	1
4	1.0	50	0

Checking for missing values:

```
Study Hours    0
Attendance     0
Pass           0
dtype: int64
```



```
# Step 4: Split Data
X = df[['Study Hours', 'Attendance']]
y = df['Pass']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# Step 5: Train Logistic Regression Model
model = LogisticRegression()
model.fit(X_train, y_train)
```

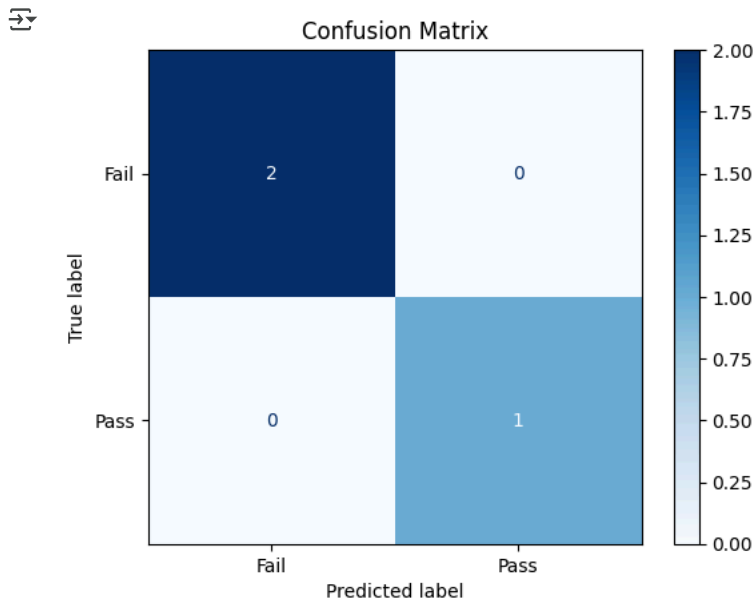
LogisticRegression ⓘ ?

```
# Step 6: Predictions
y_pred = model.predict(X_test)
```

```
# Step 7: Model Evaluation
accuracy = accuracy_score(y_test, y_pred)
cm = confusion_matrix(y_test, y_pred)
print(f"\nAccuracy: {accuracy:.2f}")
```

Accuracy: 1.00

```
# Display Confusion Matrix
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=["Fail", "Pass"])
disp.plot(cmap='Blues')
plt.title('Confusion Matrix')
plt.show()
```



```
# Step 8: Insights
print("\nModel Coefficients (Impact of features):")
coefficients = pd.DataFrame(model.coef_[0], index=X.columns, columns=["Coefficient"])
print(coefficients)
print("\nInsight: Higher study hours and attendance are strongly associated with passing.")
```

Model Coefficients (Impact of features):

	Coefficient
Study Hours	-0.059434
Attendance	0.738628

Insight: Higher study hours and attendance are strongly associated with passing.