Task 4: Feature Encoding & Scaling

Objective

To convert categorical features into numerical form (feature encoding) and normalize numerical features (feature scaling) so that the dataset can be used effectively for machine learning models.

Dataset

Adult Income Dataset

(Target: income → <=50K or >50K)

Steps Performed

1. Load Dataset

Copy code

Python

```
import pandas as pd

df = pd.read_csv("adult.csv")
```

2. Handle Missing Values

Copy code

Python

```
df.replace(" ?", pd.NA, inplace=True)

df.dropna(inplace=True)
```

3. Separate Features

Copy code

Python

```
X = df.drop("income", axis=1)

y = df["income"]
```

4. Feature Encoding

Identify Categorical & Numerical Columns

Copy code

Python

```
categorical_cols = X.select_dtypes(include="object").columns
numerical_cols = X.select_dtypes(include=["int64", "float64"]).columns
```

Apply One-Hot Encoding (Categorical Data)

Copy code

Python

```
X_encoded = pd.get_dummies(X, columns=categorical_cols, drop_first=True)
```

✔ Converts text categories into binary numerical columns

✔ Avoids dummy variable trap using drop_first=True

Encode Target Variable

Copy code

Python

```
y = y.map({"<=50K": 0, ">50K": 1})
```

5. Feature Scaling

Apply Standard Scaling

Copy code

Python

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_encoded[numerical_cols] = scaler.fit_transform(X_encoded[numerical_cols])
```

✔ Mean = 0

✔ Standard Deviation = 1

✔ Improves model performance (especially for distance-based algorithms)

6. Final Dataset Check

Copy code

Python

```
print(X_encoded.head())

print(X_encoded.shape)
```

Final Output

All categorical features are numerically encoded

All numerical features are scaled

Dataset is ML-ready

Conclusion

Feature Encoding transformed categorical data into numeric form using One-Hot Encoding, and Feature Scaling normalized numerical values using StandardScaler. The Adult Income dataset is now suitable for machine learning algorithms.