**Data Cleaning Techniques**

Week 1: Handling Missing Values

* Goal: Understand and apply various techniques to handle missing data in datasets.
* Description: Missing values are common in datasets and can significantly impact data quality. This week will focus on identifying and addressing missing values.
* Objectives:
  + Learn different types of missing data (MCAR, MAR, NMAR).
  + Implement various techniques to handle missing values.
* Sub-topics:
  + Types of missing data
  + Imputation methods
  + Removal strategies
* Techniques:
  + Removal (listwise, pairwise)
  + Imputation (mean, median, mode, KNN, MICE, predictive modeling)
* Tools:
  + Python (Pandas, Scikit-learn)
  + R (tidyr, mice)
* Data Sources:
  + UCI Machine Learning Repository
  + Kaggle Datasets

Week 2: Handling Outliers

* Goal: Detect and manage outliers to improve data quality and accuracy.
* Description: Outliers can skew statistical analyses and models. This week will focus on identifying and handling outliers.
* Objectives:
  + Understand the impact of outliers.
  + Learn methods for detecting and handling outliers.
* Sub-topics:
  + Outlier detection methods
  + Handling strategies
* Techniques:
  + Z-Score
  + Interquartile Range (IQR)
  + Visual methods (boxplots, scatter plots)
* Tools:
  + Python (Pandas, NumPy, Matplotlib)
  + R (dplyr, ggplot2)
* Data Sources:
  + UCI Machine Learning Repository
  + Kaggle Datasets

Week 3: Data Transformation

* Goal: Learn techniques for transforming data to improve model performance and analysis.
* Description: Data transformation involves normalizing, standardizing, and encoding data to make it suitable for analysis.
* Objectives:
  + Understand data transformation concepts.
  + Apply various transformation techniques.
* Sub-topics:
  + Normalization vs. Standardization
  + Encoding categorical variables
* Techniques:
  + Min-Max Normalization
  + Z-Score Standardization
  + One-Hot Encoding
* Tools:
  + Python (Scikit-learn, Pandas)
  + R (caret, dplyr)
* Data Sources:
  + UCI Machine Learning Repository
  + Kaggle Datasets

Week 4: Data Parsing and Text Data Cleaning

* Goal: Develop skills to clean and parse text data for analysis.
* Description: Text data often needs to be parsed and cleaned to extract meaningful information. This week will focus on text data cleaning techniques.
* Objectives:
  + Learn text parsing and cleaning techniques.
  + Apply methods to clean and structure text data.
* Sub-topics:
  + Text parsing techniques
  + Cleaning and preprocessing text data
* Techniques:
  + Tokenization
  + Stopword removal
  + Stemming and Lemmatization
* Tools:
  + Python (NLTK, SpaCy, Pandas)
  + R (tm, quanteda)
* Data Sources:
  + Kaggle Datasets (text data)
  + OpenText Corpus

Week 5: Dealing with Duplicate Data

* Goal: Identify and manage duplicate data to ensure data integrity.
* Description: Duplicate records can lead to misleading analysis. This week focuses on detecting and handling duplicates.
* Objectives:
  + Understand the impact of duplicate data.
  + Implement methods to identify and remove duplicates.
* Sub-topics:
  + Duplicate detection
  + Removing duplicates
* Techniques:
  + Exact Matching
  + Fuzzy Matching
* Tools:
  + Python (Pandas, FuzzyWuzzy)
  + R (dplyr, stringdist)
* Data Sources:
  + UCI Machine Learning Repository
  + Kaggle Datasets

Week 6: Data Validation

* Goal: Implement data validation techniques to ensure data accuracy and consistency.
* Description: Data validation involves checking data for consistency and accuracy. This week will focus on implementing validation techniques.
* Objectives:
  + Learn about data validation techniques.
  + Apply methods to validate data.
* Sub-topics:
  + Consistency checks
  + Range checks
* Techniques:
  + Validation Rules
  + Constraint Checking
* Tools:
  + Python (Pandas)
  + R (dplyr)
* Data Sources:
  + UCI Machine Learning Repository
  + Kaggle Datasets

Week 7: Data Type Conversion

* Goal: Convert and standardize data types to facilitate analysis and modeling.
* Description: Converting data types is essential for ensuring data compatibility and accuracy. This week will cover data type conversion techniques.
* Objectives:
  + Understand different data types and their conversions.
  + Apply data type conversion techniques.
* Sub-topics:
  + Data type conversions
  + Standardizing formats
* Techniques:
  + Conversion Functions
  + Data Type Casting
* Tools:
  + Python (Pandas)
  + R (dplyr)
* Data Sources:
  + UCI Machine Learning Repository
  + Kaggle Datasets

Week 8: Addressing Data Entry Errors

* Goal: Correct and standardize data entry errors to improve data quality.
* Description: Data entry errors can arise from manual input. This week will focus on identifying and correcting these errors.
* Objectives:
  + Identify common data entry errors.
  + Implement methods to correct and standardize data.
* Sub-topics:
  + Common data entry errors
  + Correction techniques
* Techniques:
  + Typo Correction
  + Standardization of Formats
* Tools:
  + Python (Pandas, regex)
  + R (stringr)
* Data Sources:
  + UCI Machine Learning Repository
  + Kaggle Datasets

Week 9: Handling Inconsistent Data

* Goal: Standardize and harmonize inconsistent data to ensure uniformity.
* Description: Inconsistent data can result from varying formats or units. This week will cover methods for handling inconsistent data.
* Objectives:
  + Understand sources of inconsistency.
  + Apply techniques to handle and standardize inconsistent data.
* Sub-topics:
  + Identifying inconsistencies
  + Harmonizing data formats and units
* Techniques:
  + Data Harmonization
  + Unit Conversion
* Tools:
  + Python (Pandas)
  + R (dplyr)
* Data Sources:
  + UCI Machine Learning Repository
  + Kaggle Datasets

Week 10: Data Enrichment and Feature Engineering

* Goal: Enhance data with external sources and create new features for better analysis and modeling.
* Description: Data enrichment involves adding external data, while feature engineering involves creating new features from existing data. This week will focus on these advanced techniques.
* Objectives:
  + Learn about data enrichment and feature engineering.
  + Apply techniques to enrich data and create new features.
* Sub-topics:
  + Data enrichment methods
  + Feature engineering techniques
* Techniques:
  + Merging with external data sources
  + Creating new features (e.g., interaction terms, polynomial features)
* Tools:
  + Python (Pandas, Scikit-learn)
  + R (dplyr, caret)
* Data Sources:
  + Kaggle Datasets
  + Public APIs

Chapterwise

**Chapter 1: Handling Missing Values**

* **Task 1.1: Introduction to Missing Data**
  + Research and understand the different types of missing data: MCAR, MAR, NMAR.
  + Read relevant literature or documentation on missing data.
* **Task 1.2: Identifying Missing Values**
  + Load a dataset with missing values.
  + Use Python (Pandas) or R (dplyr) to identify and visualize missing values.
* **Task 1.3: Handling Missing Data**
  + Implement removal techniques (listwise and pairwise).
  + Apply imputation methods: mean, median, mode, KNN, MICE, and predictive modeling.
  + Compare the effects of different methods on data quality.
* **Task 1.4: Evaluation and Documentation**
  + Evaluate the impact of different handling methods on dataset completeness and accuracy.
  + Document your process and findings.

**Chapter 2: Handling Outliers**

* **Task 2.1: Introduction to Outliers**
  + Research and understand the types of outliers and their impact on data analysis.
  + Read relevant literature or documentation on outlier detection.
* **Task 2.2: Detecting Outliers**
  + Load a dataset with potential outliers.
  + Use Z-Score, IQR, and visual methods (boxplots, scatter plots) to detect outliers.
* **Task 2.3: Handling Outliers**
  + Implement methods for handling outliers, including removal or transformation.
  + Compare the results before and after handling outliers.
* **Task 2.4: Evaluation and Documentation**
  + Evaluate the impact of outlier handling on dataset distribution and analysis.
  + Document your process and findings.

**Chapter 3: Data Transformation**

* **Task 3.1: Introduction to Data Transformation**
  + Research and understand normalization and standardization techniques.
  + Read relevant literature or documentation on data transformation.
* **Task 3.2: Normalization and Standardization**
  + Load a dataset and apply Min-Max Normalization and Z-Score Standardization.
  + Compare the effects of these transformations on data.
* **Task 3.3: Encoding Categorical Variables**
  + Apply One-Hot Encoding and other encoding techniques to categorical variables.
  + Evaluate the impact on dataset usability.
* **Task 3.4: Evaluation and Documentation**
  + Evaluate the impact of data transformation on dataset distribution and analysis.
  + Document your process and findings.

**Chapter 4: Data Parsing and Text Data Cleaning**

* **Task 4.1: Introduction to Text Data Cleaning**
  + Research and understand text parsing and cleaning techniques.
  + Read relevant literature or documentation on text data cleaning.
* **Task 4.2: Text Parsing**
  + Load a text dataset and apply tokenization, stopword removal, and other parsing techniques.
* **Task 4.3: Text Data Cleaning**
  + Implement stemming and lemmatization techniques to clean text data.
  + Evaluate the results of text cleaning.
* **Task 4.4: Evaluation and Documentation**
  + Evaluate the impact of text cleaning on data quality and analysis.
  + Document your process and findings.

**Chapter 5: Dealing with Duplicate Data**

* **Task 5.1: Introduction to Duplicate Data**
  + Research and understand the impact of duplicate data on analysis.
  + Read relevant literature or documentation on duplicate data handling.
* **Task 5.2: Detecting Duplicates**
  + Load a dataset with potential duplicates.
  + Use exact matching and fuzzy matching to identify duplicates.
* **Task 5.3: Handling Duplicates**
  + Implement methods for removing or consolidating duplicate records.
  + Compare the dataset before and after handling duplicates.
* **Task 5.4: Evaluation and Documentation**
  + Evaluate the impact of duplicate removal on dataset integrity.
  + Document your process and findings.

**Chapter 6: Data Validation**

* **Task 6.1: Introduction to Data Validation**
  + Research and understand data validation techniques and their importance.
  + Read relevant literature or documentation on data validation.
* **Task 6.2: Implementing Validation Checks**
  + Load a dataset and apply consistency checks and range checks.
  + Use Python (Pandas) or R (dplyr) to perform validation.
* **Task 6.3: Handling Validation Issues**
  + Address any issues identified during validation.
  + Implement strategies to ensure ongoing data quality.
* **Task 6.4: Evaluation and Documentation**
  + Evaluate the impact of validation checks on dataset accuracy.
  + Document your process and findings.

**Chapter 7: Data Type Conversion**

* **Task 7.1: Introduction to Data Types**
  + Research and understand different data types and their conversions.
  + Read relevant literature or documentation on data type conversion.
* **Task 7.2: Converting Data Types**
  + Load a dataset with mixed data types.
  + Implement conversion techniques to standardize data types.
* **Task 7.3: Standardizing Formats**
  + Apply standardization methods to ensure consistency in data formats.
  + Evaluate the results of data type conversion.
* **Task 7.4: Evaluation and Documentation**
  + Evaluate the impact of data type conversion on dataset usability.
  + Document your process and findings.

**Chapter 8: Addressing Data Entry Errors**

* **Task 8.1: Introduction to Data Entry Errors**
  + Research and understand common data entry errors and their impact.
  + Read relevant literature or documentation on data entry error correction.
* **Task 8.2: Identifying Data Entry Errors**
  + Load a dataset with potential data entry errors.
  + Use Python (Pandas) or R (dplyr) to identify and analyze errors.
* **Task 8.3: Correcting Data Entry Errors**
  + Implement methods for correcting typos and standardizing data formats.
  + Evaluate the impact of error correction on data quality.
* **Task 8.4: Evaluation and Documentation**
  + Evaluate the impact of error correction on dataset accuracy and analysis.
  + Document your process and findings.

**Chapter 9: Handling Inconsistent Data**

* **Task 9.1: Introduction to Inconsistent Data**
  + Research and understand the sources of data inconsistency and its impact.
  + Read relevant literature or documentation on handling inconsistent data.
* **Task 9.2: Identifying Inconsistencies**
  + Load a dataset with potential inconsistencies.
  + Use Python (Pandas) or R (dplyr) to identify and analyze inconsistencies.
* **Task 9.3: Harmonizing Data**
  + Implement methods for standardizing units and formats.
  + Use techniques to merge similar categories or harmonize data.
* **Task 9.4: Evaluation and Documentation**
  + Evaluate the impact of data harmonization on dataset consistency.
  + Document your process and findings.

**Chapter 10: Data Enrichment and Feature Engineering**

* **Task 10.1: Introduction to Data Enrichment and Feature Engineering**
  + Research and understand data enrichment and feature engineering techniques.
  + Read relevant literature or documentation on these advanced techniques.
* **Task 10.2: Enriching Data**
  + Load a dataset and merge it with external data sources.
  + Apply methods for enriching data with additional information.
* **Task 10.3: Feature Engineering**
  + Create new features from existing data (e.g., interaction terms, polynomial features).
  + Evaluate the impact of new features on data analysis.
* **Task 10.4: Evaluation and Documentation**
  + Evaluate the impact of data enrichment and feature engineering on dataset analysis.
  + Document your process and findings.