

May 2, 2021

Quinn E Knudsen

IST 736

Syracuse University

The dirty secrets of data science

Cleaning Messy Review Data for Modeling

**Table of Contents**

**Analysis**………………………………………………………………………………..…pg 2

About the Data ...…………………………………………………………….…...pg 2

*Similarities*….. ...…………………………………………………….…...pg 2

*Differences*….. ...…………………………………………………….…...pg 2

*Challenges*….. ...…………………………………………………….…....pg 3

**Results**…....………………………………………………………………………….…....pg 3

Results 1: Restaurant Reviews…………………………………………….……....pg 3

Data Cleaning & Prep……….…………………………………………….…….....pg 3

*Before Prep…………………*…………………………………………… ….pg 3 *Reading in the Data and Initial Preparation* …………………………..…pg 4 *Development of Word Clouds for Frequency* …………………………..….pg 4 *Document Term Matrix*………………………………………..………….. pg 5 *Normalized Document Term Matrix (tf-idf)* …………………………….…pg 6 *Remove Columns with Numbers or Words Less than 3 Letters..*…………..pg 6 *Classification Labels …………………………..*……………………………. pg 7

Results 2: Movie Reviews………………………………………………………….pg 7

Data Cleaning & Prep……….…………………………………………….………..pg 7

*Before Prep…………………*……………………………………………...…pg 7 *Reading in the Data and Initial Preparation* …………………………..… pg 8 *Development of Word Clouds for Frequency* ………………………….….. pg 8 *Document Term Matrix*………………………………………..……………pg 10 *Normalized Document Term Matrix (tf-idf)* ………………………………. pg 10  *Classification Labels ……………………..*…………………………………...pg 10

**Analysis**

Unique to this paper, the analysis section will discuss the two datasets, their similarities and differences, and the challenges faced in preparing and formatting each into labeled data frames.

**About the Data**

This analysis consisted of two disparate datasets on unrelated topics. The main objective is to clean and categorize the messy data by using the predefined review labels “positive” and “negative”.

*Dataset 1: Restaurant Reviews*

This dataset contained 9 restaurant reviews as free text under the variable “review”. Each review represented more than one vertical column indicating data preparation was needed to combine the segments of the review into one cohesive variable. The dataset also contained a variable “sentiment” which was a categorization of the underlying emotion in the review. This variable contained two levels “positive” and “negative” with the negative category containing five statements and positive category containing four.

*Dataset 2: Movie Reviews*

The second dataset also contained review data. The area of focus for this dataset, however, was on movie reviews. Similar to the prior dataset, two variables exist: one review categorization and the text. Unique to this data, the text classification occurred at the end of several columns worth of data in the .csv file. Also, this dataset was riddled with “\” and “n” that do not bring meaning to the review. Data preparation for this design required careful attention. This dataset contains five reviews with three negative classifications and two positive classifications.

*Similarities*

There are some clear similarities in the nature of the two datasets. The most overt comparison can be made through the overarching topic: sentiment classification of human reviews. In each dataset, there is a given review which represents one row and there is an associated classification of positivity or negativity. While both datasets are small in depth, they become spares in breadth when converted to a document term matrix of tf-idf. Another comparison between the two sets or text is that the classifications are imbalanced meaning that there are not equal positive and negative responses. This is important to consider when modeling and should deter looking at a metric such as accuracy in a vacuum. The datasets are also filled with stop words and alphanumeric symbols that are not meaningful for most analyses. The datasets are written exclusively in English and are devoid of symbols or non-textual indicators of emotion such as an emoji. Each dataset contained reviews for more than one topic. For example, the restaurant reviews discussed multiple restaurants and the movie reviews spanned many movies. The final comparison, albeit basic, is that both datasets were a .csv file.

*Differences*

The first key difference between the datasets is the context around the reviews that were written. One dataset provides reviews of movies while the other supplies restaurant reviews. The length of the datasets differ with the movie review containing four additional reviews. While both reviews containing subjective opinions, the restaurant review tended to use more personal language such as “I” whereas the movie reviews were written a bit more objectively. While both provide a review and a classification, the restaurant review sentiment classification was tied nicely in a unique column with an appropriate header. On the other hand, the movie review dataset placed the classification after the final column in the review meaning the column the review was located in was not unique or easily identifiable.

*Challenges*

Detailed challenges and the methodology utilized to circumvent them will be explained in great detail in the results section. As an overview, there were three key challenges with these datasets. The first, applies to both the movie and restaurant reviews. The review text spanned multiple columns instead of being contained under one labeled column header. This presents a problem in organizing the data as a data frame. A second challenge is the occurrence of undesirable words and alphanumeric characters. These would generally not be meaningful as columns in a sparse data frame and will need to be removed. These stop words were considered to be both the typical English words, numbers and select custom words. With greater customization comes a greater challenge in properly preparing the data frame. The final large-scale data preparation dilemma was how to deal with various iterations of words (capitalization, conjugation and other word syntax) that might unnecessarily widen the dataset or detract meaning from the analyses. Steps taken in the data cleaning did not utilize stemming or lemmatization, but future analyses would benefit from this approach when there are more revies, greater text and greater syntax requirements for analysis. All words were converted to lowercase for consistency.

**Results**

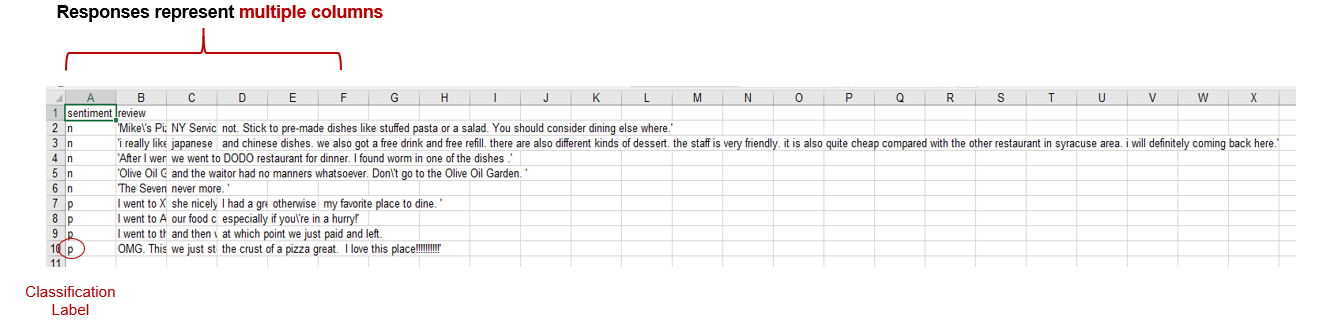
Unique to this paper, the results section will detail the steps taken to clean and prepare the datasets for further modeling using pictures to walk through the before during and aftereffects of the data preparation process.

**Results 1: Restaurant Reviews**

**Data Cleaning & Prep**

*Before Prep*

Prior to any preprocessing, as described, the dataset was not adequately prepared for conversion to a structured data frame. Before conversion to a document term matrix or tf-idf as will be detailed in the ensuing stages of preparation, the dataset appears as outlined in Figure 1.



*Figure 1.* A snapshot of the data before any preparation was taken with review data filling multiple columns.

*Reading in the Data and Initial Preparation*

Data was imported into Python skipping the header line containing the labels as to not confuse it with the text. Two lists were created with one containing the labels and the other the content with a split after the first comma to ensure the correct segmentation was achieved.

*Development of Word Clouds for Frequency*

Next, the review list was converted into a data frame with an index and a column of text strings labeled as “Review”. Leveraging the nltk stopwords and custom words “go” and “going” the text was stripped of common English words and tokenized. Each tokenized word was converted to lowercase for consistency. These words were then visualized in a word cloud to show frequency as demonstrated in Figure 1 on the next page.



*Figure 2.* The most commonly used words in the restaurant reviews after removing stop words.

An additional visualization was created in the shape of a slice of pizza which could carry additional domain specific significance.



*Figure 3.* Domain specific representation of the most commonly used words in the restaurant reviews after removing stop words.

*Document Term Matrix*

Python package sklearn CountVectorizer was instantiated to read in the 9 text files and English stop words were removed. Stop words are common English language words such as “the”, “at”, “there” that generally provide little information gain due to the commonality of their utilization. Files were then vectorized and compiled into a document term matrix. This document term matrix was then converted into an array before finally becoming structured as a data frame that was essential for the analysis.

The final preparation for creating a document term matrix was to label each row in the data frame by their classified positivity or negativity. This was done by using the text name label (‘n’ or ‘p’). This resulted in a sparse, highly dimensional, term matrix with each word representing a variable and each text file representing an observation as depicted in Table 1 below.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Index** | **15** | **abc** | **acknowledged** | **ago** | **american** | **appatizing** | **appetizer** | **applied** | **area** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 6 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 7 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

*Table 1.* A document term matrix exemplifying the conversion made.

*Normalized Document Term Matrix (tf-idf)*

An additional step was taken to convert the document term matrix into a normalized document term matrix using tf-idf transformation. This transformation takes the normalized value for which the word occurs in the text matrix (proportionality). This conversion can be found in Table 2 below.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Index** | **15** | **abc** | **acknowledged** | **ago** | **american** | **appatizing** | **appetizer** | **applied** | **area** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0.187887 | 0 | 0 | 0 | 0.187887 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0.159344 | 0 | 0 | 0 | 0.134584 | 0.159344 | 0 |
| 6 | 0 | 0.213158 | 0 | 0.213158 | 0 | 0.213158 | 0 | 0 | 0 |
| 7 | 0.165157 | 0 | 0 | 0 | 0 | 0 | 0.139494 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

*Table 2.* Conversion of document term matrix to normalized tf-idf matrix showing the normalized word frequency.

*Remove Columns with Numbers or Words Less than 3 Letters*

As indicated in Tables 1 and 2, there are still some undesirable columns included in the dataset even after stop words removal such as ‘15’ and “abc”. In order to remove these from the data, a Boolean function was written to return a string for any numbers in a string. Then, using a simple if🡪then logical statement, all numeric data (labeled now as ‘True’) were dropped from the document term matric and tf-idf datasets. A secondary if🡪then statement was then utilized to drop all columns with less than three characters. This yielded the following modifications to the matrices. There are some limitations to this approach such as the removal of “ago” which may or may not provide value in the classification. The modifications are displayed in Tables 3 and 4.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Index** | **15** | **abc** | **acknowledged** | **ago** | **american** | **appatizing** | **appetizer** | **applied** | **area** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 6 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 7 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

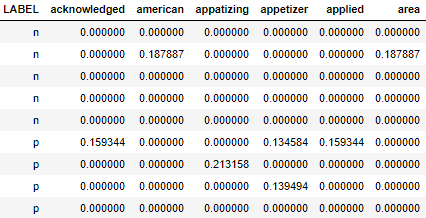
*Table 3.* Numeric columns and those less than three characters are dropped.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Index** | **15** | **abc** | **acknowledged** | **ago** | **american** | **appatizing** | **appetizer** | **applied** | **area** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0.187887 | 0 | 0 | 0 | 0.187887 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0.159344 | 0 | 0 | 0 | 0.134584 | 0.159344 | 0 |
| 6 | 0 | 0.213158 | 0 | 0.213158 | 0 | 0.213158 | 0 | 0 | 0 |
| 7 | 0.165157 | 0 | 0 | 0 | 0 | 0 | 0.139494 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

*Table 4.* Numeric columns and those less than three characters are dropped.

*Add Classification Labels*

The final stage was to then insert the sentiment label back into the dataset in the first column for both the document term matrix and the normalized tf-idf. The final clean data frame can be seen in Table 5 below.



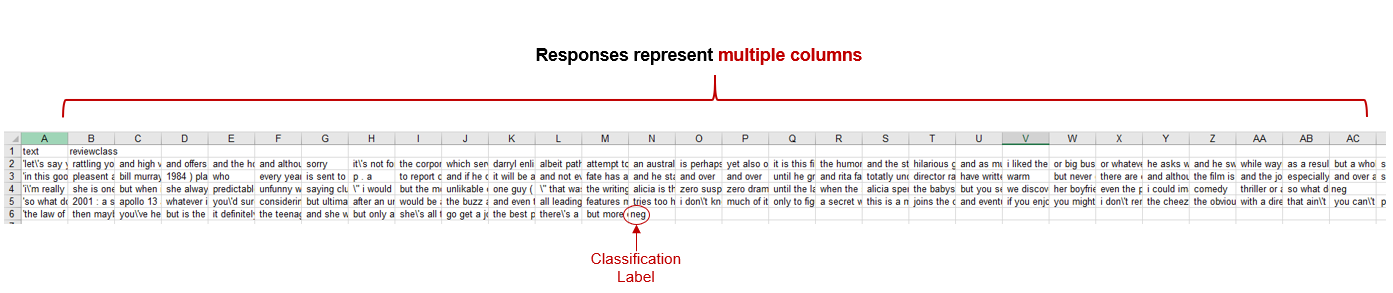
*Table 5.* Cleaned tf-idf matrix ready for modeling.

**Results 2: Movie Reviews**

**Data Cleaning & Prep**

*Before Prep*

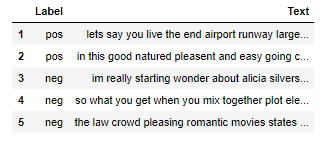
Prior to any preprocessing, as described, the dataset was not adequately prepared for conversion to a structured data frame. Before conversion to a document term matrix or tf-idf as will be detailed in the ensuing stages of preparation, the dataset appears as outlined in Figure 5.



*Figure 5.* A snapshot of the movie review data before any preparation with reviews located in multiple columns and inconsistent column placement for the sentiment.

*Reading in the Data and Initial Preparation*

Data was imported into Python and an empty csv file was created for the final clean version with columns “Label” and “Text”. The file was then read into a data frame and all “NA” values were dropped resulting in a data frame depicted in Table 6 below.



*Table 6.* Simple labeled data frame.

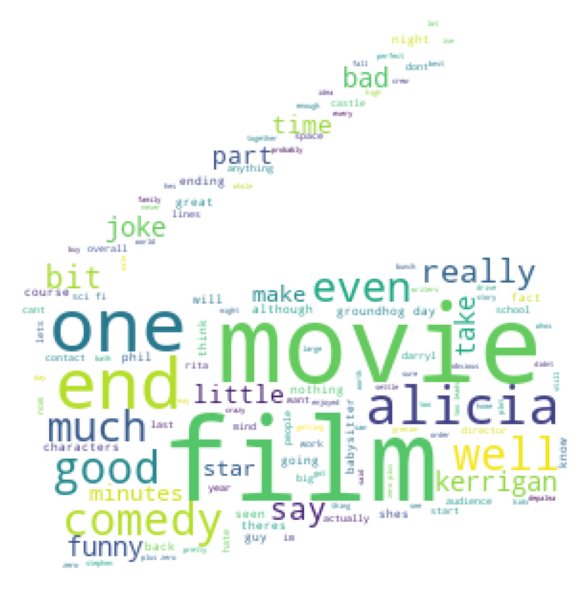
*Development of Word Clouds for Frequency*

Once again, leveraging the nltk stopwords and custom words “go” and “going”, the text was stripped of common English words and tokenized. Each tokenized word was converted to lowercase for consistency. These words were then visualized in a word cloud to show frequency as demonstrated in Figure 3 on the next page.



*Figure 5.* The most commonly used words in the restaurant reviews after removing stop words.

An additional visualization was created in the shape of a movie related image which could carry additional domain specific significance.



*Figure 6.* Domain specific representation of the most commonly used words in the movie reviews after removing stop words.

*Preparing Data for Document Term Matrix*

Next, labels were dropped from the file in order to make sure the data is ready to be converted into a document term matrix and labels should not be included in the matrix itself. This resulted in 5 elements in the list containing just the string data in row form.

*Document Term Matrix*

The unlabeled data was then passed into an empty list before CountVectorizer was used to transform the list based upon word occurrence. The array was then converted to a data frame as in analysis 1 and resulted in the following (see Table 7 for details).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Index** | **abcs** | **able** | **abound** | **absolutely** | **accordingly** | **action** | **activist** | **actual** |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 0 |
| 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 4 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

*Table 7.* Document term matrix for the movie data.

*Normalized Document Term Matrix (tf-idf)* *featuring English and Custom Stop Word Removal*

An additional step was taken to convert the document term matrix into a normalized document term matrix using tf-idf transformation. This transformation takes the normalized value for which the word occurs in the text matrix (proportionality). Common English stop words were removed from this sparse data frame along with words that are nontraditional English such as “abcs”. This conversion can be found in Table 8 below with final label results displayed in Table 9.

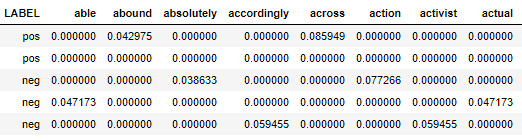
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Index** | **abcs** | **able** | **abound** | **absolutely** | **accordingly** |
| 0 | 0 | 0 | 0.042975 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0.038633 | 0 |
| 3 | 0 | 0.047173 | 0 | 0 | 0 |
| 4 | 1 | 0 | 0 | 0 | 0.059455 |

*Table 8.* TF-idf demonstrates the normalization and stop word removal.



*Table 9.* Labeled tf-idf without English or custom stop words.

The final step was to bring the labeled column to the first row in the data frame. This resulted in the following final view of the data frame after cleaning and preprocessing.



*Table 10.* Final clean movie review dataset with label in the first column.