**Practical No.05**

* **Features Scaling**

1. **Apply feature-scaling techniques like standardization and normalization to numerical features.**

**Code:**

import pandas as pd

from sklearn.preprocessing import MinMaxScaler, StandardScaler

data=pd.read\_csv('D:\Ramesh VI\Data Science\pract3\Book1.csv')

df = pd.DataFrame(data)

# Display original DataFrame

print("Original DataFrame:")

print(df)

# Normalize the numerical features

scaler = MinMaxScaler()

normalized\_features = scaler.fit\_transform(df)

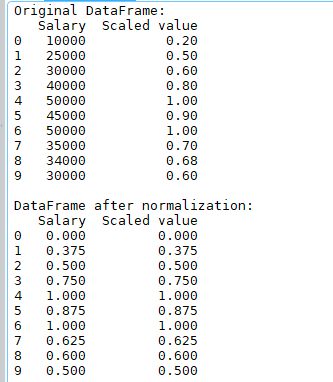
df\_normalized = pd.DataFrame(normalized\_features, columns=df.columns)

# Display DataFrame after normalization

print("\nDataFrame after normalization:")

print(df\_normalized)

**Output:**



# Standardize the numerical features

scaler = StandardScaler()

standardized\_features = scaler.fit\_transform(df)

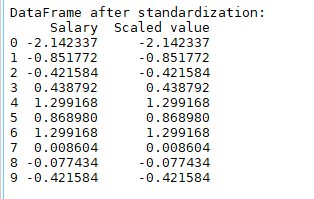
df\_standardized = pd.DataFrame(standardized\_features, columns=df.columns)

# Display DataFrame after standardization

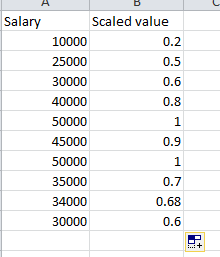
print("\nDataFrame after standardization:")

print(df\_standardized)

**Output:**



**Book1.csv:**



1. **Perform feature dummification to convert cateforical variables into numerical representation.**

**Code :**

import pandas as pd

# Read the CSV file

df = pd.read\_csv('D:/Ramesh VI/Data Science/Mall\_customers.csv')

# Display the original data

print("Original DataFrame:")

print(df)

# Dummify the 'category' variable

dummies = pd.get\_dummies(df['Gender'], prefix='Age')

# Concatenate the dummies with the original dataset

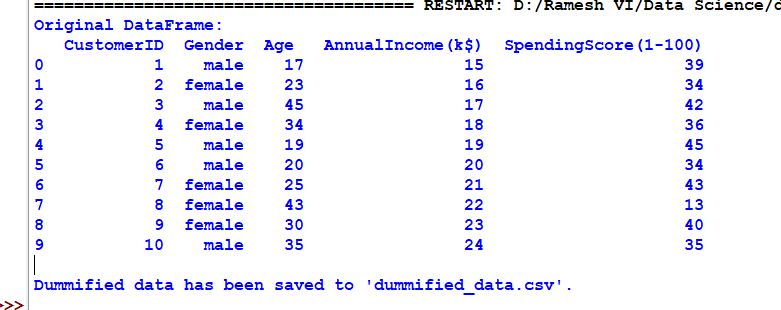
dummified\_data = pd.concat([df, dummies], axis=1)

# Write the dummified data to a new CSV file

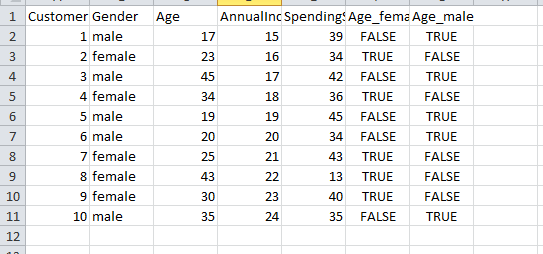
dummified\_data.to\_csv('dummified\_data.csv', index=False)

print("\nDummified data has been saved to 'dummified\_data.csv'.")

**Output:**

****

**dummified\_data.csv:**

****

* **Using LabelBinarizer**

**Code**  
: from sklearn.preprocessing import LabelBinarizer

import pandas as pd

# Example dataset

data = {'Category': ['A', 'B', 'C', 'A', 'C']}

df = pd.DataFrame(data)

# Initialize LabelBinarizer

lb = LabelBinarizer()

# Fit and transform the data

dummy\_data = lb.fit\_transform(df['Category'])

# Convert to DataFrame

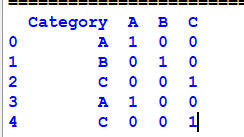
dummy\_df = pd.DataFrame(dummy\_data, columns=lb.classes\_)

# Concatenate dummy DataFrame with original DataFrame

df\_concatenated = pd.concat([df, dummy\_df], axis=1)

print(df\_concatenated)

**Output:**

****