

```

import pandas as pd

df = pd.read_csv('bank_customer_churn_dataset.csv')
print("Dataset Loaded Successfully!\n")

print("Available columns in dataset:\n", df.columns.tolist())

from sklearn.preprocessing import LabelEncoder

label_encoders = {}
categorical_columns = ['Gender', 'LoanStatus', 'Location', 'Churn']

for col in categorical_columns:
    if col in df.columns:
        le = LabelEncoder()
        df[col] = le.fit_transform(df[col])
        label_encoders[col] = le
    else:
        print(f"Warning: Column '{col}' not found in dataset, skipping encoding.")

if 'Churn' not in df.columns:
    raise ValueError("'Churn' column is missing in the dataset. This is the target variable.")

X = df.drop(columns=["CustomerID", "Churn"], errors='ignore')

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

from sklearn.linear_model import LogisticRegression
model = LogisticRegression(random_state=42)
model.fit(X_train_scaled, y_train)

y_pred = model.predict(X_test_scaled)

output_df = X_test.copy()
if 'CustomerID' in df.columns:
    output_df['CustomerID'] = df.loc[X_test.index, 'CustomerID']
if 'Age' in df.columns:
    output_df['Age'] = df.loc[X_test.index, 'Age']

```

```

if 'MonthlyIncome' in df.columns:
    output_df['MonthlyIncome'] = df.loc[X_test.index, 'MonthlyIncome']
else:
    print("'MonthlyIncome' column not found — skipping.")
if 'LoanStatus' in df.columns:
    output_df['LoanStatus'] = df.loc[X_test.index, 'LoanStatus']
if 'Gender' in df.columns:
    output_df['Gender'] = df.loc[X_test.index, 'Gender']
if 'Tenure' in df.columns:
    output_df['Tenure'] = df.loc[X_test.index, 'Tenure']

output_df['Predicted Churn'] = y_pred
output_df['Predicted Churn'] = output_df['Predicted Churn'].apply(lambda x: 'Churned' if x == 1
else 'Non-churned')

columns_to_display = ['CustomerID', 'Age', 'MonthlyIncome', 'LoanStatus', 'Gender', 'Tenure',
'Predicted Churn']
existing_columns = [col for col in columns_to_display if col in output_df.columns]

output_df = output_df[existing_columns]

print("\n === Customer Churn Prediction Results ===")
print(output_df)

from sklearn.metrics import classification_report, accuracy_score

print("\n Model Evaluation:")
print(classification_report(y_test, y_pred))
print("Accuracy Score:", accuracy_score(y_test, y_pred))

```