```python
# --- Student Grade Predictor (binary: Pass/Fail) ---
# Runs in Colab. No installs needed.

import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# 1) Make a small synthetic dataset (you can say you simulated based on typical patterns)
rng = np.random.default_rng(42)
n = 600
study_hours   = rng.uniform(0, 6, n)           # 0–6 hrs/day
attendance    = rng.uniform(50, 100, n)        # %
test1         = rng.normal(65, 12, n).clip(0, 100)
test2         = rng.normal(68, 12, n).clip(0, 100)

# Ground truth rule (simple but realistic): weighted sum + noise
score = (0.35*test1 + 0.40*test2 + 0.15*attendance + 6*study_hours) + rng.normal(0, 5, n)
y = (score >= 70).astype(int)  # 1 = Pass, 0 = Fail

X = pd.DataFrame({
    "study_hours": study_hours,
    "attendance": attendance,
    "test1": test1,
    "test2": test2
})

# 2) Train/test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=7, stratify=y)

# 3) Scale + model
scaler = StandardScaler().fit(X_train)
Xtr = scaler.transform(X_train)
Xte = scaler.transform(X_test)

model = LogisticRegression(max_iter=200)
model.fit(Xtr, y_train)

# 4) Evaluate
pred = model.predict(Xte)
print("Accuracy:", round(accuracy_score(y_test, pred), 3))
print("\nClassification Report:\n", classification_report(y_test, pred, target_names=["Fail","Pass"]))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, pred))

# 5) Simple predictor function
def predict_grade(study_hours, attendance, test1, test2):
    import numpy as np
    x = np.array([[study_hours, attendance, test1, test2]])
    x = scaler.transform(x)
    p = model.predict_proba(x)[0,1]
    label = "Pass" if p >= 0.5 else "Fail"
    return label, float(p)

# Example:
label, p = predict_grade(3.0, 85, 72, 75)
print(f"\nExample prediction → {label} (prob={p:.2f})")

# 6) Save artifacts (optional)
import joblib
joblib.dump(model, "grade_model.joblib")
joblib.dump(scaler, "grade_scaler.joblib")
```

```
Accuracy: 0.913

Classification Report:
              precision    recall  f1-score   support

        Fail       0.80      0.90      0.85        41
        Pass       0.96      0.92      0.94       109

    accuracy                           0.91       150
   macro avg       0.88      0.91      0.89       150
weighted avg       0.92      0.91      0.91       150
```

```
Confusion Matrix:
 [[ 37   4]
 [  9 100]]

Example prediction → Pass (prob=0.99)
/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but Sta
  warnings.warn(
['grade_scaler.joblib']
```

Start coding or generate with AI.