Data Wrangling in Data Science - Notes

Data Wrangling is the process of cleaning, transforming, and preparing raw data into a usable format for analysis and machine learning. It is a crucial step in any Data Science project because raw data often contains missing values, duplicates, errors, or inconsistent formatting.

Steps in Data Wrangling:

1 1. Handling Missing Values

2 2. Handling Duplicates

3 3. Handling Outliers

4 4. Data Transformation (Scaling, Normalization, Encoding)

5 5. Data Visualization checks

1. Handling Missing Values

Missing values are common in datasets. We can handle them by: - Replacing numeric columns with median, mean, or a fixed value. - Replacing categorical columns with mode (most frequent) or a constant like 'Unknown'.

Example in Python:

```
import pandas as pd df = pd.read_csv('data.csv') # Numeric columns: fill with median
df['Age'] = df['Age'].fillna(df['Age'].median()) # Categorical columns: fill with
mode df['Gender'] = df['Gender'].fillna(df['Gender'].mode()[0])
```

2. Handling Duplicates

Duplicates can affect analysis results. We can remove duplicates using: df = df.drop_duplicates()

3. Handling Outliers

Outliers are extreme values that can distort analysis. Methods to handle outliers: - Using Interquartile Range (IQR) - Using Z-score method - Capping values within a range

4. Data Transformation

Data transformation improves data quality and model performance: - Scaling: StandardScaler, MinMaxScaler - Normalization: Convert values to a common scale - Encoding: Convert categorical values into numbers using Label Encoding or One-Hot Encoding

5. Data Visualization Checks

Visualization helps in understanding data distribution and detecting anomalies: - Histogram: For distribution - Boxplot: For detecting outliers - Scatter plot: For relationship between variables

Final Summary

Data Wrangling ensures that data is clean, consistent, and ready for analysis. Key functions used in Python include: - df.info(), df.describe() - df.fillna(), df.dropna() - df.drop_duplicates() - Visualization: df.hist(), df.plot(kind='box')