In [5]:
```
#5

from google.colab import files
uploaded = files.upload()

import numpy as np
import pandas as pd
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import LabelEncoder, StandardScaler, MinMaxScaler

filename = list(uploaded.keys())[0]
df = pd.read_csv(filename)

print("Dataset Info:")
print(df.info())
print("\nFirst 5 Rows:")
print(df.head())

print("\nChecking missing values before imputation:")
print(df.isnull().sum())

imputer = SimpleImputer(strategy='mean')
df_imputed = pd.DataFrame(imputer.fit_transform(df.select_dtypes(include=[np.r
                          columns=df.select_dtypes(include=[np.number]).columr

print("\nAfter Imputation (numeric columns):")
print(df_imputed.head())

label_enc = LabelEncoder()
df_encoded = df.copy()
for col in df.select_dtypes(include=['object']).columns:
      df_encoded[col] = label_enc.fit_transform(df[col].astype(str))

print("\nAfter Label Encoding:")
print(df_encoded.head())

final_set = df_encoded.values

print("\nFinal Dataset (as NumPy array):")
print(final_set[:5])

mms = MinMaxScaler(feature_range=(0,1))
mms.fit(final_set)
feat_minmax_scaler = mms.transform(final_set)

print("\nAfter Min-Max Scaling:")
print(pd.DataFrame(feat_minmax_scaler).head())

print("\nScaling Done! Each feature is now between 0 and 1.")
print("Shape of final scaled data:", feat_minmax_scaler.shape)
```

Browse...

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving pre_process_datasample.csv to pre_process_datasample (4).csv
Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Country    10 non-null     object
 1   Age        9 non-null      float64
 2   Salary     9 non-null      float64
 3   Purchased  10 non-null     object
dtypes: float64(2), object(2)
memory usage: 452.0+ bytes
None

First 5 Rows:
   Country   Age   Salary Purchased
0   France  44.0  72000.0        No
1    Spain  27.0  48000.0       Yes
2  Germany  30.0  54000.0        No
3    Spain  38.0  61000.0        No
4  Germany  40.0      NaN       Yes

Checking missing values before imputation:
Country      0
Age          1
Salary       1
Purchased    0
dtype: int64

After Imputation (numeric columns):
    Age        Salary
0  44.0  72000.000000
1  27.0  48000.000000
2  30.0  54000.000000
3  38.0  61000.000000
4  40.0  63777.777778

After Label Encoding:
   Country   Age   Salary  Purchased
0        0  44.0  72000.0          0
1        2  27.0  48000.0          1
2        1  30.0  54000.0          0
3        2  38.0  61000.0          0
4        1  40.0      NaN          1

Final Dataset (as NumPy array):
[[0.0e+00 4.4e+01 7.2e+04 0.0e+00]
 [2.0e+00 2.7e+01 4.8e+04 1.0e+00]
 [1.0e+00 3.0e+01 5.4e+04 0.0e+00]
 [2.0e+00 3.8e+01 6.1e+04 0.0e+00]
 [1.0e+00 4.0e+01     nan 1.0e+00]]

After Min-Max Scaling:
```

```
     0         1         2    3
0  0.0  0.739130  0.685714  0.0
1  1.0  0.000000  0.000000  1.0
2  0.5  0.130435  0.171429  0.0
3  1.0  0.478261  0.371429  0.0
4  0.5  0.565217       NaN  1.0

Scaling Done! Each feature is now between 0 and 1.
Shape of final scaled data: (10, 4)
```