Name : Dion D Rodrigues Assignment.no: 01

Roll.no : 56 Class : FYMCA B Subject : ML Lab Batch : B3

 # a) IMPORTING LIBRARIES

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import OneHotEncoder
from sklearn.model_selection import train_test_split
from sklearn.impute import SimpleImputer
```

#b) **IMPORTING LIBRARIES**

```
data= pd.read_csv('C:/Users/Admin/Desktop/ML/Assignment_1/Online_shopper.csv')
print(data)
```

	Region	Age	Income	Online Shopper
0	India	49.0	86400.0	No
1	Brazil	32.0	57600.0	Yes
2	USA	35.0	64800.0	No
3	Brazil	43.0	73200.0	No
4	USA	45.0	NaN	Yes
5	India	40.0	69600.0	Yes
6	Brazil	NaN	62400.0	No
7	India	53.0	94800.0	Yes
8	USA	55.0	99600.0	No
9	India	42.0	80400.0	Yes

c) **IDENTIFYING AND HANDLING MISSING DATA**

#identifying

```
print (data.isnull())

print(data.isnull().sum())
```

```
   Region    Age  Income  Online Shopper
0   False  False   False           False
1   False  False   False           False
2   False  False   False           False
3   False  False   False           False
4   False  False    True           False
5   False  False   False           False
6   False   True   False           False
7   False  False   False           False
8   False  False   False           False
9   False  False   False           False
Region            0
Age               1
Income            1
Online Shopper    0
dtype: int64
```

#handling (FILL WITH THE MODE OF THE COLUMN)

```
data['Age']=data['Age'].fillna(data['Age'].mode())

print(data)
```

```
   Region   Age    Income Online Shopper
0   India  49.0  86400.0             No
1  Brazil  32.0  57600.0            Yes
2     USA  35.0  64800.0             No
3  Brazil  43.0  73200.0             No
4     USA  45.0      NaN            Yes
5   India  40.0  69600.0            Yes
6  Brazil  49.0  62400.0             No
7   India  53.0  94800.0            Yes
8     USA  55.0  99600.0             No
9   India  42.0  80400.0            Yes
```

#handling (FILL WITH THE MEAN OF THE COLUMN)

```
data['Income']=data['Income'].fillna(data['Income'].mean())

print(data)
```

```
   Region   Age         Income Online Shopper
0   India  49.0  86400.000000             No
1  Brazil  32.0  57600.000000            Yes
2     USA  35.0  64800.000000             No
3  Brazil  43.0  73200.000000             No
4     USA  45.0  76533.333333            Yes
5   India  40.0  69600.000000            Yes
6  Brazil  49.0  62400.000000             No
7   India  53.0  94800.000000            Yes
8     USA  55.0  99600.000000             No
9   India  42.0  80400.000000            Yes
```

 # Verify if missing data is handled

```
print(data.isnull().sum())
```

```
Region            0
Age               0
Income            0
Online Shopper    0
dtype: int64
```

#d) IDENTIFYING AND HANDLING CATEGORICAL DATAZ

#identify categorial columns

```
categorical_columns = data.select_dtypes(include=[np.number]).columns

print(categorical_columns)
```

```
Index(['Age', 'Income'], dtype='object')
```

#handling missing data by replacing with the mode (for categorical columns)

```
imputer = SimpleImputer (strategy='most_frequent')

data[categorical_columns]=imputer.fit_transform(data[categorical_columns])
```

#verify if missing data is handled

```
print(data.isnull().sum())
```

```
Region            0
Age               0
Income            0
Online Shopper    0
dtype: int64
```

#Apply OneHotEnchoder to Categorical columns

```
encoder = OneHotEncoder(sparse_output=False,drop='first')

encoded_data = encoder.fit_transform(data[categorical_columns])
```

 # Getting the names of encoded columns

```
encoded_columns=encoder.get_feature_names_out(categorical_columns)
```

#Drop original categorical columns and concatenate encoded columns

```
data =data.drop(categorical_columns,axis=1)

data=pd.concat([data,pd.DataFrame (encoded_data,columns=encoded_columns)],axis=1)
```

#Assuming 'Online shopping ' is the target variable, adjust if different

```
x=data.drop('Online Shopper',axis=1)

y=data['Online Shopper']
```

#e) SPLIT THE DATA INTO TRAINING AND TESTING SETS (e.g., 80% train,20% test)

```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.2,random_state=42)


print("Traning set Shape : ",x_train.shape,y_train.shape)

print("Testing set Shape : ",x_test.shape, y_test.shape)
```

```
Traning set Shape :  (8, 18) (8,)
Testing set Shape :  (2, 18) (2,)
```