Lucas Reynolds
Springboard Data Science Career Track
January 2020 Cohort
Capstone Project 1

Data Wrangling

Problem Statement: Can text provided in reviews from medicine users be used to predict the rating they will provide?

Dataset

The Drugs.com dataset was downloaded as a tsv file from the UCI Machine Learning Repository. The dataset contained 161,297 rows and the following 7 columns: *Unnamed* (review ID number), *drugName* (name of the drug reviewed), *condition* (diagnosed condition for which the drug was prescribed), *review* (given text of the review), *rating* (rating from 1.0 to 10.0), *date* (review data), *usefulCount* (number of website visitors who found the review useful).

The dataset was imported as a pandas dataframe and previewed to gain an understanding overall structure and context. The *review* and *rating* columns were checked for any null values. No null values were present.

Rating Context

The number of reviews and normalized percentages for each rating score were calculated. The majority of reviews were for ratings of 1.0 (13.4%), 8.0 (11.7%), 9.0 (17.1%), and 10.0 (31.6%) with ratings 2.0 through 7.0 each holding 6% or less of the reviews.

10 reviews were sampled from each rating score to understand general sentiment and context. Reviews sampled for ratings 1.0 through 4.0 generally had an overall negative sentiment, reviews sampled for ratings 5.0 through 7.0 were generally neutral or contained a mixture of positive and negative sentiment, and reviews sampled for ratings 8.0 through 10.0 generally had an overall positive sentiment.

Cleaning Text

A bag of words model approach was chosen to quantify and evaluate the review text. Review text was normalized by making all lower case, punctuation was removed, terms were separated and tokenized, numeric and alphanumeric terms were removed, stop words were removed, extra spaces and single-letter tokens were removed.

The text was turned into a document term matrix using CountVectorize with a min_df of 30 and max_df of 0.20 to exclude terms that showed up too little or too frequently to be useful.

Saving the Matrix

To speed up the process of saving and accessing later the sparse matrix created from CountVectorize, the matrix was converted to a csr matrix and saved as an npz file. The column headers were pickled so they could be added back to the matrix.