|  |  |
| --- | --- |
| **EXP NO: 1** | **SETTING UP THE ENVIRONMENT AND PREPROCESSING THE DATA** |

# AIM:

To set up a fully functional machine learning development environment and to perform

data preprocessing operations like handling missing values, encoding categorical variables, feature scaling, and splitting datasets**.**

# ALGORITHM:

1. Install Required Libraries:
   * Install numpy, pandas, matplotlib, seaborn, and scikit-learn using pip.
2. Import Libraries.
3. Load Dataset:
   * Load any dataset (e.g., Titanic or Iris) using pandas.
4. Data Exploration:
   * Use df.info(), df.describe(), df.isnull().sum() to understand the data.
5. Handle Missing Values:
   * Use .fillna() or .dropna() depending on the strategy.
6. Encode Categorical Data:
   * Use pd.get\_dummies() or LabelEncoder.
7. Feature Scaling:
   * Normalize or standardize the numerical features using StandardScaler or MinMaxScaler.
8. Split Dataset:
   * Use train\_test\_split() from sklearn to create training and testing sets.
9. Display the Preprocessed Data.

# CODE:

# 1. Install necessary libraries (if not already installed)

# !pip install numpy pandas matplotlib seaborn scikit-learn

# 2. Import libraries import pandas as pd import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler, LabelEncoder import seaborn as sns

import matplotlib.pyplot as plt

# 3. Load dataset

df = sns.load\_dataset(‘titanic’) # Titanic dataset df.head()

# 4. Explore the dataset print(df.info()) print(df.describe()) print(df.isnull().sum())

# 5. Handle missing values

# Fill age with median, embark\_town with mode df[‘age’].fillna(df[‘age’].median(), inplace=True) df[‘embark\_town’].fillna(df[‘embark\_town’].mode()[0], inplace=True) df.drop(columns=[‘deck’], inplace=True) # too many missing values

# 6. Encode categorical variables

# Convert ‘sex’ and ‘embark\_town’ using LabelEncoder

le = LabelEncoder()

df[‘sex’] = le.fit\_transform(df[‘sex’])

df[‘embark\_town’] = le.fit\_transform(df[‘embark\_town’])

# Drop non-informative or redundant columns

df.drop(columns=[‘embarked’, ‘class’, ‘who’, ‘alive’, ‘adult\_male’, ‘alone’], inplace=True)

# 7. Feature Scaling scaler = StandardScaler()

numerical\_cols = [‘age’, ‘fare’]

df[numerical\_cols] = scaler.fit\_transform(df[numerical\_cols])

# 8. Split dataset

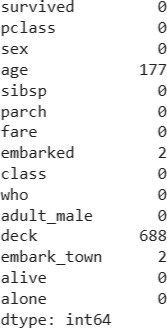
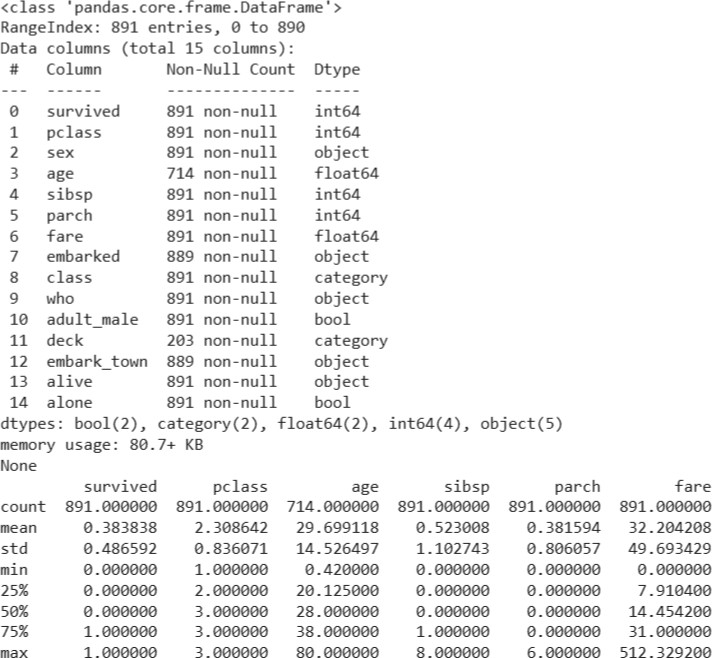
# Define features (X) and label (y)

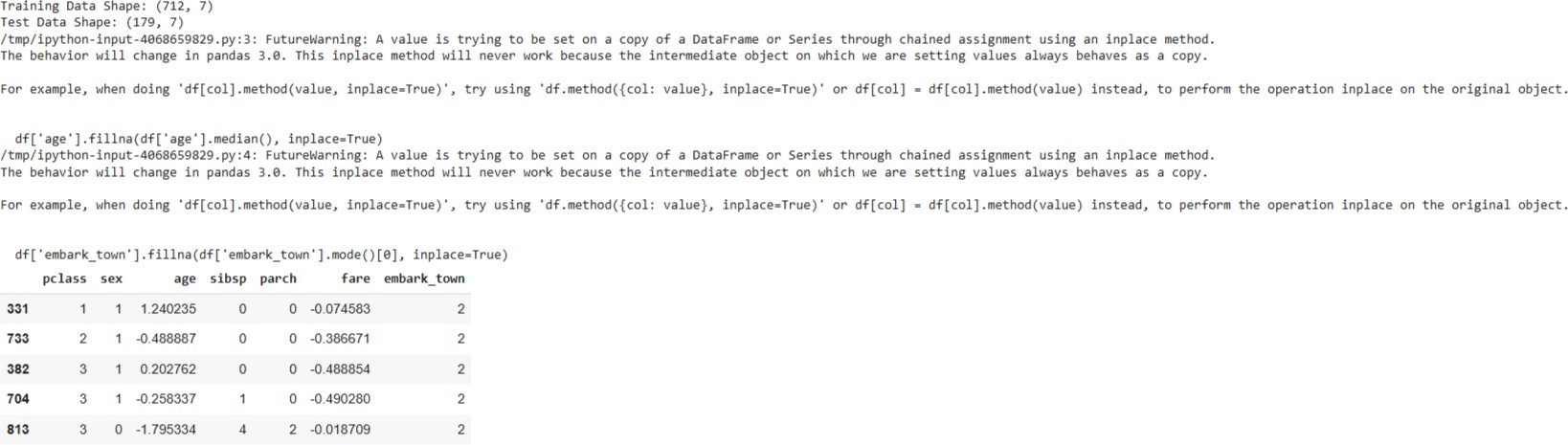
X = df.drop(‘survived’, axis=1) y = df[‘survived’]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# 9. Show final preprocessed data print(“Training Data Shape:”, X\_train.shape) print(“Test Data Shape:”, X\_test.shape) X\_train.head()

# OUTPUT:

****



**RESULT:**

The Python environment was successfully set up and the dataset was pre-processed by handling missing values, encoding categorical data, performing feature scaling, and splitting the data into training and testing sets. The dataset is now ready for model training and analysis.