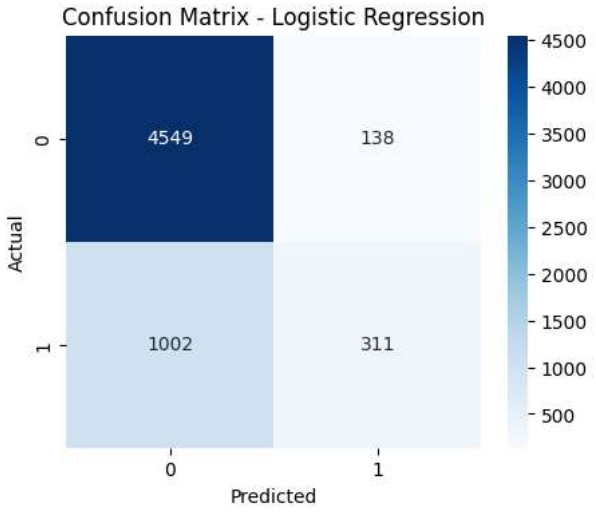
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score, confusion_matrix


df = pd.read_excel("/content/default of credit card clients.xls", skiprows=1)
print("Dataset Shape:", df.shape)
df.head()
```

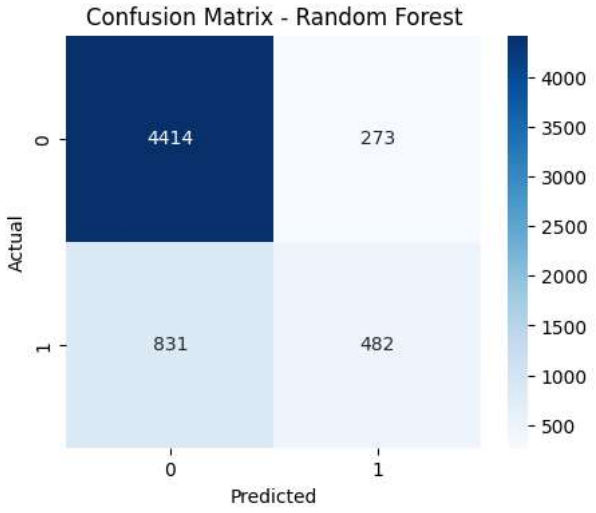
```
Dataset Shape: (30000, 25)
```

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	...	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2
0	1	20000	2	2	1	24	2	2	-1	-1	...	0	0	0	0	689
1	2	120000	2	2	2	26	-1	2	0	0	...	3272	3455	3261	0	1000
2	3	90000	2	2	2	34	0	0	0	0	...	14331	14948	15549	1518	1500
3	4	50000	2	2	1	37	0	0	0	0	...	28314	28959	29547	2000	2019
4	5	50000	1	2	1	57	-1	0	-1	0	...	20940	19146	19131	2000	36681

5 rows × 25 columns

```
df.drop('ID', axis=1, inplace=True)


X = df.drop('default payment next month', axis=1)
y = df['default payment next month']


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)


log_model = LogisticRegression(max_iter=1000)
log_model.fit(X_train, y_train)
y_pred_lr = log_model.predict(X_test)


rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
y_pred_rf = rf_model.predict(X_test)


def evaluate_model(y_true, y_pred, model_name):
    print(f"\n{model_name} Performance:")
    print("✅ Accuracy:", round(accuracy_score(y_true, y_pred), 4))
    print("✅ Precision:", round(precision_score(y_true, y_pred), 4))
    print("✅ Recall:", round(recall_score(y_true, y_pred), 4))
    print("✅ F1 Score:", round(f1_score(y_true, y_pred), 4))
    print("✅ ROC-AUC:", round(roc_auc_score(y_true, y_pred), 4))


def plot_confusion_matrix(y_true, y_pred, model_name):
    plt.figure(figsize=(5,4))
    sns.heatmap(confusion_matrix(y_true, y_pred), annot=True, fmt='d', cmap='Blues')
    plt.title(f"Confusion Matrix - {model_name}")
    plt.xlabel("Predicted")
    plt.ylabel("Actual")
    plt.show()

plot_confusion_matrix(y_test, y_pred_lr, "Logistic Regression")
plot_confusion_matrix(y_test, y_pred_rf, "Random Forest")
```





Start coding or generate with AI.