CODTECH IT Internship.

TASK ONE : DATA PROCESSING

Ensure its quality and suitability for analysis. This task involves cleaning, transforming, and preparing raw data for AI model training.

Certainly! Let’s first generate a small synthetic data and then we pre-process it a little before running classification on it. Just for this purpose

1. Generate Synthetic Data:

- Numerical features: It can be named “Age,” “Income,” and “Score. ”
- Categorical features: Gender: ‘Male’, ‘Female’; City: ‘New York’, ‘Los Angeles’, ’Chicago’.

Here’s a sample dataset:

Age	Income	Score	Gender	City
30	60000	0.75	Male	New York
25	45000	0.60	Female	Los Angeles
40	80000	0.90	Male	Chicago
…	…	…	…	…

2. Data Preprocessing:

➢ Data Cleaning:

- Delete any repetitiveness, if any exists in your work, after proof reading.
- Don’t eliminate rows, although hedonic channels can be effective when they are able to exert impressive and substantial control over their supply-side communications with manufacturers.
- Correct any inconsistencies.

➢ Data Transformation:

- Scale the features by Min-Max scaling to make them available and rename the variables “Age,” “Income,” and “Score”.
- Perform one-hot encoding on the ‘Gender’ column.
- In the same manner, perform one-hot encoding on “City”.

➢ Feature Engineering:

- Develop a new binary feature, namely “Wealth Index,” that is the result of a combination of two original features: “Age” and “Income”.
- Choose attributes that can be supported based on existing knowledge concerning the domain of application.

➢ Data Splitting:

- Divide the data into training set, validation set and the test set.

CODING SECTION

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler, OneHotEncoder
from sklearn.model_selection import train_test_split

data = {
   "Age": [30, 25, 40],
   "Income": [60000, 45000, 80000],
    "Score": [0.75, 0.60, 0.90],
    "Gender": ["Male", "Female", "Male"],
    "City": ["New York", "Los Angeles", "Chicago"]
}

df = pd.DataFrame(data)

df.drop_duplicates(inplace=True)  # Remove duplicates

df.dropna(inplace=True)  # Remove rows with missing values

scaler = MinMaxScaler()
df[["Age", "Income", "Score"]] = scaler.fit_transform(df[["Age", "Income",
"Score"]])

encoder = OneHotEncoder(sparse=False, drop="first")
encoded_gender = pd.DataFrame(encoder.fit_transform(df[["Gender"]]),
columns=["Gender_Male"])
encoded_city = pd.get_dummies(df["City"], prefix="City")

df_encoded = pd.concat([df, encoded_gender, encoded_city], axis=1)
df_encoded.drop(["Gender", "City"], axis=1, inplace=True)

df_encoded["Wealth_Index"] = df_encoded["Age"] *
df_encoded["Income"]

X = df_encoded.drop("Score", axis=1)
y = df_encoded["Score"]
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2,
random_state=42)

print("Preprocessed dataset:")
print(X_train)
```

OUTPUT

```
   Age    Income     Gender_Male  City_Los Angeles  City_New York
Wealth_Index
0  0.5  0.333333            1             0            1     0.166667
2  1.0  1.000000            1             0            0     1.000000
1  0.0  0.000000            0             1            0     0.000000
```