

```
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report

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

X = df.drop(columns=['Final_Result'])
y = df['Final_Result']

model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X, y)

y_pred = model.predict(X)

print("Accuracy:", accuracy_score(y, y_pred))
print("\nClassification Report:\n", classification_report(y, y_pred))
```

↗ Accuracy: 1.0

Classification Report:				
	precision	recall	f1-score	support
Fail	1.00	1.00	1.00	2
Pass	1.00	1.00	1.00	3
accuracy			1.00	5
macro avg	1.00	1.00	1.00	5
weighted avg	1.00	1.00	1.00	5

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import learning_curve

# Confusion matrix
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['Fail', 'Pass'], yticklabels=['Fail', 'Pass'])
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()

# Learning curve
train_sizes, train_scores, test_scores = learning_curve(model, X, y, cv=3)
train_scores_mean = train_scores.mean(axis=1)
test_scores_mean = test_scores.mean(axis=1)

plt.plot(train_sizes, train_scores_mean, label="Training score")
plt.plot(train_sizes, test_scores_mean, label="Validation score")
plt.xlabel("Training Set Size")
plt.ylabel("Accuracy")
plt.title("Learning Curve")
plt.legend()
plt.show()
```

Confusion Matrix

	Predicted Fail	Predicted Pass
Actual Fail	1	0
Actual Pass	0	0

```
warnings.warn(
/usr/local/lib/python3.11/dist-packages/sklearn/model_selection/_validation.py:2180: RuntimeWarning: Removed duplicate entries from
warnings.warn(
```

