DATA CLEANING CHECKLIST

www.calebbunton.net

Understand the Dataset

Start by thoroughly reviewing the dataset to understand its structure and purpose.

- ☐ Review the dataset's documentation and metadata.
- ☐ Familiarize yourself with key variables and data types.
- ☐ Identify the expected ranges and formats for each column.

NOTES

Handling Missing Data

Detect and handle any missing values to ensure your dataset is complete and accurate.

- ☐ Identify missing values using .isnull() (Python) or is.na() (R).
- ☐ Analyze if the missing data follows any patterns.
- ☐ Impute missing data (mean, median, or mode), or remove rows/columns with excessive missing data.

NOTES

Fix Data Types

Ensure all variables are correctly formatted to avoid errors during analysis.

- ☐ Check that numeric, categorical, and date columns are correctly typed.
- ☐ Convert columns to appropriate data types (e.g., pd.to_numeric()).
- ☐ Format date/time columns for consistency (e.g., pd.to_datetime()).

NOTES

Handling Duplicates

Detect and eliminate duplicate rows to prevent redundancy in your data.

- ☐ Detect duplicate rows using .duplicated() (Python) or duplicated() (R).
- ☐ Remove duplicates using .drop_duplicates() (Python).
- ☐ Ensure key columns are unique where necessary (e.g., IDs).

NOTES

Outliers Detection and Treatment

Identify and assess outliers. Remove or retain them based on your analysis goals.

- ☐ Use statistical methods (e.g., z-scores, IQR) or boxplots to identify outliers.
- ☐ Analyze outliers to determine if they are valid or data entry errors.
- ☐ Remove or transform outliers (e.g., apply log transformation if necessary).

NOTES

Handle Inconsistent Data

Standardize inconsistent entries, especially in categorical data, to ensure uniformity.

- ☐ Detect inconsistent values (e.g., 'USA' vs 'United States').
- ☐ Standardize categories and ensure text formatting (e.g., all lowercase).
- ☐ Cross-check categories for spelling or case sensitivity issues.

NOTES

Remove Unwanted Characters Clean up unnecessary spaces, special characters, and formatting errors.	☐ Remove leading/trailing spaces using .strip() (Python) or trimws() (R).	**NOTES**
	☐ Eliminate special characters using regex or string replacement.	
	☐ Ensure uniform formatting for text fields.	

Normalize and Standardize Data Scale numeric data to make it comparable across different units or ranges.	☐ Normalize data to bring all values into a similar range (e.g., Min-Max scaling).	**NOTES**
	☐ Standardize data (mean = 0, standard deviation = 1) for algorithms that require standardized inputs.	
	☐ Apply appropriate scaling methods based on analysis needs.	

Create and Validate Derived Variables Generate new features and validate their correctness before analysis.	☐ Create new variables (e.g., age from date of birth).	**NOTES**
	☐ Validate derived variables with cross-checks and summary statistics.	
	☐ Ensure that new variables follow consistent formatting and logical rules.	

Ensure No Data Leakage Double-check data. Remove information that would be unavailable in real-world analysis.	☐ Ensure no target variables or future data are included in the training set.	**NOTES**
	☐ Review features to prevent information leaks during model training.	
	☐ Split your data into training, validation, and test sets before performing feature engineering	

Ensure Consistent Formatting Ensure dates, text, and other formatted fields follow consistent patterns.	☐ Standardize date formats (e.g., YYYY-MM-DD).	**NOTES**
	☐ Ensure consistent text formatting (e.g., same case, punctuation, etc.).	
	☐ Check for and correct any formatting irregularities in all columns.	

Validate Data Use summary statistics and external references to verify the accuracy of your cleaned data.	☐ Run summary statistics (.describe() in Python) to verify data ranges and distributions.	**NOTES**
	☐ Cross-check your data with external sources or domain expertise.	
	☐ Investigate any values that fall outside expected ranges.	

Backup the Cleaned Data Safeguard your progress by creating a backup of your cleaned dataset.	☐ Create a backup copy of the cleaned dataset.	**NOTES**
	☐ Store backups securely (e.g., cloud storage, version control system).	
	☐ Keep a version history to track changes in case of errors.	

SNIPPETS / TIPS

Use Pandas Profiling for Quick Data Insights

Tip: Generate a comprehensive report of your dataset using the pandas_profiling package in Python. It provides an overview of missing values, statistics, correlations, and more.

```
import pandas_profiling
profile = pandas_profiling.ProfileReport(df)
profile.to_file("data_report.html")
```

Use this to get a quick overview of your dataset before cleaning.

Detect and Replace Outliers Efficiently

Tip: Use the IQR (Interquartile Range) method to detect outliers and handle them by capping or removing them.

```
Q1 = df['column_name'].quantile(0.25)
Q3 = df['column_name'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
df = df[(df['column_name'] >= lower_bound) &
(df['column_name'] <= upper_bound)]
```

This method helps you handle outliers in numerical data.

Fill Missing Data Using Interpolation

Tip: For time series data, you can use interpolation to fill in missing values in a logical way based on surrounding data.

```
df['column_name'].interpolate(method='linear',
inplace=True)
```

Use this when missing values are within a time sequence and can be estimated using surrounding points.

Normalize Data Easily

Tip: Use the MinMaxScaler from scikit-learn to scale numeric values to a range between 0 and 1.

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
df[['column1', 'column2']] =
scaler.fit_transform(df[['column1', 'column2']])
```

This is useful for preparing data for machine learning models.

Split Data Properly

Tip: Always split your dataset into training, validation, and test sets to avoid data leakage.

```
from sklearn.model_selection import train_test_split
train, test = train_test_split(df, test_size=0.2,
random_state=42)
```

This ensures your model is evaluated on unseen data.

Looking for more data science resources and practical tools?

Visit my website at www.calebbunton.net for in-depth guides, tutorials, and exclusive downloads to accelerate your data science journey!

Connect with me on LinkedIn for updates on new content and tips to improve your data analysis skills. Let's clean data the smart way!