COL341 : Machine Learning

Naive Bayes: Amazon Review Classification

Priyank | 2016MT10628

Definitions:

Stemming:
In linguistic morphology and information retrieval, stemming is the process for reducing inflected (or sometimes derived) words to their stem, base or root form—generally a written word form. The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. Algorithms for stemming have been studied in computer science since the 1960s. Many search engines treat words with the same stem as synonyms as a kind of query expansion, a process called conflation.
(Definition from: https://textminingonline.com/dive-into-nltk-part-iv-stemming-and-lemmatization)

Lemmatization:
Lemmatization (or lemmatization) in linguistics, is the process of grouping together the different inflected forms of a word so they can be analysed as a single item.
In computational linguistics, lemmatization is the algorithmic process of determining the lemma for a given word. Since the process may involve complex tasks such as understanding context and determining the part of speech of a word in a sentence (requiring, for example, knowledge of the grammar of a language) it can be a hard task to implement a lemmatizer for a new language.
In many languages, words appear in several inflected forms. For example, in English, the verb 'to walk' may appear as 'walk', 'walked', 'walks', 'walking'. The base form, 'walk', that one might look up in a dictionary, is called the lemma for the word. The combination of the base form with the part of speech is often called the lexeme of the word.
Lemmatization is closely related to stemming. The difference is that a stemmer operates on a single word without knowledge of the context, and therefore cannot discriminate between words which have different meanings depending on part of speech. However, stemmers are typically easier to implement and run faster, and the reduced accuracy may not matter for some applications.
(Definition from: https://textminingonline.com/dive-into-nltk-part-iv-stemming-and-lemmatization)

Stopwords:
A stop word is a commonly used word (such as "the", "a", "an", "in") that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query.
(Definition from: https://www.geeksforgeeks.org/removing-stop-words-nltk-python/)

N-gram:
In the fields of computational linguistics and probability, an n-gram is a contiguous sequence of n items from a given sample of text or speech. The items can be phonemes, syllables, letters, words or base pairs according to the application. The n-grams typically are collected from a text or speech corpus.
(Definition from: https://en.wikipedia.org/wiki/N-gram)

Unigram & Bigram:
When n = 1, it is called unigram and when n = 2, it is called bigram:

Experimenting with various methods:

Various cleaning methods are mentioned below:

1. The string review contained some HTML syntax still remained so some cleaning was done. The HTML present was **'"'** which represents '"' in HTML. This was changed to ' '(a blank space). All the symbols and characters were then removed i.e., **symbols** like '.', ',', '/', '\', '!' etc. All the words(alphabets) were converted into lower case. These were removed using the string library containing ascii_letters. Then the review just consisted of words. Then it was split up using .split(' ') command in python. And removing **stop words** and words of length less than 3.
2. **Method 1** along with performing **stemming** of the words to make the dictionary for the training dataset.
3. **Method 1** along with performing **lemmatization** of the words to make the dictionary for the training dataset.
4. **Method 2** and **method 3** combined in two ways:
    a. First the word was **stemmed** and after stemming it was **lemmatized**.
    b. First the word was **lemmatized** and after lemmatizing it was **stemmed**.
5. **Method 1** along with **method 2** consecutive words(bigrams) were made i.e. the dictionary for the training dataset consisted of unigrams and bigrams.
6. **Method 5** along with performing stemming of the words to make the dictionary for the training dataset.
7. **Method 5** along with performing lemmatization of the words to make the dictionary for the training dataset.
8. **Method 6** and **method 7** combined in two ways:
    a. First the **unigram** and **bigram** was stemmed and after stemming it was lemmatized.
    b. First the **unigram** and **bigram** was lemmatized and after lemmatizing it was stemmed.
9. **Method 1** was used for cleaning and then only 2 consecutive words were considered i.e., only bigrams.
10. **Method 10** along with performing **stemming** of the words to make the dictionary for the training dataset.
11. **Method 10** along with performing **lemmatization** of the words to make the dictionary for the training dataset.
12. **Method 11** and **method 12** combined in two ways:
    a. First the **bigram** was **stemmed** and after stemming it was **lemmatized.**
    b. First the **bigram** was **lemmatized** and after lemmatizing it was **stemmed**.

The above methods resulted in following results:

Method	Time	Accuracy	Macro F-score
1	0m26.874s	0.5940532855288372	0.2367680484033472
2	2m9.868s	0.5908388452741546	0.24056064375206768
3	0m55.238s	0.5907152129566668	0.2352818565064215
4a	2m39.567s	0.5910242937503863	0.24065481979463996
4b	2m37.092s	0.5904679483216912	0.240375294638711
5	1m0.833s	0.5564072448538048	0.16608133086704066
6	4m25.853s	0.555665450948878	0.1658934226834042
7	1m50.783s	0.5559127155838536	0.16506219834202465
8a	5m40.712s	0.5564072448538048	0.16608133086704066

8b	5m41.645s	0.5564072448538048	0.16608133086704066
9	0m54.847s	0.5580762811398899	0.17294196210414725
10	4m34.601s	0.5580762811398899	0.17294196210414725
11	4m32.408s	0.5580762811398899	0.17294196210414725
12a	5m39.134s	0.5580762811398899	0.17294196210414725
12b	5m30.612s	0.5580762811398899	0.17294196210414725

From the above experiments it was clear that **Method 1** is giving the best accuracy while **method 4a** is giving best Macro F-score.