

```
# ✅ Step 1: Install required libraries
!pip install xgboost scikit-learn pandas matplotlib seaborn

# ✅ Step 2: Import libraries
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 LabelEncoder, StandardScaler
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from xgboost import XGBClassifier

# ✅ Step 3: Mount Google Drive to access the dataset
from google.colab import drive
drive.mount('/content/drive')

# ✅ Step 4: Load dataset
file_path = '/content/drive/MyDrive/Churn_Modelling.csv'
df = pd.read_csv(file_path)

# ✅ Step 5: Data Preprocessing
# Drop irrelevant columns
df.drop(['RowNumber', 'CustomerId', 'Surname'], axis=1, inplace=True)

# Encode categorical features
le = LabelEncoder()
df['Gender'] = le.fit_transform(df['Gender']) # Male=1, Female=0

# One-hot encode Geography
df = pd.get_dummies(df, columns=['Geography'], drop_first=True)

# Feature-target split
X = df.drop('Exited', axis=1)
y = df['Exited']


# Scale features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)


# ✅ Step 6: Train-test split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=


# ✅ Step 7: Train model
model = XGBClassifier(use_label_encoder=False, eval_metric='logloss')
model.fit(X_train, y_train)

# ✅ Step 8: Predictions
y_pred = model.predict(X_test)
```

```
y_probs = model.predict_proba(X_test)[: , 1]
```

```
#  Step 9: Evaluation
```

```
print("  Accuracy Score:", accuracy_score(y_test, y_pred))
```

```
print("\n  Classification Report:\n", classification_report(y_test, y_pred))
```

```
# Confusion Matrix
```

```
plt.figure(figsize=(5, 4))
```


```
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='Blues')
```

```
plt.xlabel('Predicted')
```

```
plt.ylabel('Actual')
```

```
plt.title('Confusion Matrix')
```

```
plt.show()
```

```
#  Step 10: Churn Probabilities
```

```
df_results = pd.DataFrame({
```


```
    'Actual': y_test.values,
```

```
    'Predicted': y_pred,
```

```
    'Churn_Probability': y_probs
```

```
})
```


```
print(df_results.head(10))
```

```
#  Step 11: Export predictions (for Power BI or Excel)
```

```
df_export = df.copy()
```

```
df_export['Churn_Probability'] = model.predict_proba(X_scaled)[: , 1]
```

```
df_export.to_csv('/content/drive/MyDrive/churn_predictions.csv', index=False)
```

```
print("  Predictions saved to Google Drive as churn_predictions.csv")
```



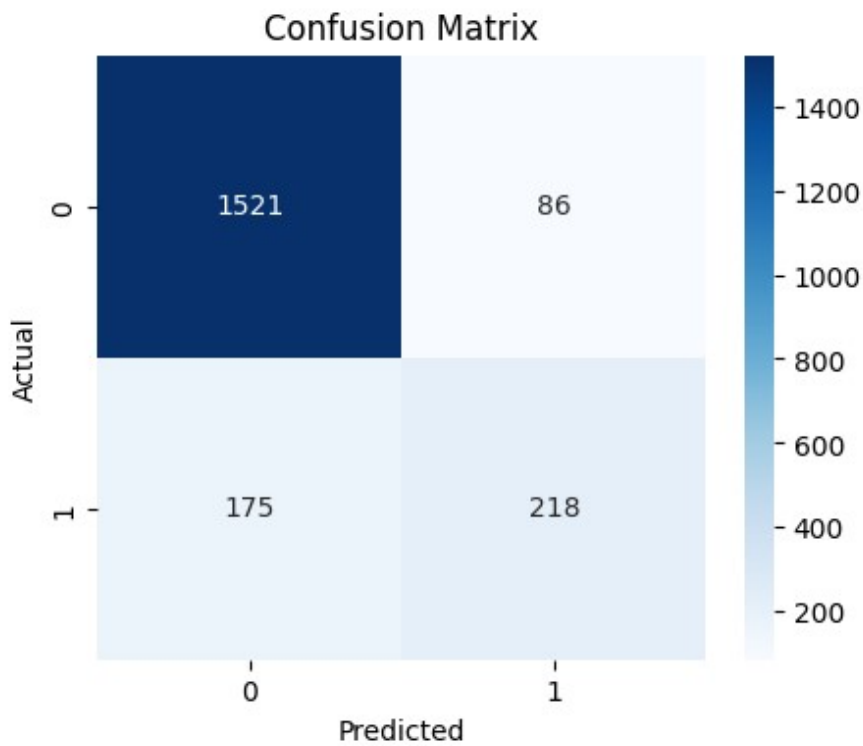
```
Requirement already satisfied: xgboost in /usr/local/lib/python3.11/dist-packages (3.0.
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (2.2.2
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages (3
Requirement already satisfied: seaborn in /usr/local/lib/python3.11/dist-packages (0.13
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (from x
Requirement already satisfied: nvidia-nccl-cu12 in /usr/local/lib/python3.11/dist-packa
Requirement already satisfied: scipy in /usr/local/lib/python3.11/dist-packages (from x
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packa
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packages (fr
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packa
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (fr
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mou
/usr/local/lib/python3.11/dist-packages/xgboost/training.py:183: UserWarning: [09:20:59
Parameters: { "use_label_encoder" } are not used.
```

```
bst.update(dtrain, iteration=i, fobj=obj)
```

◆ Accuracy Score: 0.8695

◆ Classification Report:

	precision	recall	f1-score	support
0	0.90	0.95	0.92	1607
1	0.72	0.55	0.63	393
accuracy			0.87	2000
macro avg	0.81	0.75	0.77	2000
weighted avg	0.86	0.87	0.86	2000



	Actual	Predicted	Churn_Probability
0	0	0	0.021329
1	0	0	0.008369
2	0	0	0.116290
3	0	0	0.019407
4	0	0	0.137873
5	0	0	0.014634
6	0	0	0.053385
7	1	0	0.119835
8	0	0	0.251440
9	0	1	0.515188

✓ Predictions saved to Google Drive as churn_predictions.csv

