CMPT 310
How-To Guide
Albert Hong

# *SiftMail*

Data Cleaning and Preprocessing Report

## Summary:

This report presents a data preprocessing pipeline designed to transform a raw, messy email dataset into a structured, machine learning ready format. The pipeline uses many techniques in text processing, feature extraction, embedding generation, and data quality management. Will all work together to produce a refined dataset for analytical purposes.

## Introduction & Overview:

In real world scenarios, datasets are rarely clean or are ready to be used for analysis upon retrieving. Instead, they often contain missing values, inconsistencies, incorrect formats, and irrelevant information. This report/guide details the process that was used to clean and preprocess a messy dataset, preparing it for analysis and modeling. The aim was to ensure data quality, integrity, and usability. Each procedure described here addresses a common issue encountered in raw data and outlines how it was resolved in this project.

### Initial Inspection:

The first step of the workflow involved performing an initial inspection of the dataset to understand its structure, quality, and overall content inside. This included loading the dataset into a data frame and using summary statistics, data type checks, and shape inspection to get an introductory understanding of the dataset we are using. For us, this phase identified various data issues, such as missing values, inconsistent column formats, and incorrect data types. This gave us a clear picture of the current dataset's condition which helped us guide the decisions needed to be taken in the preprocessing phase.

It cannot be stressed enough that understanding the type and scale of issues upfront is important. Not only does it inform us what strategies need to be applied but also prevents the misuse of methods that might introduce new errors or bias. Therefore, a thorough inspection was essential before undertaking any transformation or modeling steps.

CMPT 310
How-To Guide
Albert Hong

Missing Values:

After the initial inspection, the first problem to tackle was handling missing values. Missing data was addressed using a column specified approach rather than a one size fit all solutions. Depending on how much were missing, some columns were estimated. For categorical columns, missing entries were filled with a placeholder value. This step is crucial as missing values can cause errors in many algorithms or can create bias results if left untreated. Filling them retains as much information as possible without discarding valuable data.

Misclassification of Data Types & Normalization:

Next issue encountered is the misclassification of data types. For example, columns containing dates were stored as strings, and numerical columns were sometimes represented as object types due to formatting errors. These were addressed through type conversion using appropriate methods for each case.

Date strings were converted into proper datetime objects, allowing for time-based feature extraction such as day of week. Categorical variables were cast into the category type to optimize memory usage and processing speed. These conversions also enabled efficient encoding later in the project.

The raw dataset also underwent several normalization procedures to improve consistency and reduce redundancy. In our case, all text were converted to lowercase to eliminate case related duplicates. In addition, extra spaces, special characters, and HTML tags were stripped.

Feature Engineering and Encoding:

Several new features were created from existing data to improve the dataset as well as its predictive power. Categorical data was handled in different ways depending on the type of information. For categories without any specific order, like folder names or job titles, we used one-hot encoding to turn them into separate yes/no columns. This avoids accidentally implying any sort of ranking. For categories that do have a natural order, like priority levels, we used label encoding to keep that order intact. In some cases, we also grouped numerical values into ranges or bins to make them easier to understand and more useful for models that work better with categories.

CMPT 310
How-To Guide
Albert Hong

## Conclusion:

This preprocessing pipeline demonstrates a careful and systematic approach to preparing a real, raw dataset for analysis. It includes methods for handling missing data, correcting data types, cleaning and normalizing text, feature extraction, encoding categorical variables, and much more. All these steps together ensure that the dataset is consistent, complete, and ready for analysis or machine learning.