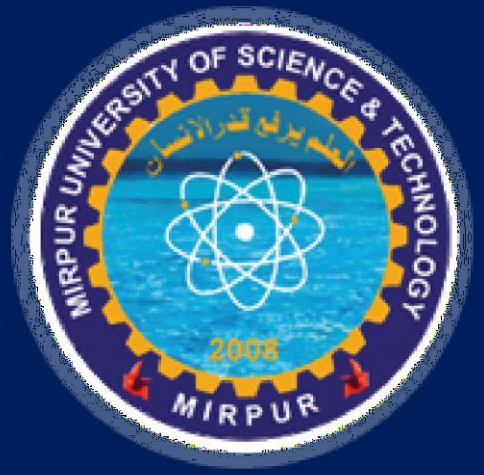


MUST

Wisdom & Virtue

# MIRPUR UNIVERSITY OF SCIENCE AND TECHNOLOGY
# DEPARTMENT OF SOFTWARE ENGINEERING

# Reasons for dirty data and types of dirty data

***Engr. Saman Fatima***
*(Lecturer)*

# Reasons for Dirty Data

Dirty data refers to data that is inaccurate, incomplete, inconsistent, or contains errors or discrepancies. It can be caused by various factors and comes in several different types. Here are some common reasons for dirty data and the types of dirty data:

**Reasons for Dirty Data:**

**Data Entry Errors:** Human errors during data entry, such as typos, incorrect values, or misinterpreted information, can introduce inaccuracies into the data.

**Missing Data:** Some records may have missing values for certain attributes, which can affect data analysis and modeling.

**Duplicate Data:** Duplicate entries for the same entity can lead to inconsistencies and errors in data analysis. This can happen due to data entry mistakes or integration issues.

# CONTD

- **Outliers:** Outliers are data points that deviate significantly from the expected range or pattern. They can skew statistical analysis and modeling results.
- **Data Integration Issues:** When data from multiple sources is combined, integration issues like schema mismatch, conflicting data formats, or data transformation errors can lead to dirty data.
- **Inconsistent Data:** Data from different sources may use different formats, units of measurement, or reference systems. Inconsistencies in data formatting can cause problems when combining or comparing data.

# Types of Dirty Data

- **Inaccurate Data:** Inaccurate data contains incorrect information. This can include incorrect numerical values, misspelled names, or outdated information.
- **Incomplete Data:** Incomplete data lacks necessary information. It may have missing values in certain fields or attributes.
- **Inconsistent Data:** Inconsistent data has variations in format, units, or coding. For example, dates may be recorded in different formats (e.g., MM/DD/YYYY vs. DD/MM/YYYY).
- **Duplicate Data:** Duplicate data consists of multiple copies of the same records or entries. This can occur due to data entry errors or issues during data integration.

# CONTD

**Missing Data:** Missing data occurs when certain values or records are entirely absent from the dataset. This can result from oversight during data collection or storage.

**Data Imbalance:** Data imbalance occurs when one category or class in a dataset is significantly overrepresented or underrepresented compared to others. This can impact the performance of machine learning models.

**Non-Standardized Data:** Data that lacks standardization in terms of units, reference points, or naming conventions can lead to confusion and errors in analysis.

**Temporal Data Issues:** Problems related to the timing and sequencing of data, such as timestamp inaccuracies or data recorded in the wrong order.

# Tips for data cleaning

**Understand Your Data:**

Before you start cleaning, thoroughly understand the data you're working with. This includes its structure, meaning, and context within your project.

**Document the Cleaning Process:**

Keep a record of all the cleaning steps you perform. This documentation can be invaluable for reproducing results and troubleshooting issues.

**Identify and Handle Missing Data:**

Identify missing values in your dataset. Decide whether to remove rows or columns with too many missing values or use imputation techniques to fill in missing data.

Be cautious when imputing data and choose appropriate methods (mean, median, regression, etc.) based on the nature of the data.

# Cond

**Remove Duplicates:**

Identify and remove duplicate records from your dataset. Duplicate data can skew analysis results and waste computational resources.

**Standardize Data:**

Standardize data formats, units of measurement, and naming conventions to ensure consistency. For example, convert all dates to a consistent format and ensure that categorical data is labeled consistently.

**Check for Inaccurate Data:**

Identify and correct inaccuracies, such as outliers or values that fall outside expected ranges. Consider using statistical methods or domain knowledge to detect anomalies.

**Handle Outliers:**

Decide how to handle outliers—whether to remove them, transform them, or keep them as-is. Your choice should align with the goals of your analysis.

# Contd

- **Address Inconsistent Data:**
  - Look for inconsistent data, such as typos or variations in capitalization. Consider using text processing techniques or tools to standardize text data.
- **Validate Data Integrity:**
  - Check for data integrity issues, such as data tampering or corruption. Implement security measures to protect against these issues, especially in sensitive datasets.
- **Perform Data Validation:**
  - Implement data validation checks to ensure that data adheres to predefined rules and constraints. For example, validate that email addresses follow a specific format.
- **Deal with Encoding Issues:**
  - Be aware of character encoding problems, especially when working with text data from diverse sources. Convert data to a consistent encoding if needed.

# Contd

- **Use Data Profiling Tools:**
  - Consider using data profiling tools or libraries to automatically analyze your data and detect potential issues.
- **Visualize Data:**
  - Create visualizations to explore and spot anomalies in your data. Visualization tools can be powerful aids in identifying data issues.
- **Domain Knowledge Matters:**
  - Leverage domain expertise to understand what data values are valid and reasonable for your specific application.
- **Test and Validate:**
  - After cleaning, thoroughly test your data to ensure it's suitable for your analysis or modeling tasks. Validate the cleaned data against known benchmarks or standards.

# Contd

**Iterate and Refine:**

Data cleaning is often an iterative process. As you analyze the cleaned data, you may discover additional issues that require further cleaning and refinement.

**Backup Original Data:**

Before making any significant changes during data cleaning, make sure to back up the original dataset to preserve the raw data for reference.

**Automate Where Possible:**

Use data cleaning scripts or tools to automate repetitive cleaning tasks, especially when dealing with large datasets.

**Collaborate and Seek Feedback:**

If you're working in a team, collaborate with colleagues and seek feedback on your cleaning process to ensure consistency and thoroughness.

**Maintain Data Quality:**

Consider implementing data governance practices to maintain data quality over time. Regularly monitor and validate your data as it evolves.

# data cleaning tools

**OpenRefine (formerly Google Refine)**

What it is:

A free tool used to clean messy data. It works in your browser and helps fix spelling issues, remove duplicates, and make data consistent.

What you can do with it:

Change all spellings of “pakistan”, “Pakistan”, “PAKISTAN” to the same format

Remove rows that are repeated

Find and fix spelling mistakes (like “Ali” and “Alee”)

# data cleaning tools

**Trifacta (Now part of Google Cloud as Dataprep)**

What it is:

A paid tool for cleaning and preparing data. It has a simple drag-and-drop design and gives suggestions to clean your data automatically.

What you can do with it:

Find and fill missing data

Get suggestions on how to fix issues

Clean big files without coding

See your data using graphs and charts

# data cleaning tools

**DataWrangler (from Stanford University)**

What it is:

A web-based tool that helps clean data step-by-step using your mouse. You don’t need to write code — just click and apply actions.

What you can do with it:

Remove extra spaces or symbols

Change text from lowercase to Title Case (like “john doe” → “John Doe”)

Fix formatting in tables

# data cleaning tools

**Pandas (Python Library)**
**What it is:**
A **Python programming library** that gives you powerful tools to clean and manage data using code.
**What you can do with it (Example code):**
python
import pandas as pd df = pd.read_csv('data.csv') # Load your data
df.drop_duplicates(inplace=True) # Remove duplicate rows df.fillna("Unknown", inplace=True) # Fill missing values df['Name'] = df['Name'].str.title() # Capitalize each word in names

# Time series forecasting with BI tools:

**Data Preparation**:

Gather historical time-series data: Collect and organize your historical data, ensuring it's in a format that your BI tool can work with (e.g., CSV, Excel, or a compatible database).

**Example:**

Collect daily sales data of a store from 2020 to 2024 in an Excel sheet.

**Data Loading**:

Import data into your BI tool: Use the data loading or integration features of your BI tool to bring in your time-series data.

**Example:**

In Power BI, click on “Get Data” → select your Excel or CSV file and load the sales data.

**Data Exploration**:

Explore your time-series data: Use the visualization and exploration capabilities of your BI tool to understand patterns, trends, and seasonality in your data.

**Example:**

Create a line chart to see how sales change month by month.

**Feature Engineering**:

Create relevant features: Depending on your data and domain, you may need to create additional features, such as lag variables, moving averages, or indicators for holidays or events.

**Examples:**

**Lag variable:** Add a column that shows previous days sales

# Time series forecasting with BI tools:

**Feature Engineering**:

Create relevant features: Depending on your data and domain, you may need to create additional features, such as lag variables, moving averages, or indicators for holidays or events.

**Examples:**

**Lag variable:** Add a column that shows previous day's sales

**Moving average:** Add a column with average sales of past 7 days

**Holiday flag:** Mark holidays with 1 and normal days with 0

# Time series forecasting with BI tools:

**Model Selection**:

Choose a forecasting model: Select an appropriate time series forecasting method, such as ARIMA (AutoRegressive Integrated Moving Average), Exponential Smoothing, or machine learning models like LSTM (Long Short-Term Memory).

**Examples of models:**

**ARIMA:** For classic time series data

**Exponential Smoothing:** For simple trends

**LSTM (Neural Network):** For complex data using AI

**Data Splitting**:

Split the data: Divide your historical data into training and testing sets to evaluate the accuracy of your forecast.

**Example:**

Use 2020–2023 data to train, and 2024 data to test your model's forecast.

# Contd

**Model Training**:

Train the forecasting model: Use the training data to fit your chosen forecasting model. Depending on your BI tool's capabilities, you may need to implement this using custom calculations or scripts.

**Forecast Generation**:

Generate forecasts: Apply the trained model to your historical data to create forecasts for future time periods. This may involve scripting or custom calculations in your BI tool.

**Visualization**:

Visualize the forecasts: Create charts and graphs in your BI tool to visualize the historical data alongside the forecasts, helping stakeholders understand the predictions.

**Evaluation**:

Assess forecast accuracy: Use your testing data to evaluate the accuracy of your forecasts. Common metrics include Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

# Contd

**Deployment**:

Publish and share forecasts: Share the results, charts, and forecasts with relevant stakeholders within your organization through the BI tool's reporting and sharing capabilities.

**Monitoring and Updating**:

Continuously monitor and update forecasts: Regularly update your forecasts with new data and retrain your model to improve accuracy over time.

# THANKS