**Practical 1 of DSABDL**

**CODE:**

import pandas as pd

import numpy as np

from sklearn.preprocessing import LabelEncoder

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler

df=sns.load\_dataset('iris')

print("First 5 rows of the dataset: ")

print(df.head())

**First 5 rows of the dataset:**

**sepal\_length sepal\_width petal\_length petal\_width species**

**0 5.1 3.5 1.4 0.2 setosa**

**1 4.9 3.0 1.4 0.2 setosa**

**2 4.7 3.2 1.3 0.2 setosa**

**3 4.6 3.1 1.5 0.2 setosa**

**4 5.0 3.6 1.4 0.2 setosa**

print("Dataset Info: ")

print(df.info())

**Dataset Info:**

**<class 'pandas.core.frame.DataFrame'>**

**RangeIndex: 150 entries, 0 to 149**

**Data columns (total 5 columns):**

**# Column Non-Null Count Dtype**

**--- ------ -------------- -----**

**0 sepal\_length 150 non-null float64**

**1 sepal\_width 150 non-null float64**

**2 petal\_length 150 non-null float64**

**3 petal\_width 150 non-null float64**

**4 species 150 non-null object**

**dtypes: float64(4), object(1)**

**memory usage: 6.0+ KB**

**None**

print("Basic statistics of numerical varibles: ")

print(df.describe())

**Basic statistics of numerical varibles:**

**sepal\_length sepal\_width petal\_length petal\_width**

**count 150.000000 150.000000 150.000000 150.000000**

**mean 5.843333 3.057333 3.758000 1.199333**

**std 0.828066 0.435866 1.765298 0.762238**

**min 4.300000 2.000000 1.000000 0.100000**

**25% 5.100000 2.800000 1.600000 0.300000**

**50% 5.800000 3.000000 4.350000 1.300000**

**75% 6.400000 3.300000 5.100000 1.800000**

**max 7.900000 4.400000 6.900000 2.500000**

print("Missing values in each columns: ")

print(df.isnull().sum())

**Missing values in each columns:**

**sepal\_length 0**

**sepal\_width 0**

**petal\_length 0**

**petal\_width 0**

**species 0**

**dtype: int64**

print("Dimensions of the datafram: ")

print("Number of rows: ",df.shape[0])

print("Number of columns: ",df.shape[1])

**Dimensions of the dataframe:**

**Number of rows: 150**

**Number of columns: 5**

le = LabelEncoder()

df['species\_encoded'] = le.fit\_transform(df['species'])

scaler = StandardScaler()

numerical\_features = ['sepal\_length','sepal\_width','petal\_length','petal\_width']

df\_normalized = pd.DataFrame(scaler.fit\_transform(df[numerical\_features]), columns = numerical\_features)

df\_normalized['species\_encoded'] = df['species\_encoded']

print("First 5 rows of normalized data with encoded categorial varialbe: ")

print(df\_normalized.head())

**First 5 rows of normalized data with encoded categorial varialbe:**

**sepal\_length sepal\_width petal\_length petal\_width species\_encoded**

**0 -0.900681 1.019004 -1.340227 -1.315444 0**

**1 -1.143017 -0.131979 -1.340227 -1.315444 0**

**2 -1.385353 0.328414 -1.397064 -1.315444 0**

**3 -1.506521 0.098217 -1.283389 -1.315444 0**

**4 -1.021849 1.249201 -1.340227 -1.315444 0**

print("Species encoding mapping: ")

for i, species in enumerate(le.classes\_):

print(f"{species}:{i}")

**Species encoding mapping:**

**setosa:0**

**versicolor:1**

**virginica:2**