Dan Burke

Dburke01@syr.edu

Homework 3

Count Vectorization

04/25/2023

|  |  |
| --- | --- |
| The Mystery of Satoshi Nakamoto | |
| **Introduction** | Sentiment analysis has a multitude of economic applications with text data both on the side of the consumer and that of the producer. This report attempts to address one application of sentiment analysis within the scope of restaurant reviews.  This report intends to apply the Naïve Bayes approach to restaurant review text data sourced from Trip Advisor and accessed via Kaggle. This data possesses sentiment, review title and full review features, however for this particular use only the sentiment labels and the full review.  Naïve Bayes is a probabilistic machine learning process utilizing the Bayes’ theorem. This process assumes that all features (columns) of the data set are conditionally independent. Within the scope of this report the features refer to induvial words; all individual words contained within the dataset. |
| **Analysis** | data preparation The restaurant review data has been provided by Trip Advisor and sourced from Kaggle. This data was initially contained and imported directly into a data frame from a comma separated file using a custom method written specifically for this purpose.    Figure Custom Importation and Data Formatting Method    Figure Output of Custom Importation and Cleaning Method  Though this data was contained within a clean data frame, it was required to remove the “review\_title” column and change the “sentiment” feature (and values contained) to a categorical data type. A Categorical datatype is required as Naïve Bayes’ models cannot ingest string datatypes.  Due to the nature of a data frame this data was not in the form required for count vectorizing. The first step after importation was to separate the labels (“sentiment” feature/column within data set) from the desired feature to derive Naïve Bayes coefficients (“full\_review” feature/column).    Figure Sentiment Feature set to Categorical    Figure Newly Formatted Two Feature Data Frame  .  With this new two feature/column data frame, it had become necessary to count vectorize our “review\_full” feature, that being aggregating all unique words contained within the data and counting their occurrences. models and methods AND Parameters This effort’s central focus is to generate a classifying model via the Naïve Bayes Classifier (leveraging SkLearn). Previously, the labels and feature data (“review\_full”) was extracted from each data set and concatenated into a single data frame. The next step was to count vectorize and separate the labels from the data; yet maintaining relative indexes between each data row and label (show in figure below).      Figure Test Train, Count Vectorize, Extract Labels  Once the labels and count vectorized data had been stored. It was necessary to split the data into test and train sections. This effort used an 80/20 split: 80 percent for training 20 percent for testing, as shown below.    Figure 6 Test Train Split  Now the data has been fully prepared for fitting within a Naïve Bayes model. This requires the instantiation of a model variable leveraging the “fit()” method with “X\_train” and “y\_train” hyper parameters. At the completion of fitting the data to the model, “y\_pred” variable is used to store the results of running the model’s predict method with parameters of “X\_test”. “X\_test” contains 20 percent of our data reserved for testing, generated previously by the SkLearn “train\_test\_split()” method.    Figure Model Generation and Prediction |
| **results** | technical results Resulting data frame after importation of textual data, cleaning, removal of special characters and count vectorizing.    Figure Resulting 3 Document Count Vectorized Data Frame  . Resulting Multinomial Naïve Bayes Model Prediction Accuracy:    Figure Multinomial Naive Bayes Accuracy  Figure Multinomial Naive Bayes Model Confusion Matrix  **Top 20 Most Deterministic Features (Words):**  Figure Top 20 Deterministic Features |
| **conclusions** | The Multinomial Naïve Bayes approached proved to produce results of exceptional accuracy ~95%. Though this figure is impressive there are a multitude of means for adjusting and increasing accuracy and sensitivity.  Should this effort be taken up again, more detailed care could be taken within the preprocessing steps, most notably stop words, limiting language to English characters and the removal of numerical values (such as prices denoted within reviews). Though not all review possessed numerical text (prices) it did prove to be a feature which was not common amongst most reviews and could increase computational complexity (more text to count vectorize) it introduced ambiguity in context which in further analysis (binoms/trinoms) may potential induce error. |