Kenan Stredics
Handling NaN Values Report

1. Imputation:
   - Mean/Median/Mode Imputation: Replace NaN values with the mean, median, or mode of the respective feature. This approach works well for numerical features and is simple to implement.
   - Custom Value Imputation: Replace NaN values with a specific value chosen based on domain knowledge or data analysis. This can be useful when the mean or median might not be appropriate.

2. Deletion:
   - Row Deletion: Remove entire rows containing NaN values. This approach is straightforward but may result in loss of valuable data, especially if NaN values are prevalent.
   - Column Deletion: Remove entire columns with a significant number of NaN values. This can be effective if certain features have a high proportion of missing values and are not crucial for the analysis.

3. Prediction:
   - Use machine learning algorithms, such as regression or k-nearest neighbors, to predict missing values based on other features in the dataset. This approach can be computationally intensive but can provide accurate imputations, especially if the missing data pattern is complex.

4. Flagging and Encoding:
   - Create a new binary feature indicating whether a value is missing or not. This allows the model to learn from the absence of data. Additionally, encode categorical features with NaN values as a separate category to preserve information about missingness.

5. Feature Engineering:
   - Create new features that capture the missing data pattern. For example, create a binary feature indicating whether a certain feature has missing values or engineer features based on the relationships between missing and non-missing values in the dataset.

6. Domain-specific Knowledge:
   - Utilize knowledge about the domain or the data collection process to decide the best approach for handling NaN values. For example, if NaN values occur due to sensor malfunction, imputing with a custom value may be more appropriate than using the mean or median.

Each approach has its trade-offs in terms of computational complexity, data loss, and potential impact on model performance. Experimenting with different strategies and considering the specific characteristics of the dataset are essential for effective handling of NaN values in machine learning.