

```
In [2]: import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3' # Suppress TensorFlow INFO/WARNING

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler, LabelEncoder
from sklearn.metrics import accuracy_score, confusion_matrix
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Input
```

```
In [3]: # --- Load Dataset ---
df = pd.read_csv("Churn_Modelling.csv")
```

```
In [4]: # --- Preprocessing ---
df = df.drop(columns=['RowNumber', 'CustomerId', 'Surname'])
df['Gender'] = LabelEncoder().fit_transform(df['Gender'])
df = pd.get_dummies(df, columns=['Geography'], drop_first=True)

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

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

scaler = MinMaxScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
In [5]: # --- Neural Network ---
model = Sequential([
    Input(shape=(X_train.shape[1],)),
    Dense(12, activation='relu'),
    Dense(8, activation='relu'),
    Dense(1, activation='sigmoid')
])

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

```
In [6]: # Train model silently
model.fit(X_train, y_train, epochs=20, batch_size=32, verbose=0, validation_spli
```

```
Out[6]: <keras.src.callbacks.history.History at 0x20cfe120d40>
```

```
In [7]: # Predict
nn_pred = (model.predict(X_test) > 0.5).astype(int)

63/63 ━━━━━━━━━━ 0s 5ms/step
```

```
In [8]: # Evaluate
acc = accuracy_score(y_test, nn_pred)
cm = confusion_matrix(y_test, nn_pred)
```

```
In [9]: # --- Final Clean Output ---
print("--- Neural Network Performance ---")
print(f"Accuracy Score: {acc:.4f}")
print("Confusion Matrix:")
```

```
print(cm)
print("Points of Improvement: Adam, Learning Rate, Layer Size, Early Stopping, R
--- Neural Network Performance ---
Accuracy Score: 0.8280
Confusion Matrix:
[[1575  18]
 [ 326  81]]
Points of Improvement: Adam, Learning Rate, Layer Size, Early Stopping, Regulariz
ation, Hyperparameter Tuning.
```

In []: