Logistic Regression

Use the dataset, perform necessary pre-processing and build a logistic regression model. divide the train data itself into 70-30 ratio and print the performance metrics

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
```

```
df = pd.read_csv("/content/telecom_customer_churn.csv")
```

```
df
```

	Customer ID	Gender	Age	Married	Number of Dependents	City	Zip Code	Latitude	Long
0	0002-ORFBO	Female	37	Yes	0	Frazier Park	93225	34.827662	-118.
1	0003-MKNFE	Male	46	No	0	Glendale	91206	34.162515	-118.
2	0004-TLHLJ	Male	50	No	0	Costa Mesa	92627	33.645672	-117.
3	0011-IGKFF	Male	78	Yes	0	Martinez	94553	38.014457	-122.
4	0013-EXCHZ	Female	75	Yes	0	Camarillo	93010	34.227846	-119.
...	...	...	...	...	...	...	...	...	
6584	9986-BONCE	Female	36	No	0	Fallbrook	92028	33.362575	-117.
6585	9987-LUTYD	Female	20	No	0	La Mesa	91941	32.759327	-116.
6586	9992-RRAMN	Male	40	Yes	0	Riverbank	95367	37.734971	-120.
6587	9993-LHIEB	Male	21	Yes	0	Solana Beach	92075	33.001813	-117.
6588	9995-HOTOH	Male	36	Yes	0	Sierra City	96125	39.600599	-120.

6589 rows × 38 columns

```
df = df.drop(columns=["Customer ID", "City", "Zip Code","Longitude","Latitude", "Churn Category","Churn Reason"])
```

```
df.isnull().sum()
```

```
Gender                                0
Age                                   0
Married                               0
Number of Dependents                  0
Number of Referrals                   0
Tenure in Months                      0
Offer                                 0
Phone Service                         0
Avg Monthly Long Distance Charges     644
Multiple Lines                        644
Internet Service                      0
Internet Type                         1344
Avg Monthly GB Download               1344
Online Security                       1344
Online Backup                         1344
Device Protection Plan                1344
Premium Tech Support                  1344
Streaming TV                          1344
Streaming Movies                      1344
```

```
Streaming Music                    1344
Unlimited Data                     1344
Contract                              0
Paperless Billing                     0
Payment Method                        0
Monthly Charge                        0
Total Charges                         0
Total Refunds                         0
Total Extra Data Charges              0
Total Long Distance Charges           0
Total Revenue                         0
Customer Status                       0
dtype: int64
```

```
df["Avg Monthly Long Distance Charges"] = df["Avg Monthly Long Distance Charges"].fillna(0)
```

```
df["Avg Monthly GB Download"] = df["Avg Monthly GB Download"].fillna(0)
```

```
df["Multiple Lines"] = df["Multiple Lines"].fillna("No Phone")
```

```
df["Internet Type"] = df["Internet Type"].fillna("None")
```

```
na_cols = ["Online Security", "Online Backup", "Device Protection Plan", "Premium Tech Support", "Streaming TV", "Stream
for column in na_cols:
  df[column] = df[column].fillna("No")
```

```
df.isnull().sum()
```

```
Gender                             0
Age                                0
Married                            0
Number of Dependents               0
Number of Referrals                0
Tenure in Months                   0
Offer                              0
Phone Service                      0
Avg Monthly Long Distance Charges  0
Multiple Lines                     0
Internet Service                   0
Internet Type                      0
Avg Monthly GB Download            0
Online Security                    0
Online Backup                      0
Device Protection Plan             0
Premium Tech Support               0
Streaming TV                       0
Streaming Movies                   0
Streaming Music                    0
Unlimited Data                     0
Contract                           0
Paperless Billing                  0
Payment Method                     0
Monthly Charge                     0
Total Charges                      0
Total Refunds                      0
Total Extra Data Charges           0
Total Long Distance Charges        0
Total Revenue                      0
Customer Status                    0
dtype: int64
```

```
df.shape
```

```
(6589, 31)
```

```
from sklearn.preprocessing import LabelEncoder
```

```
le = LabelEncoder()

for col in df.columns[1:]:
    if df[col].dtype == 'object':
        if len(list(df[col].unique())) <= 2:
            le.fit(df[col])
            df[col] = le.transform(df[col])
```

```
df['Gender'] = [1 if each == 'Female' else 0 for each in df['Gender']]
```

```
def encode_data(dataframe):
    if dataframe.dtype == "object":
        dataframe = LabelEncoder().fit_transform(dataframe)
    return dataframe

data = df.apply(lambda x: encode_data(x))
data.head()
```

	Gender	Age	Married	Number of Dependents	Number of Referrals	Tenure in Months	Offer	Phone Service	Avg Monthly Long Distance Charges
0	1	37	1	0	2	9	0	1	42.39
1	0	46	0	0	0	9	0	1	10.69
2	0	50	0	0	0	4	5	1	33.65
3	0	78	1	0	1	13	4	1	27.82
4	1	75	1	0	3	3	0	1	7.38

5 rows × 31 columns

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6589 entries, 0 to 6588
Data columns (total 31 columns):
 #   Column                             Non-Null Count  Dtype
---  ------                             --------------  -----
 0   Gender                             6589 non-null   int64
 1   Age                                6589 non-null   int64
 2   Married                            6589 non-null   int64
 3   Number of Dependents               6589 non-null   int64
 4   Number of Referrals                6589 non-null   int64
 5   Tenure in Months                   6589 non-null   int64
 6   Offer                              6589 non-null   int64
 7   Phone Service                      6589 non-null   int64
 8   Avg Monthly Long Distance Charges  6589 non-null   float64
 9   Multiple Lines                     6589 non-null   int64
 10  Internet Service                   6589 non-null   int64
 11  Internet Type                      6589 non-null   int64
 12  Avg Monthly GB Download            6589 non-null   float64
 13  Online Security                    6589 non-null   int64
 14  Online Backup                      6589 non-null   int64
 15  Device Protection Plan             6589 non-null   int64
 16  Premium Tech Support               6589 non-null   int64
 17  Streaming TV                       6589 non-null   int64
 18  Streaming Movies                   6589 non-null   int64
 19  Streaming Music                    6589 non-null   int64
 20  Unlimited Data                     6589 non-null   int64
 21  Contract                           6589 non-null   int64
 22  Paperless Billing                  6589 non-null   int64
 23  Payment Method                     6589 non-null   int64
 24  Monthly Charge                     6589 non-null   float64
 25  Total Charges                      6589 non-null   float64
 26  Total Refunds                      6589 non-null   float64
 27  Total Extra Data Charges           6589 non-null   int64
 28  Total Long Distance Charges        6589 non-null   float64
 29  Total Revenue                      6589 non-null   float64
 30  Customer Status                    6589 non-null   int64
dtypes: float64(7), int64(24)
memory usage: 1.6 MB
```

```
X = data.drop(['Customer Status'], axis=1)
y = data.loc[:, 'Customer Status'].values
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

```
X_train.shape
```

```
(4612, 30)
```

```
from sklearn.preprocessing import StandardScaler
```

```
cols = ["Tenure in Months","Avg Monthly Long Distance Charges","Avg Monthly GB Download","Monthly Charge","Total Charges
```

```
scaler = StandardScaler()
X_train[cols] = scaler.fit_transform(X_train[cols])
X_test[cols] = scaler.fit_transform(X_test[cols])
```

```
model = LogisticRegression()
```

```
model.fit(X_train, y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458: ConvergenceWarning: lbfgs failed to c
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

▾ LogisticRegression

LogisticRegression()

```
y_pred = model.predict(X_test)
```

```
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1-score:", f1)
```

```
Accuracy: 0.821446636317653
Precision: 0.8690476190476191
Recall: 0.8820184790334044
F1-score: 0.875485008818342
```