Machine Learning Model for XYZ Company Churn Analysis using Logistic Regression

Importing the Necessary Modules

In [115]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.impute import SimpleImputer
import warnings
warnings.filterwarnings("ignore")
```

Loading the Dataset

In [116]:
```
data = pd.read_csv('data.csv')
data
```

Out[116]:

	year	customer_id	phone_no	gender	age	no_of_days_subscribed	multi_screen	mail_su
0	2015	100198	409-8743	Female	36	62	no	
1	2015	100643	340-5930	Female	39	149	no	
2	2015	100756	372-3750	Female	65	126	no	
3	2015	101595	331-4902	Female	24	131	no	
4	2015	101653	351-8398	Female	40	191	no	
...	...	...	...	...	...	...	...	
1995	2015	997132	385-7387	Female	54	75	no	
1996	2015	998086	383-9255	Male	45	127	no	
1997	2015	998474	353-2080	NaN	53	94	no	
1998	2015	998934	359-7788	Male	40	94	no	
1999	2015	999961	414-1496	Male	37	73	no	

2000 rows × 16 columns

Imputation of Missing Values

In [117]:
```
imp_most_freq = SimpleImputer(missing_values=np.nan,strategy='most_frequ
data.gender = pd.DataFrame(imp_most_freq.fit_transform(np.array(data.ger
data.maximum_days_inactive = pd.DataFrame(imp_most_freq.fit_transform(np
data.churn = pd.DataFrame(imp_most_freq.fit_transform(np.array(data.chur
print(data.gender.unique())
print()
print(data.gender.value_counts())
```

```
['Female' 'Male']

Male      1077
Female     923
Name: gender, dtype: int64
```

Feature Selection

In [118]:
```
sns.pairplot(data)
```

Out[118]:
```
<seaborn.axisgrid.PairGrid at 0x7f78fe208a90>
```

In [119]:
```
data = data.drop(['year','customer_id','phone_no'],axis=1)
data
```

Out[119]:

	gender	age	no_of_days_subscribed	multi_screen	mail_subscribed	weekly_mins_watched
0	Female	36	62	no	no	148.35
1	Female	39	149	no	no	294.45
2	Female	65	126	no	no	87.30
3	Female	24	131	no	yes	321.30
4	Female	40	191	no	no	243.00
...	...	...	...	...	...	...
1995	Female	54	75	no	yes	182.25
1996	Male	45	127	no	no	273.45
1997	Male	53	94	no	no	128.85
1998	Male	40	94	no	no	178.05
1999	Male	37	73	no	no	326.70

2000 rows × 13 columns

All I Love is LabelEncoding

In [120]:
```
le = LabelEncoder()
le.fit(['Male','Female'])
print({k:v for k,v in zip(le.classes_,le.transform(['Female','Male'])) }
data.gender = le.fit_transform(data.gender)
data.churn = data.churn.astype("int64")
```

```
{'Female': 0, 'Male': 1}
```

In [121]:
```
le.fit(['no','yes'])
data.multi_screen= pd.DataFrame(le.fit_transform(data.multi_screen))
data.mail_subscribed = pd.DataFrame(le.fit_transform(data.mail_subscribe
data
```

Out[121]:

	gender	age	no_of_days_subscribed	multi_screen	mail_subscribed	weekly_mins_watched
0	0	36	62	0	0	148.35
1	0	39	149	0	0	294.45
2	0	65	126	0	0	87.30
3	0	24	131	0	1	321.30
4	0	40	191	0	0	243.00
...	...	...	...	...	...	...
1995	0	54	75	0	1	182.25
1996	1	45	127	0	0	273.45
1997	1	53	94	0	0	128.85
1998	1	40	94	0	0	178.05
1999	1	37	73	0	0	326.70

2000 rows × 13 columns

In [122]:
```
x = data.drop('churn',axis=1)
y = data.churn
```

In [123]:
```
from sklearn.model_selection import train_test_split as tts
xtr,xte,ytr,yte = tts(x,y,train_size=0.8,random_state=42)
```

Model Building

In [124]:
```
from sklearn.linear_model import LogisticRegression
logr = LogisticRegression(random_state=42).fit(x,y)
ypred =logr.predict(xte)
```

Accuracy

In [125]:
```
from sklearn.metrics import accuracy_score
print("Accuracy of Model is:",accuracy_score(yte,ypred)*100,"%")
```

```
Accuracy of Model is: 88.0 %
```