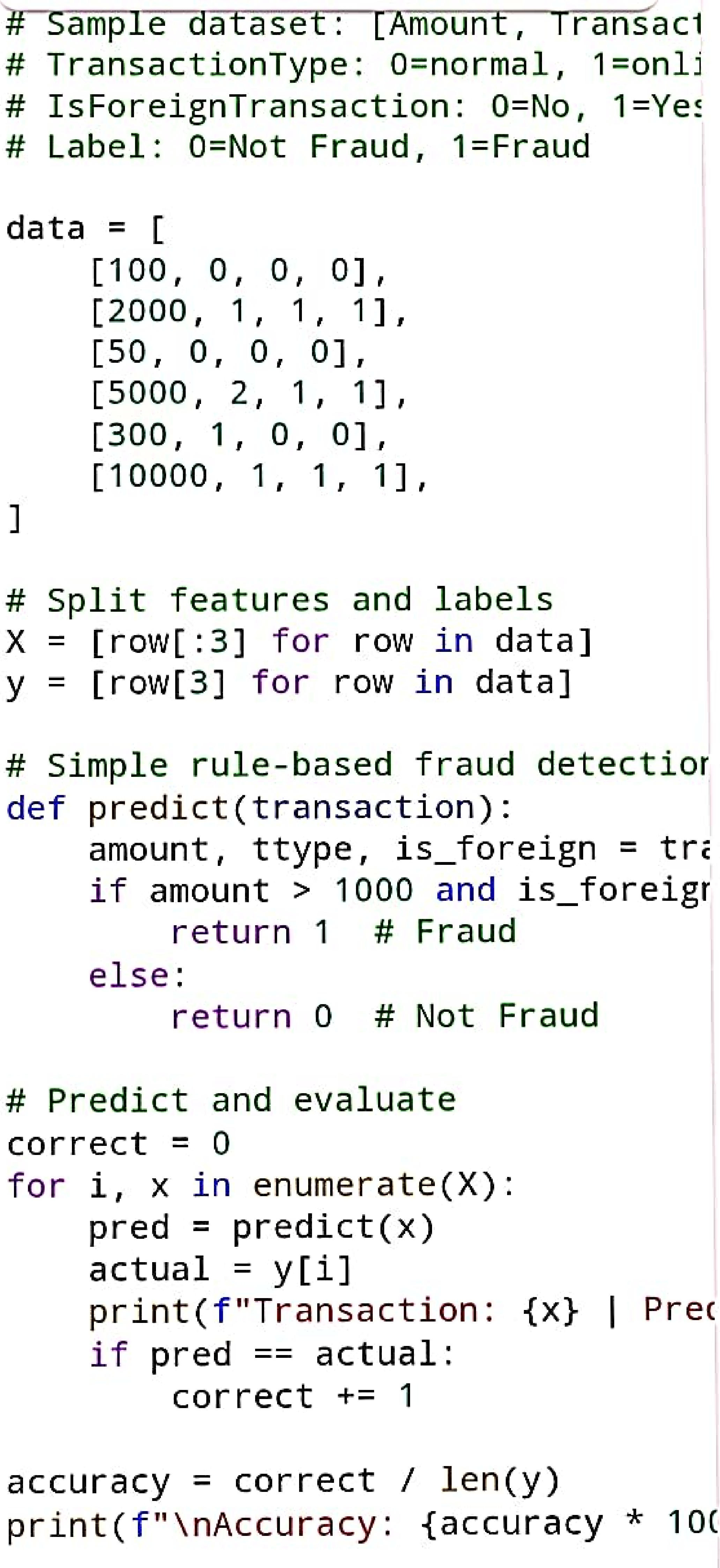
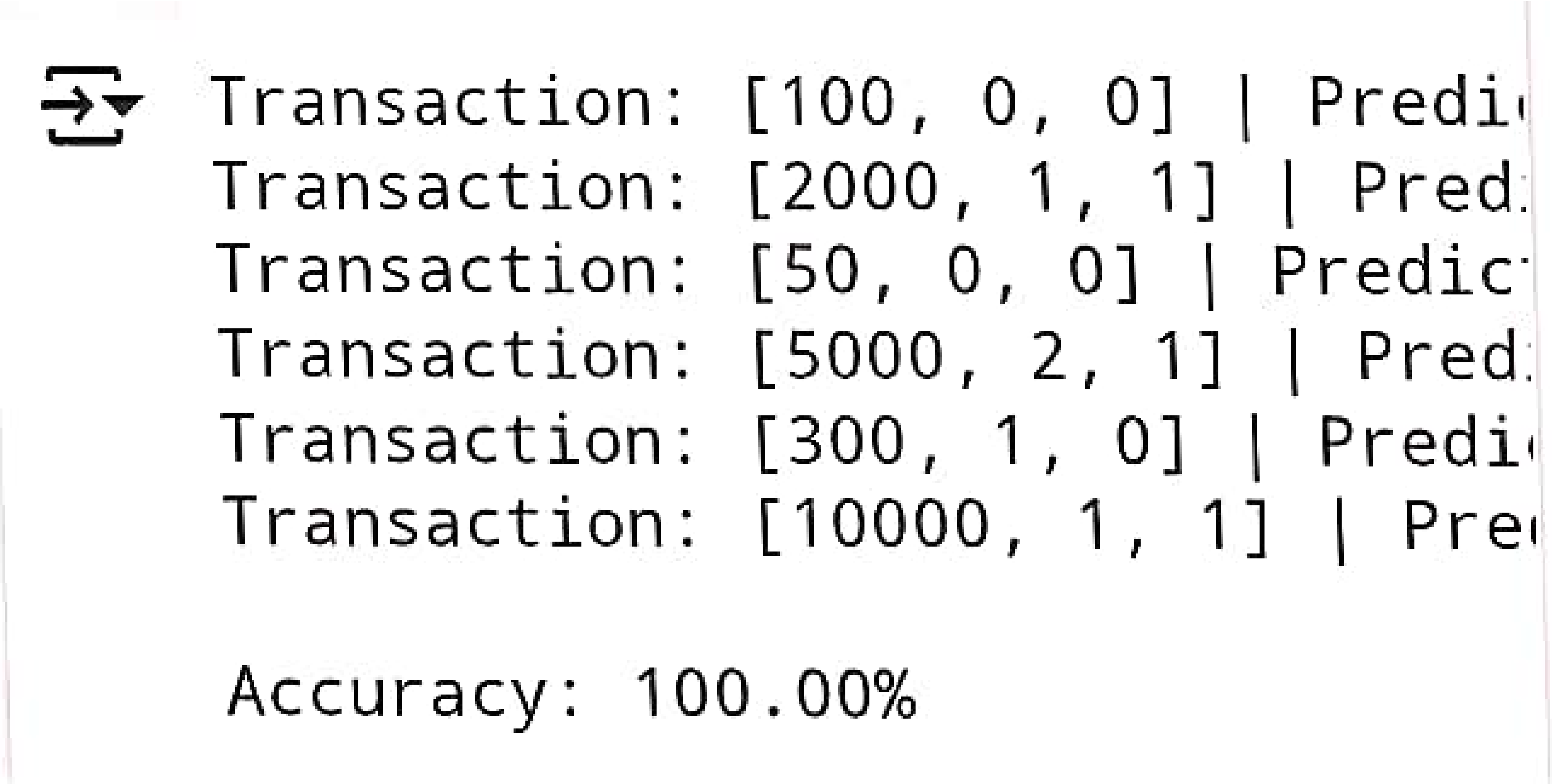
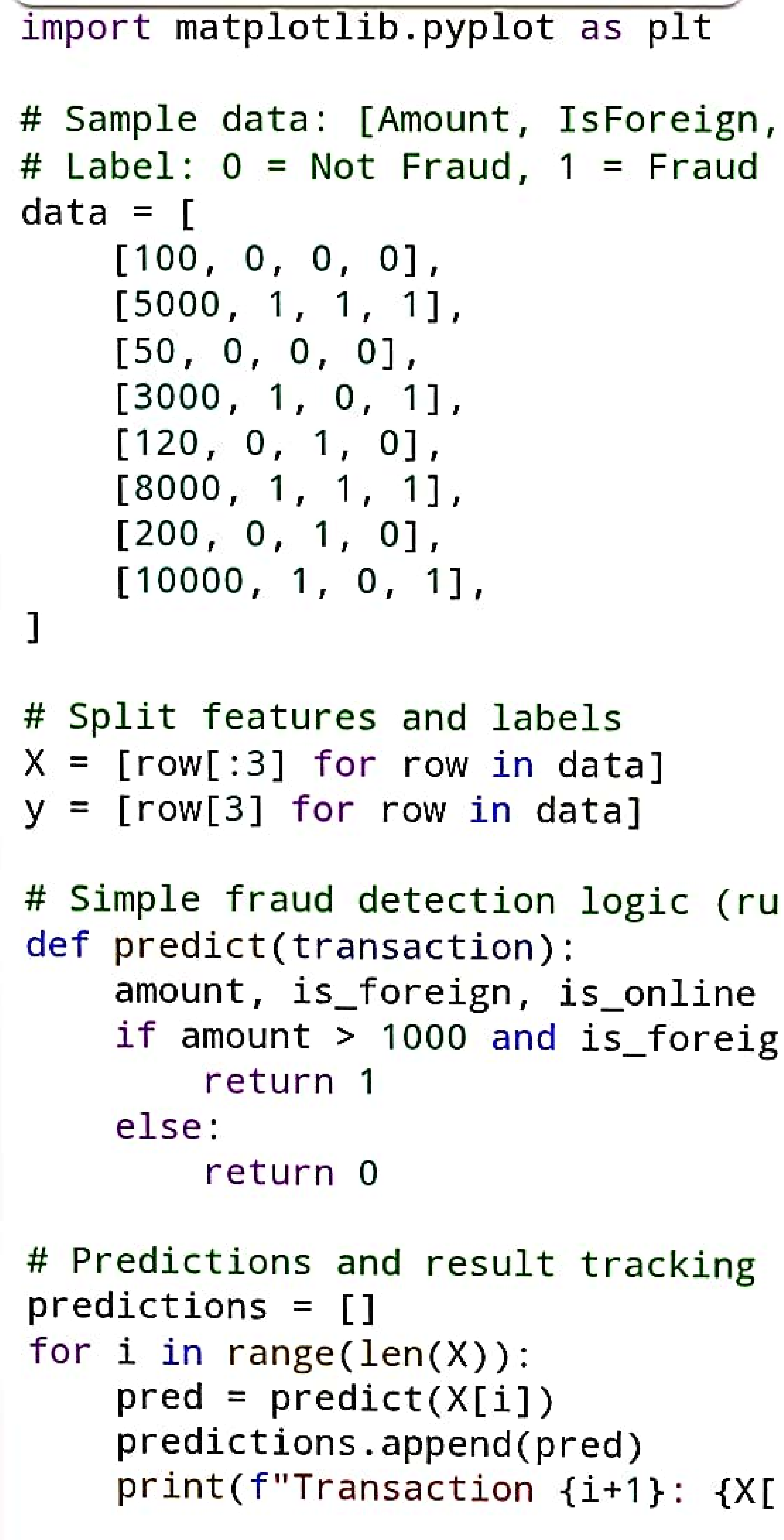


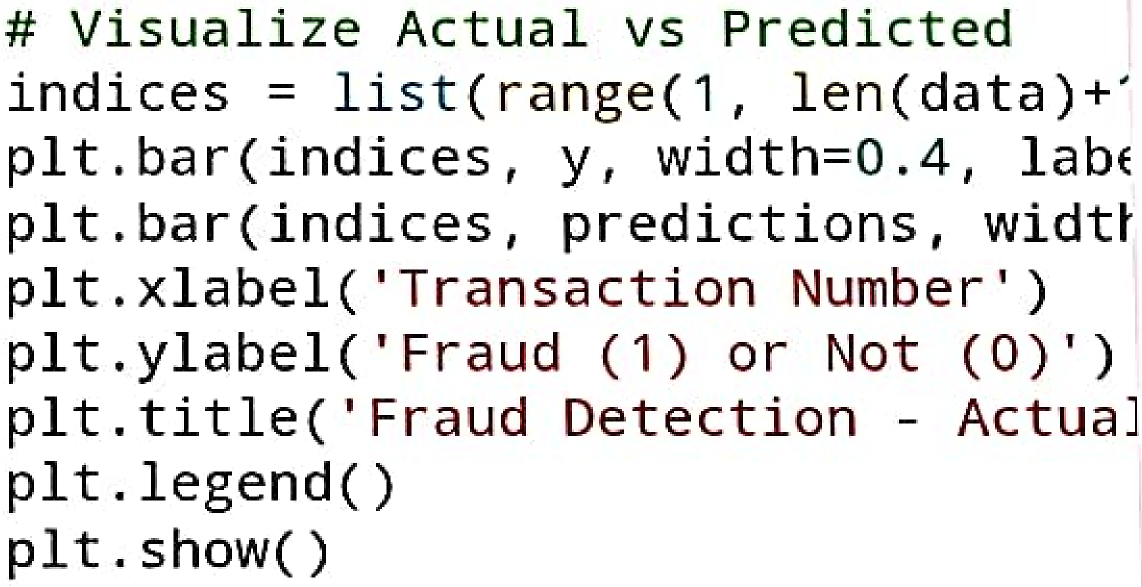
```
# Sample dataset: [Amount, Transact
# TransactionType: 0=normal, 1=onli
# IsForeignTransaction: 0=No, 1=Yes
# Label: 0=Not Fraud, 1=Fraud

data = [
    [100, 0, 0, 0],
    [2000, 1, 1, 1],
    [50, 0, 0, 0],
    [5000, 2, 1, 1],
    [300, 1, 0, 0],
    [10000, 1, 1, 1],
]

# Split features and labels
X = [row[:3] for row in data]
y = [row[3] for row in data]

# Simple rule-based fraud detection
def predict(transaction):
    amount, ttype, is_foreign = tra
    if amount > 1000 and is_foreign
        return 1  # Fraud
    else:
        return 0  # Not Fraud

# Predict and evaluate
correct = 0
for i, x in enumerate(X):
    pred = predict(x)
    actual = y[i]
    print(f"Transaction: {x} | Pred
    if pred == actual:
        correct += 1

accuracy = correct / len(y)
print(f"\nAccuracy: {accuracy * 100
```



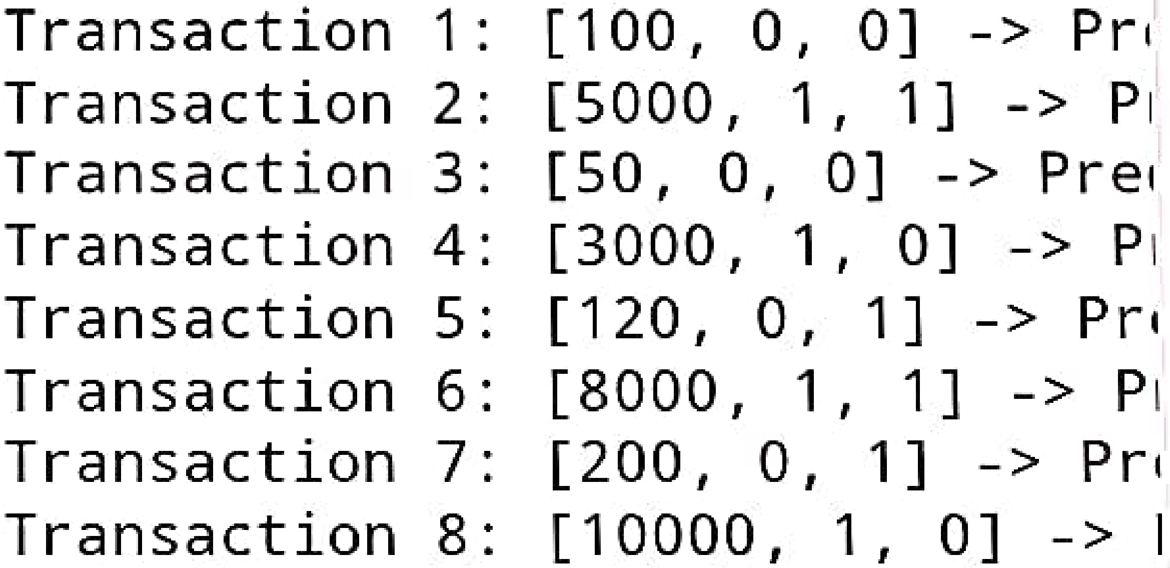
```
Transaction: [100, 0, 0] | Predi
Transaction: [2000, 1, 1] | Pred
Transaction: [50, 0, 0] | Predic
Transaction: [5000, 2, 1] | Pred
Transaction: [300, 1, 0] | Predi
Transaction: [10000, 1, 1] | Pre

Accuracy: 100.00%
```



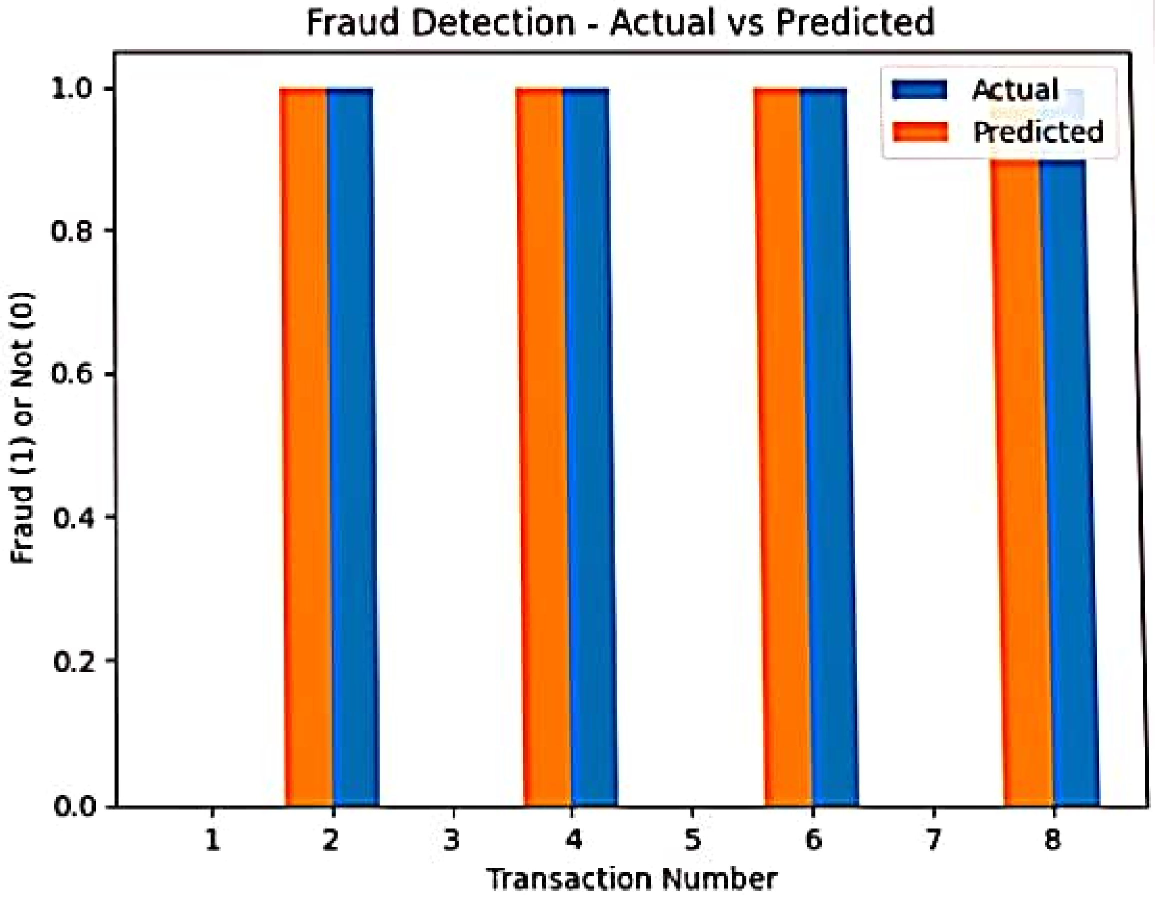
```
import matplotlib.pyplot as plt

# Sample data: [Amount, IsForeign,
# Label: 0 = Not Fraud, 1 = Fraud
data = [
    [100, 0, 0, 0],
    [5000, 1, 1, 1],
    [50, 0, 0, 0],
    [3000, 1, 0, 1],
    [120, 0, 1, 0],
    [8000, 1, 1, 1],
    [200, 0, 1, 0],
    [10000, 1, 0, 1],
]

# Split features and labels
X = [row[:3] for row in data]
y = [row[3] for row in data]

# Simple fraud detection logic (ru
def predict(transaction):
    amount, is_foreign, is_online
    if amount > 1000 and is_foreig
        return 1
    else:
        return 0

# Predictions and result tracking
predictions = []
for i in range(len(X)):
    pred = predict(X[i])
    predictions.append(pred)
    print(f"Transaction {i+1}: {X[
```

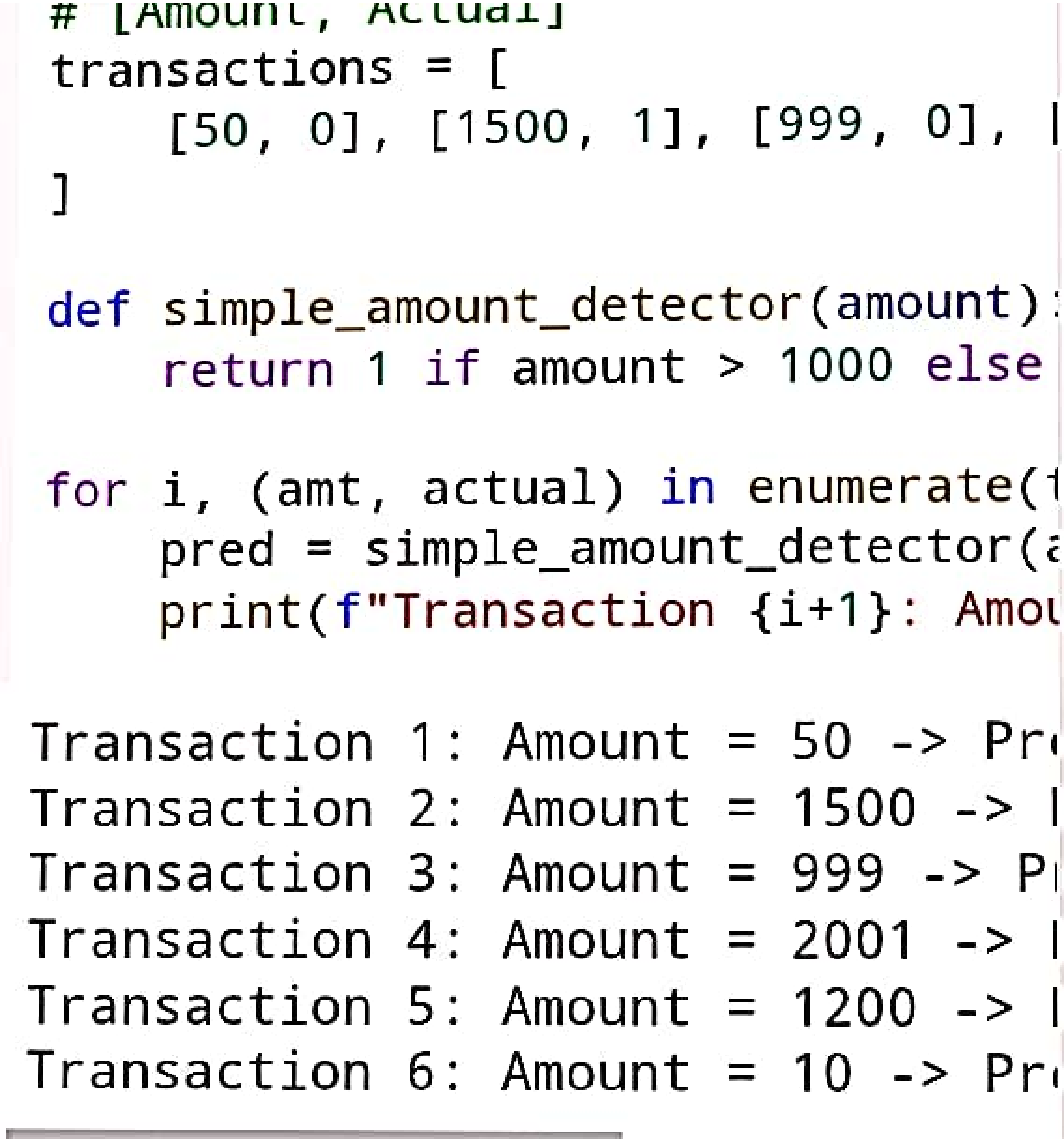


```
# Visualize Actual vs Predicted
indices = list(range(1, len(data)+
plt.bar(indices, y, width=0.4, labe
plt.bar(indices, predictions, widt
plt.xlabel('Transaction Number')
plt.ylabel('Fraud (1) or Not (0)')
plt.title('Fraud Detection - Actual
plt.legend()
plt.show()
```



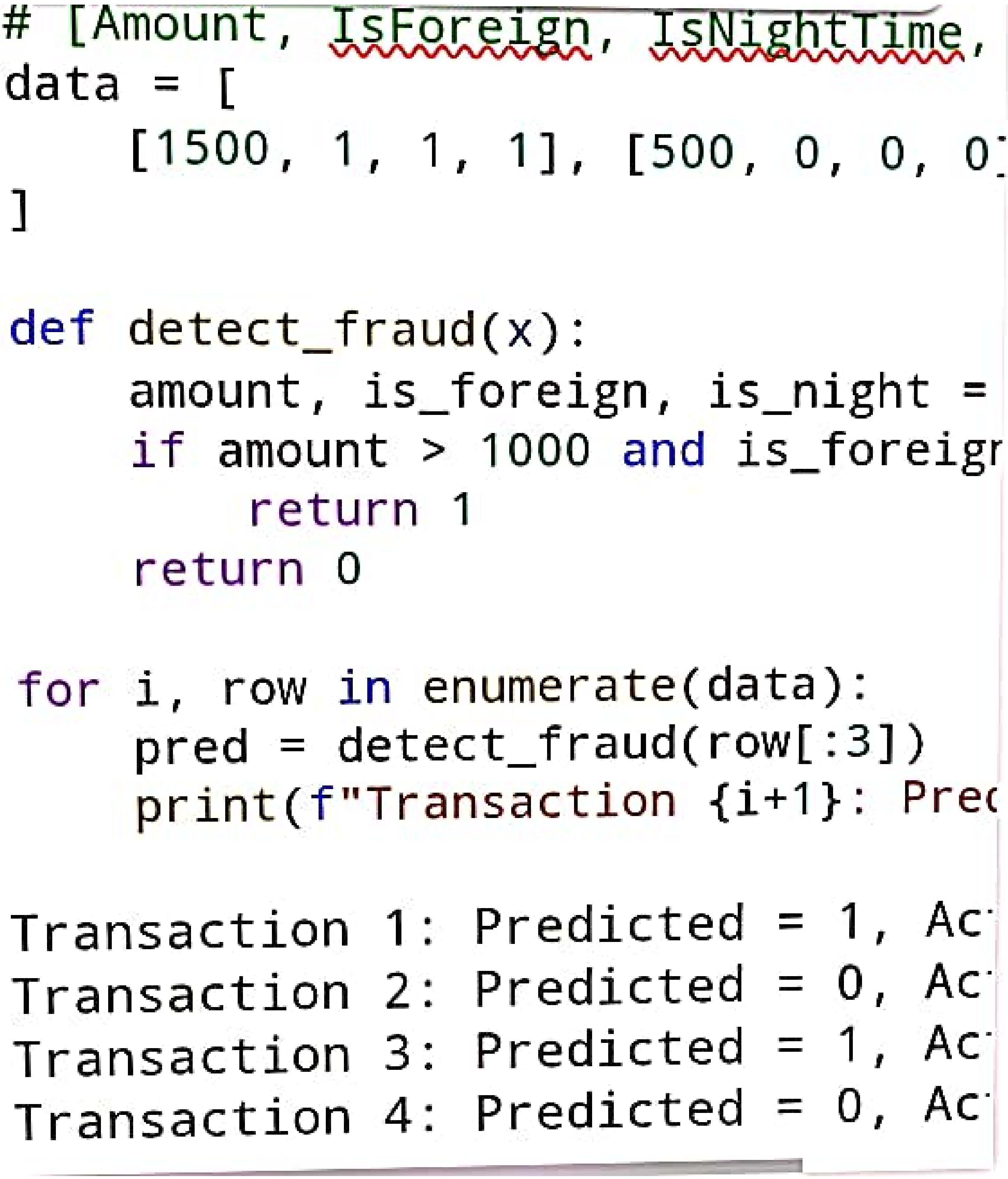
```
Transaction 1: [100, 0, 0] -> Pr
Transaction 2: [5000, 1, 1] -> P
Transaction 3: [50, 0, 0] -> Pre
Transaction 4: [3000, 1, 0] -> P
Transaction 5: [120, 0, 1] -> Pr
Transaction 6: [8000, 1, 1] -> P
Transaction 7: [200, 0, 1] -> Pr
Transaction 8: [10000, 1, 0] ->
```





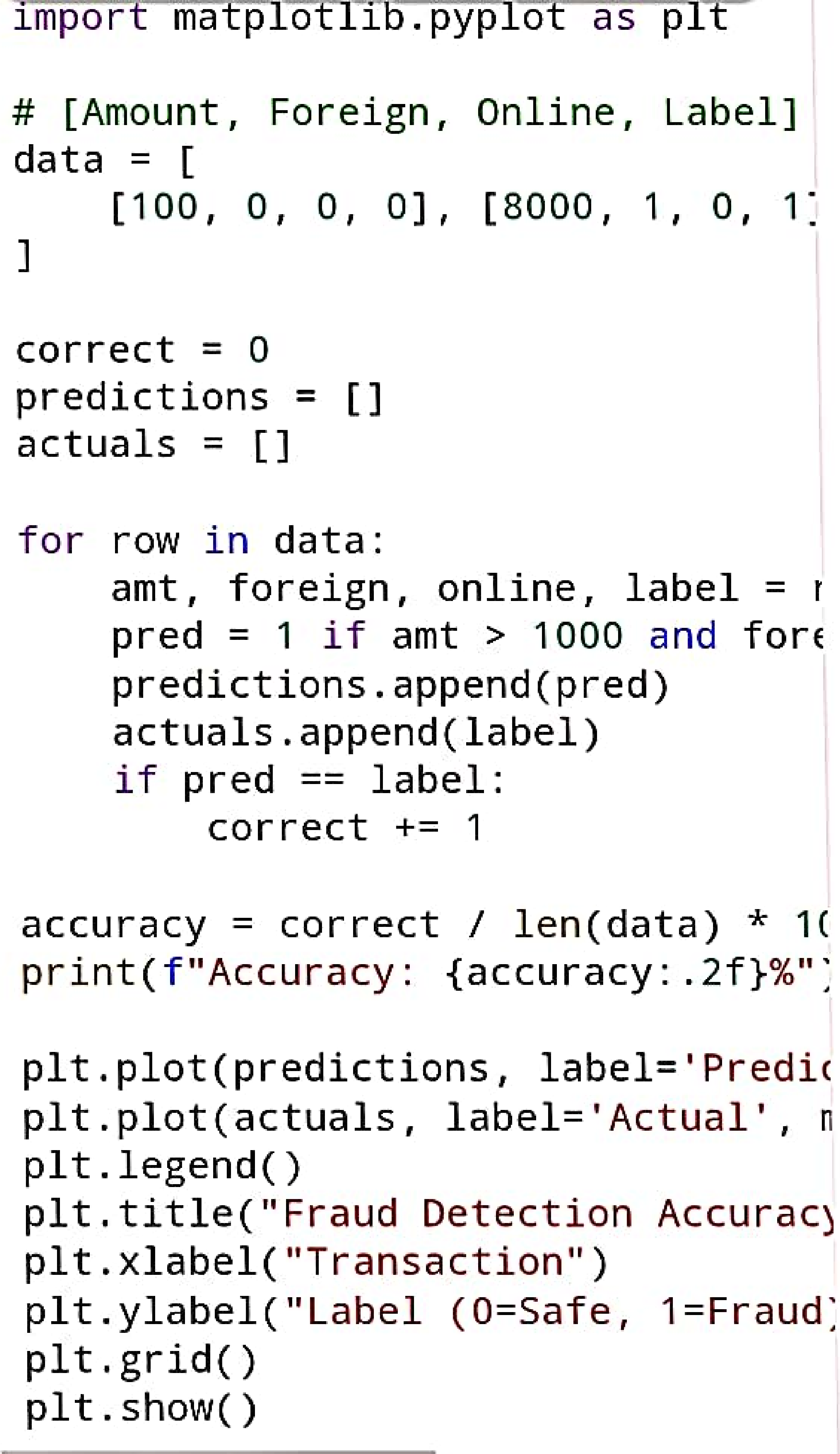
```
# [Amount, Actual]
transactions = [
    [50, 0], [1500, 1], [999, 0], [
]

def simple_amount_detector(amount):
    return 1 if amount > 1000 else

for i, (amt, actual) in enumerate(t
    pred = simple_amount_detector(a
    print(f"Transaction {i+1}: Amou
```

```
Transaction 1: Amount = 50 -> Pr
Transaction 2: Amount = 1500 -> P
Transaction 3: Amount = 999 -> Pr
Transaction 4: Amount = 2001 -> P
Transaction 5: Amount = 1200 -> P
Transaction 6: Amount = 10 -> Pr
```

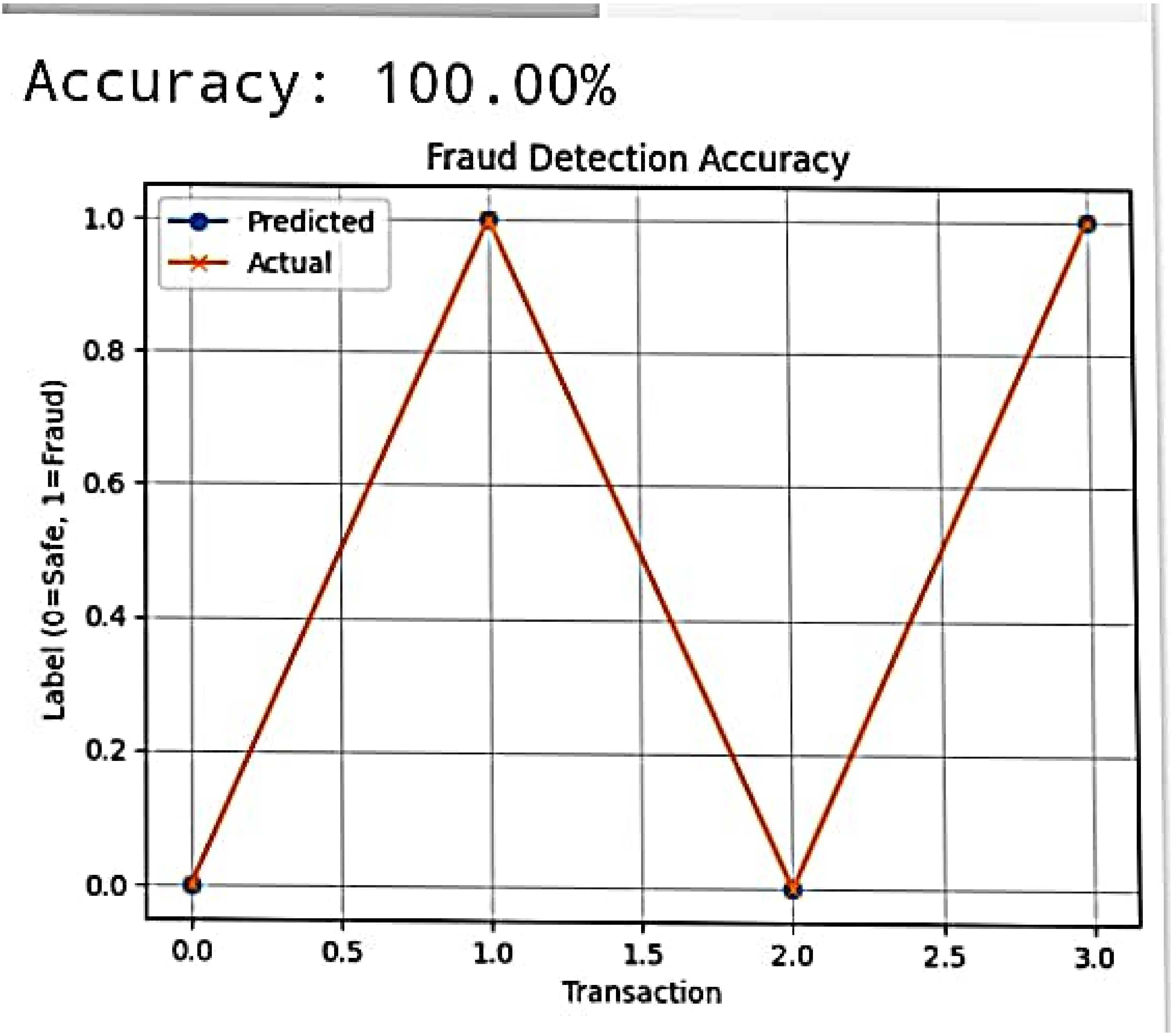


```
# [Amount, IsForeign, IsNightTime,
data = [
    [1500, 1, 1, 1], [500, 0, 0, 0]
]

def detect_fraud(x):
    amount, is_foreign, is_night =
    if amount > 1000 and is_foreign
        return 1
    return 0

for i, row in enumerate(data):
    pred = detect_fraud(row[:3])
    print(f"Transaction {i+1}: Prec
```

```
Transaction 1: Predicted = 1, Ac
Transaction 2: Predicted = 0, Ac
Transaction 3: Predicted = 1, Ac
Transaction 4: Predicted = 0, Ac
```



```
import matplotlib.pyplot as plt

# [Amount, Foreign, Online, Label]
data = [
    [100, 0, 0, 0], [8000, 1, 0, 1]
]

correct = 0
predictions = []
actuals = []

for row in data:
    amt, foreign, online, label = r
    pred = 1 if amt > 1000 and fore
    predictions.append(pred)
    actuals.append(label)
    if pred == label:
        correct += 1

accuracy = correct / len(data) * 10
print(f"Accuracy: {accuracy:.2f}%")

plt.plot(predictions, label='Predic
plt.plot(actuals, label='Actual', m
plt.legend()
plt.title("Fraud Detection Accuracy
plt.xlabel("Transaction")
plt.ylabel("Label (0=Safe, 1=Fraud)
plt.grid()
plt.show()
```



```
Accuracy: 100.00%
```