Name : Dipti Bhavsar

File : Data Preprocessing explaination

Queries and there Meaning :

Cell 1 :

```
import pandas as pd

import numpy as np

from sklearn.impute import SimpleImputer

from sklearn.preprocessing import LabelEncoder, StandardScaler
```

Meaning :

This code imports the necessary libraries for data preprocessing. Pandas and NumPy are used for data handling and numerical operations. SimpleImputer handles missing values, LabelEncoder converts categorical data into numbers, and StandardScaler normalizes numerical features for better model performance.

Cell 2:

```
df = pd.read_csv(
   "exoplanet_data.csv",   # your file name
   comment="#",          # skip NASA metadata lines
   engine="python",
   on_bad_lines="skip"
)
# Clean column names
df.columns = df.columns.str.strip().str.lower()
print(df.shape)
print(df.columns.tolist()[:30])
```

Meaning :

This code loads the exoplanet dataset while ignoring metadata lines and corrupted rows. Column names are cleaned by removing extra spaces and converting them to lowercase to avoid errors during processing. The dataset size and a sample of column names are printed to verify successful loading.

Cell 3:

```
df["habitable"] = (
   (df["pl_eqt"] < 300) &
   (df["pl_rade"].between(0.5, 2.0))
).astype(int)
print(df["habitable"].value_counts())
```

Meaning :

This code creates a new binary column called **habitable** based on scientific conditions. Planets with suitable temperature and size are marked as habitable (1), while others are marked as non-habitable (0). The count of each class is printed to check the distribution.

Cell 4:

```
target_col = "habitable"
X = df.drop(columns=[target_col])
y = df[target_col]
```

Meaning:

This code separates the dataset into input features (X) and the target variable (y). The target column **habitable** is removed from the feature set and stored separately for machine learning training.

Cell 5:

```
drop_like = [c for c in X.columns if any(
```

k in c for k in ["name", "hostname", "rowid", "id"]

)]

X = X.drop(columns=drop_like)

Meaning:

This code identifies and removes columns that contain names or IDs. Such identifier columns do not help in predicting habitability and may add unnecessary noise to the model.

Cell 6:

X = X.dropna(axis=1, how="all")

Meaning:

This line removes columns that contain only missing values. Since these columns have no useful information, dropping them simplifies the dataset and improves model efficiency.

Cell 7:

num_cols = X.select_dtypes(include=["int64", "float64"]).columns

cat_cols = X.select_dtypes(include=["object"]).columns

Numerical → Median

X[num_cols] = SimpleImputer(strategy="median").fit_transform(X[num_cols])

Categorical → Most frequent

X[cat_cols] = SimpleImputer(strategy="most_frequent").fit_transform(X[cat_cols])

Meaning :

This code separates numerical and categorical columns. Missing values in numerical columns are filled using the median, while missing values in categorical columns are filled using the most frequent value. This ensures the dataset has no missing values and is ready for model training.

Cell 8:

```
for col in cat_cols:
    le = LabelEncoder()
    X[col] = le.fit_transform(X[col].astype(str))
```

Meaning:

This code converts categorical (text) columns into numerical values using label encoding. This step is required because machine learning models work only with numeric data.

Cell 9:

```
for col in num_cols:
    Q1 = X[col].quantile(0.25)
    Q3 = X[col].quantile(0.75)
    IQR = Q3 - Q1
    lower = Q1 - 1.5 * IQR
    upper = Q3 + 1.5 * IQR
    X[col] = np.clip(X[col], lower, upper)
```

Meaning :

This code detects outliers in numerical columns using the Interquartile Range (IQR) method and limits extreme values within a safe range. This reduces the impact of outliers on the machine learning model.

Cell 10:

```
        scaler = StandardScaler()
        X[num_cols] = scaler.fit_transform(X[num_cols])
```

Meaning :

This code scales numerical features so they have a mean of 0 and a standard deviation of 1. Feature scaling helps machine learning models learn more effectively.

Cell 11:

```
preprocessed_df = pd.concat([X, y], axis=1)

preprocessed_df.to_csv(
  "exoplanet_preprocessed.csv",
  index=False
)

print("File saved as exoplanet_preprocessed.csv")

print(preprocessed_df.shape)
```

Meaning :

This code combines the processed features and the target variable into a single dataset and saves it as a new CSV file. The file is stored without index values and is ready to be used for machine learning model training.