MACHINE LEARNING LAB

EXERCISE 3

Aim :

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

Algorithm :

1. Load the dataset.
2. Perform necessary pre-processing steps (e.g., handle missing values, encode categorical variables).
3. Split the dataset into training and testing sets with a 70-30 ratio.
4. Build a logistic regression model using the training data.
5. Evaluate the model using performance metrics (e.g., accuracy, precision, recall, F1-score) on the testing set and print the results.

Code and Output :

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
data=pd.read_csv(r"C:\Users\TEJU\Downloads\telecom_customer_churn (1).csv")
```

In [3]:
```
from sklearn import preprocessing
```

In [4]:
```
data.head()
```

Out[4]:

	Customer ID	Gender	Age	Married	Number of Dependents	City	Zip Code	Latitude	Longitude	Number of Referrals
0	0002-ORFBO	Female	37	Yes	0	Frazier Park	93225	34.827662	-118.999073	2
1	0003-MKNFE	Male	46	No	0	Glendale	91206	34.162515	-118.203869	0
2	0004-TLHLJ	Male	50	No	0	Costa Mesa	92627	33.645672	-117.922613	0
3	0011-IGKFF	Male	78	Yes	0	Martinez	94553	38.014457	-122.115432	1
4	0013-	Female	75	Yes	0	Camarillo	93010	34.227846	-119.079903	3

	Customer ID	Gender	Age	Married	Number of Dependents	City	Zip Code	Latitude	Longitude	Number of Referrals
	EXCHZ									

5 rows × 38 columns

In [5]:
```
data=data.drop(['Customer ID', 'City', 'Zip Code', 'Latitude', 'Longitude', 'Churn C
data.head()
```

Out[5]:

	Gender	Age	Married	Number of Dependents	Number of Referrals	Tenure in Months	Offer	Phone Service	Avg Monthly Long Distance Charges	Multiple Lines	...	Cc
0	Female	37	Yes	0	2	9	None	Yes	42.39	No	...	Or
1	Male	46	No	0	0	9	None	Yes	10.69	Yes	...	N
2	Male	50	No	0	0	4	Offer E	Yes	33.65	No	...	N
3	Male	78	Yes	0	1	13	Offer D	Yes	27.82	No	...	N
4	Female	75	Yes	0	3	3	None	Yes	7.38	No	...	N

5 rows × 31 columns

In [6]:
```
data.shape
```

Out[6]:
```
(6589, 31)
```

In [7]:
```
data['Avg Monthly Long Distance Charges'] = data['Avg Monthly Long Distance Charges'
data['Avg Monthly GB Download'] = data['Avg Monthly GB Download'].fillna(0)
data['Multiple Lines'] = data['Multiple Lines'].fillna('No')
```

In [8]:
```
cols = ['Internet Type', 'Online Security', 'Online Backup', 'Device Protection Plan
data[cols] = data[cols].fillna('No')
data.head()
```

Out[8]:

	Gender	Age	Married	Number of Dependents	Number of Referrals	Tenure in Months	Offer	Phone Service	Avg Monthly Long Distance Charges	Multiple Lines	...	Co
0	Female	37	Yes	0	2	9	None	Yes	42.39	No	...	Or
1	Male	46	No	0	0	9	None	Yes	10.69	Yes	...	N
2	Male	50	No	0	0	4	Offer E	Yes	33.65	No	...	N
3	Male	78	Yes	0	1	13	Offer D	Yes	27.82	No	...	N
4	Female	75	Yes	0	3	3	None	Yes	7.38	No	...	N

5 rows × 31 columns

In [9]:

```
obj = list(data.select_dtypes(include=['object']).columns)
data[obj]
label_encoder = preprocessing.LabelEncoder()
for col in obj:
    data[col] = label_encoder.fit_transform(data[col])
data.head()
```

Out[9]:

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

5 rows × 31 columns

In [10]:

```
X = data.iloc[:,:-1].values
Y = data.iloc[:,-1].values
```

In [11]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_stat
```

In [12]:
```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression(random_state=42, max_iter=5000)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
print('Y_pred', y_pred)
print('Y_test', y_test)
```

```
Y_pred [0 1 1 ... 0 1 0]
Y_test [0 1 1 ... 1 1 0]
C:\Users\TEJU\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:469: Conv
ergenceWarning: lbfgs failed to converge (status=1):
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(
```

In [13]:
```
from sklearn.metrics import accuracy_score,confusion_matrix

accuracy=accuracy_score(y_test,y_pred)*100
print("Accuracy of the model is {:.2f}".format(accuracy))
```

```
Accuracy of the model is 84.52
```

In [14]:
```
from sklearn.metrics import f1_score
f1=f1_score(y_test,y_pred,average="macro")*100
print("F1 score of the model is {:.2f}".format(f1))
```

```
F1 score of the model is 80.94
```

In [15]:
```
from sklearn.metrics import recall_score
recall=recall_score(y_test,y_pred,average="macro")*100
print("Recall score of the model is {:.2f}".format(recall))
```

```
Recall score of the model is 80.62
```

In [16]:
```
from sklearn.metrics import precision_score
precision=precision_score(y_test,y_pred,average="macro")*100
print("Precision score of the model is {:.2f}".format(precision))
```

```
Precision score of the model is 81.29
```

Result :

Therefore, we were successfully able to build a logistic regression model for the given dataset and compute the performance metrics of the same.