SHERAZ BIN TAHIR

Mail: sherazbintahir@gmail.com

Week 2 – Task 2

Article: Data Preprocessing with Python

Article: Data Preprocessing with Python

(First Step Toward Clean and Reliable Data)

When working with data, one of the biggest lessons you quickly learn is this: raw data is never perfect. It often comes with missing values, duplicates, inconsistent formats, or unwanted noise. Before we can build models or visualize insights, we need to clean and prepare the data. This crucial stage is called data preprocessing.

In this article, I'll break down what data preprocessing is, why it matters, and some common techniques with simple examples.

Why is Preprocessing Important?

Imagine you are analyzing laptop prices, and it may have some problems like

- different format
- Missing values
- Inconsistent Data
- Duplicate

For example

Name	Price	Ram
HP laptop i5	Rs. 79,999	8GB
Dell Inspiron 15	124500	8GB
Lenovo Thinkpad	N/A	16GB

You can already spot the problems:

- Prices are in different formats (Rs. 79,999 vs 124,500)
- RAM values are inconsistent (8GB vs 8 GB)
- One row is missing a price

If you feed this directly into a machine learning model, it will simply fail or give meaningless results. Preprocessing makes sure the data is clean, structured, and ready for analysis.

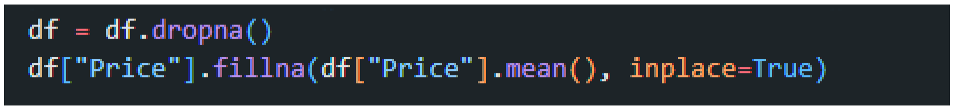
Key Steps in Data Preprocessing

- Handeling Missing Values
- Data cleaning
- Data transformation
- Removing duplicates

1. Handling Missing Values

Missing values are common in real datasets. The options are:

- Remove rows/columns with too many missing values
- Fill in missing values using methods like mean, median, or mode

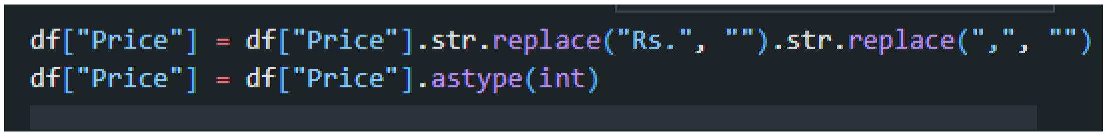


```
df = df.dropna()
df["Price"].fillna(df["Price"].mean(), inplace=True)
```

2. Data Cleaning

This step removes unwanted characters, formatting issues, and inconsistencies.

Example: Cleaning the price column.



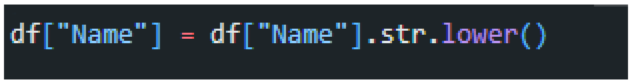
```
df["Price"] = df["Price"].str.replace("Rs.", "").str.replace(",", "")
df["Price"] = df["Price"].astype(int)
```

3. Data Transformation

Transforming data into a consistent format helps models understand it better.

Examples:

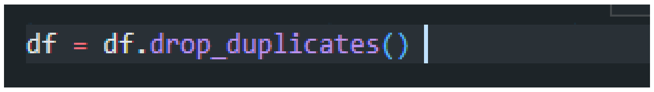
- Converting text to lowercase
- Scaling numeric values
- Encoding categorical values into numbers



```
df["Name"] = df["Name"].str.lower()
```

Removing Duplicates

Datasets often contain repeated rows.



```
df = df.drop_duplicates()
```

Final Clean Dataset

Name	Price	Ram
HP laptop i5	79,999	8GB
Dell Inspiron 15	124500	8GB
Lenovo Thinkpad	99999	16GB

Conclusion

Data preprocessing may feel boring compared to building fancy machine learning models, but it’s the step that decides whether your project succeeds or fails. Think of it as preparing ingredients before cooking: the cleaner and more organized they are, the better the dish will taste.

So, always remember: good data leads to good results.