Practical No.3

Aim:- Feature scaling dummification.

- Apply feature-scaling techniques like standardization and normalization to numerical features.
- Perform feature dummification to convert categorical variables into numerical representations.

Importing Libraries

In [2]:
```
import pandas as pd
from sklearn.preprocessing import StandardScaler,MinMaxScaler,OneHotEncoder
from sklearn.compose import ColumnTransformer
```

Define the data

In [3]:
```
data = {
    'Product':['Apple_Juice','Banana_Smoothie','Orange_Jam','Grape_Jelly',
               'Kiwi_Juice','Mango_Pickle','Pineapple_Sorbet',
               'Strawberry_Yoghurt','Blueberry_Pie','Cherry_Salsa'],
    'Category':['Apple','Banana','Orange','Grape','Kiwi','Mango',
                'Pineapple','Strawberry','Blueberry','Cherry'],
    'Sales':[1200,1700,2200,1400,2000,1000,1500,1800,1300,1600],
    'Cost':[600,850,1100,700,1000,500,750,900,650,800],
    'Profit':[600,850,1100,700,1000,500,750,900,650,800]
}
```

In [4]:
```
#Create a DataFrame
df = pd.DataFrame(data)
#Display the original dataset
print("Original Dataset:")
print(df)
```

```
Original Dataset:
              Product     Category  Sales  Cost  Profit
0         Apple_Juice        Apple   1200   600     600
1     Banana_Smoothie       Banana   1700   850     850
2          Orange_Jam       Orange   2200  1100    1100
3         Grape_Jelly        Grape   1400   700     700
4          Kiwi_Juice         Kiwi   2000  1000    1000
5        Mango_Pickle        Mango   1000   500     500
6    Pineapple_Sorbet    Pineapple   1500   750     750
7  Strawberry_Yoghurt   Strawberry   1800   900     900
8       Blueberry_Pie    Blueberry   1300   650     650
9        Cherry_Salsa       Cherry   1600   800     800
```

Feature Scaling(Standardization and Normalization)

In [5]:
```
numeric_columns = ['Sales','Cost','Profit']
scaler_std = StandardScaler()
scaler_normal = MinMaxScaler()
df_scaled_std = pd.DataFrame(scaler_std.fit_transform(df[numeric_columns]),columns = numeric_columns)
df_scaled_normal = pd.DataFrame(scaler_normal.fit_transform(df[numeric_columns]),columns = numeric_columns)
```

In [6]:
```
df_scaled_std
```

Out[6]:

	Sales	Cost	Profit
0	-1.058873	-1.058873	-1.058873
1	0.372036	0.372036	0.372036
2	1.802946	1.802946	1.802946
3	-0.486509	-0.486509	-0.486509
4	1.230582	1.230582	1.230582
5	-1.631237	-1.631237	-1.631237
6	-0.200327	-0.200327	-0.200327
7	0.658218	0.658218	0.658218
8	-0.772691	-0.772691	-0.772691
9	0.085855	0.085855	0.085855

In [7]:
```
#Combine the scaled numeric features with the categorical values
df_scaled = pd.concat([df_scaled_std,df.drop(numeric_columns,axis =1)],axis = 1)
#Display the dataset after feature scaling
print("\nDataset after Feature Scaling:")
print(df_scaled)
```

```
Dataset after Feature Scaling:
      Sales      Cost    Profit            Product    Category
0 -1.058873 -1.058873 -1.058873        Apple_Juice       Apple
1  0.372036  0.372036  0.372036    Banana_Smoothie      Banana
2  1.802946  1.802946  1.802946         Orange_Jam      Orange
3 -0.486509 -0.486509 -0.486509        Grape_Jelly       Grape
4  1.230582  1.230582  1.230582         Kiwi_Juice        Kiwi
5 -1.631237 -1.631237 -1.631237       Mango_Pickle       Mango
6 -0.200327 -0.200327 -0.200327   Pineapple_Sorbet   Pineapple
7  0.658218  0.658218  0.658218  Strawberry_Yoghurt  Strawberry
8 -0.772691 -0.772691 -0.772691      Blueberry_Pie   Blueberry
9  0.085855  0.085855  0.085855       Cherry_Salsa      Cherry
```

Feature Dummification(Convert Categorical Columns to numerical representation)

In [11]:
```
#Identify categorical columns
categorical_columns = ['Product','Category']

#Create a column transformer for dummification
preprocessor = ColumnTransformer(
 transformers=[
     ('categorical',OneHotEncoder(),categorical_columns)
 ],remainder='passthrough')
```

In [12]:
```
#Apply the column transformer to the dataset
df_dummified = pd.DataFrame(preprocessor.fit_transform(df))
```

In [13]:
```
#Display the dataset after feature dummification
print("\nDataset after Feature Dummification:")
print(df_dummified)
```

```
Dataset after Feature Dummification:
                                                   0
0    (0, 0)\t1.0\n  (0, 10)\t1.0\n  (0, 20)\t1200...
1    (0, 1)\t1.0\n  (0, 11)\t1.0\n  (0, 20)\t1700...
2    (0, 7)\t1.0\n  (0, 17)\t1.0\n  (0, 20)\t2200...
3    (0, 4)\t1.0\n  (0, 14)\t1.0\n  (0, 20)\t1400...
4    (0, 5)\t1.0\n  (0, 15)\t1.0\n  (0, 20)\t2000...
5    (0, 6)\t1.0\n  (0, 16)\t1.0\n  (0, 20)\t1000...
6    (0, 8)\t1.0\n  (0, 18)\t1.0\n  (0, 20)\t1500...
7    (0, 9)\t1.0\n  (0, 19)\t1.0\n  (0, 20)\t1800...
8    (0, 2)\t1.0\n  (0, 12)\t1.0\n  (0, 20)\t1300...
9    (0, 3)\t1.0\n  (0, 13)\t1.0\n  (0, 20)\t1600...
```