In [46]:

```python
import pandas as pd
import numpy as np
import xgboost as xgb
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report
```

In [48]:

```python
df = pd.read_csv("Credit Card Fraud Detection.csv")
df = df.drop(columns=["Time"])

X = df.drop(columns=["Class"])
y = df["Class"]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta

# Scale features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

In [50]:

```python
dtrain = xgb.DMatrix(X_train_scaled, label=y_train)
dtest = xgb.DMatrix(X_test_scaled, label=y_test)

params = {
    'objective': 'binary:logistic',
    'eval_metric': 'auc',
    'max_depth': 5,
    'eta': 0.2,
    'seed': 42
}

model = xgb.train(params, dtrain, num_boost_round=100)
```

In [58]:

```python
probs = model.predict(dtest)

for t in [0.5, 0.4, 0.3, 0.2, 0.1]:
    preds = (probs > t).astype(int)
    fraud_total = np.sum(preds == 1)
    print(f"Threshold {t:.1f} →  Predicted Fraud Count: {fraud_total}")
```

```
Threshold 0.5 →  Predicted Fraud Count: 91
Threshold 0.4 →  Predicted Fraud Count: 93
Threshold 0.3 →  Predicted Fraud Count: 95
Threshold 0.2 →  Predicted Fraud Count: 98
Threshold 0.1 →  Predicted Fraud Count: 99
```

In [60]:

```python
# Scale full dataset
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
dall = xgb.DMatrix(X_scaled)

# Train on all
model = xgb.train(params, xgb.DMatrix(X_scaled, label=y), num_boost_round=100)
```

```python
# Predict
probs_all = model.predict(dall)
preds_all = (probs_all > 0.1).astype(int)  # use lower threshold

fraud_total = np.sum(preds_all == 1)
print("Predicted total frauds in full dataset:", fraud_total)

# Actual
print("Actual total frauds:", np.sum(y == 1))
```

```
Predicted total frauds in full dataset: 506
Actual total frauds: 492
```

```
In [ ]:
```