Assignment No-3

Create own dataset and do simple preprocessing

Dataset Name: Data.CSV (save following data in Excel and save it with .CSV extension)

Country, Age, Salary, Purchased

France, 44, 72000, No

Spain, 27, 48000, Yes

Germany, 30, 54000, No

Spain, 38, 61000, No

Germany, 40, , Yes

France, 35, 58000, Yes

Spain, 52000, No

France, 48, 79000, Yes

Germany, 50, 83000, No

France, 37, 67000, Yes

**Write a program in python to perform following task**

**1. Import Dataset and do the followings:**

**a) Describing the dataset**

**b) Shape of the dataset**

**c) Display first 3 rows from dataset**

import pandas as pd

# Step 1: Load dataset

df = pd.read\_csv("Data.csv")

# Step 2: Display basic info

print("Dataset Description:\n", df.describe(include='all'))

print("\nShape of Dataset:", df.shape)

print("\nFirst 3 rows:\n", df.head(3))

**2. Handling Missing Value: a) Replace missing value of salary,age column with mean of that Column.**

import pandas as pd

# Step 1: Load dataset

df = pd.read\_csv("Data.csv")

# Step 3: Handle missing values (mean for Age and Salary)

df['Age'] = df['Age'].fillna(df['Age'].mean())

df['Salary'] = df['Salary'].fillna(df['Salary'].mean())

print("\nAfter filling missing values:\n", df)

**3. Data.csv have two categorical column (the country column, and the purchased column).**

**a. Apply OneHot coding on Country column.**

**b. Apply Label encoding on purchased column**

**pip install pandas scikit-learn**

import pandas as pd

from sklearn.preprocessing import OneHotEncoder, LabelEncoder

# Step 1: Load dataset

df = pd.read\_csv("Data.csv")

# Step 4: One-Hot Encoding for 'Country'

ohe = OneHotEncoder(sparse\_output=False)

country\_encoded = ohe.fit\_transform(df[['Country']])

country\_df = pd.DataFrame(country\_encoded, columns=ohe.get\_feature\_names\_out(['Country']))

# Merge and drop original 'Country'

df = df.drop('Country', axis=1)

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

# Step 5: Label Encoding for 'Purchased'

le = LabelEncoder()

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

print("\nAfter Encoding:\n", df)

**Set B**

**Write a program in python to perform following task**

**1. Import Dataset from above link.**

**2. Rescaling: Normalised the dataset using MinMaxScaler class**

**3. Standardizing Data (transform them into a standard Gaussian distribution with a mean of 0 and a standard deviation of 1)**

**4. Normalizing Data ( rescale each observation to a length of 1 (a unit norm). For this, use**

**the Normalizer class.)**

**5. Binarizing Data using we use the Binarizer class (Using a binary threshold, it is possible**

**to transform our data by marking the values above it 1 and those equal to or below it, 0)**

import pandas as pd

from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer, Binarizer

# Step 1: Import dataset

url = "http://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv"

df = pd.read\_csv(url, sep=';')

print("Dataset Shape:", df.shape)

print("\nFirst 5 rows:\n", df.head())

# Step 2: Rescaling (MinMaxScaler)

minmax\_scaler = MinMaxScaler()

df\_minmax = pd.DataFrame(minmax\_scaler.fit\_transform(df), columns=df.columns)

print("\nMin-Max Scaled Data (first 5 rows):\n", df\_minmax.head())

# Step 3: Standardizing Data (StandardScaler)

standard\_scaler = StandardScaler()

df\_standardized = pd.DataFrame(standard\_scaler.fit\_transform(df), columns=df.columns)

print("\nStandardized Data (first 5 rows):\n", df\_standardized.head())

# Step 4: Normalizing Data (Normalizer)

normalizer = Normalizer()

df\_normalized = pd.DataFrame(normalizer.fit\_transform(df), columns=df.columns)

print("\nNormalized Data (first 5 rows):\n", df\_normalized.head())

# Step 5: Binarizing Data (Binarizer with threshold = 5.0 for example)

binarizer = Binarizer(threshold=5.0)

df\_binarized = pd.DataFrame(binarizer.fit\_transform(df), columns=df.columns)

print("\nBinarized Data (threshold = 5.0, first 5 rows):\n", df\_binarized.head())

### **What this does:**

1. **Loads dataset** directly from UCI link.
2. **Min-Max Scaling** → All values scaled to **[0,1]**.
3. **Standardization** → Mean 0, standard deviation 1.
4. **Normalization** → Each row has unit length.
5. **Binarization** → Values > threshold become 1, else 0.

**Set C**

**1) Write python code to import the required libraries and load the dataset into a pandas**

**dataframe.**

**2) Display the first five rows of the dataframe.**

**3) Discretized the marks column into five discrete buckets, the labels need to be populated**

**accordingly with five values: Poor, Below\_average, Average, Above\_average, and**

**Excellent. Perform bucketing using the cut () function on the marks column and display the**

**top 10 columns.**

import pandas as pd

# 1) Import dataset from the given GitHub link

url = "https://raw.githubusercontent.com/TrainingByPackt/Data-Science-with-Python/master/Chapter01/Data/Student\_bucketing.csv"

df = pd.read\_csv(url)

# 2) Display the first five rows of the dataframe

print("First five rows of dataset:\n")

print(df.head())

# 3) Discretize 'marks' into 5 buckets

labels = ['Poor', 'Below\_average', 'Average', 'Above\_average', 'Excellent']

df['bucket'] = pd.cut(df['marks'], bins=5, labels=labels)

# Display the top 10 rows with bucket column

print("\nTop 10 rows with bucketing:\n")

print(df.head(10))