```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

```
# Load dataset
df = pd.read_csv('WA_Fn-UseC_-Telco-Customer-Churn.csv')

# Show first 5 rows
df.head()
```

| | customerID | gender | SeniorCitizen | Partner | Dependents | tenure | PhoneService | MultipleLines | InternetService | OnlineSecurity | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 7590-VHVEG | Female | 0 | Yes | No | 1 | No | No phone service | DSL | No | ... |
| **1** | 5575-GNVDE | Male | 0 | No | No | 34 | Yes | No | DSL | Yes | ... |
| **2** | 3668-QPYBK | Male | 0 | No | No | 2 | Yes | No | DSL | Yes | ... |
| **3** | 7795-CFOCW | Male | 0 | No | No | 45 | No | No phone service | DSL | Yes | ... |
| **4** | 9237-HQITU | Female | 0 | No | No | 2 | Yes | No | Fiber optic | No | ... |

5 rows × 21 columns

```
# Remove missing rows
df = df.dropna()

# Convert Churn column to 0/1
df['Churn'] = df['Churn'].apply(lambda x: 1 if x=='Yes' else 0)

# Some TotalCharges are strings; convert them
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')
df = df.dropna()

# Select numerical columns
X = df[['tenure','MonthlyCharges','TotalCharges']]
y = df['Churn']
```

```
print(X.head())
print(y.head())
```

```
   tenure  MonthlyCharges  TotalCharges
0       1           29.85         29.85
1      34           56.95       1889.50
2       2           53.85        108.15
3      45           42.30       1840.75
4       2           70.70        151.65
0    0
1    0
2    1
3    0
4    1
Name: Churn, dtype: int64
```

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

```
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)
```

```
LogisticRegression
LogisticRegression(max_ite
```

What can I help you build?

```
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy of the model:", accuracy)
```

```
Accuracy of the model: 0.7796730632551528
```

```
sample = X_test.head(5)
predictions = model.predict(sample)
print("Predictions (0=No Churn, 1=Churn):", predictions.tolist())
```

```
Predictions (0=No Churn, 1=Churn): [0, 0, 1, 0, 0]
```