Text Classification with Naive Bayes

20221009 GKS Kim Eunhong

1. Introduction

Text classification is a natural language processing (NLP) task that involves assigning predefined categories or labels to text documents. In this implementation, I construct the Naive Bayes classifier, a probabilistic algorithm based on Bayes' theorem, which assumes that the features (words in this case) are conditionally independent. Naive Bayes models are a group of extremely fast and simple classification algorithms that are often suitable for very high-dimensional datasets. Because they are so fast and have so few tunable parameters, they end up being very useful as a quick-and-dirty baseline for a classification problem. This text classification systems also lies the Naive Bayes algorithm, a probabilistic model rooted in Bayes' theorem.

Bayesian Foundation: Bayesian methods are grounded in probability theory, particularly Bayes' theorem, which calculates the probability of a hypothesis given observed evidence. In the context of text classification, the hypothesis is the category to which a document belongs, and the evidence is the content of the document itself.

Naive Bayes Assumption: The "naive" in Naive Bayes arises from the assumption of conditional independence among features, in this case, the words in a document. Despite this simplification, Naive Bayes has proven remarkably effective for text classification tasks, demonstrating its resilience and efficiency in the face of large and diverse datasets.

2. Implementation

2.1 Data Fetching:
I used the `fetch_20newsgroups` dataset from scikit-learn, which contains a collection of approximately 20,000 newsgroup documents across 20 different categories.

2.2 Vocabulary Creation:
I constructed a vocabulary set and make into a list from the training dataset.

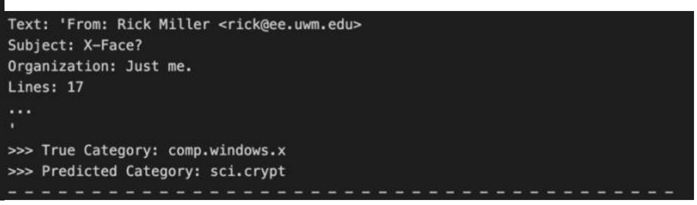
2.3 Model Training:
For each category, Laplace smoothing was applied to the word counts, and the probabilities were computed. Log probabilities were then calculated for each word, providing a more stable measure for very small probabilities.

2.4 Model Testing:
The trained model was applied to the test dataset. For each test sample, log probabilities for each category were computed, and the category with the maximum log probability was selected as the predicted category.

```
Text: 'From: v064mb9k@ubvmsd.cc.buffalo.edu (NEIL B. GANDLER)
Subject: Need info on 88-89 Bonneville
Organization: University at Buffalo
Lines: 10
News-Software: VAX/VMS VNEWS 1.41
Nntp-Posting-Host: ubvmsd.cc.buffalo.edu


 I am a little confused on all of the models of the 88-89 bonnevilles.
I have heard of the LE SE LSE SSE SSEI. Could someone tell me the
differences are far as features or performance. I am also curious to
know what the book value is for prefereably the 89 model. And how much
less than book value can you usually get them for. In other words how
much are they in demand this time of year. I have heard that the mid-spring
early summer is the best time to buy.

                        Neil Gandler
'
>>> True Category: rec.autos
>>> Predicted Category: rec.autos
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```



```
Text: 'From: Rick Miller <rick@ee.uwm.edu>
Subject: X-Face?
Organization: Just me.
Lines: 17
...
'
>>> True Category: comp.windows.x
>>> Predicted Category: sci.crypt
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

*Example

3. Performance Evaluation
3.1 Accuracy:
The accuracy is a measure of the proportion of correctly predicted labels among all test samples. The measured model accuracy was about 76%.

3.2 Confusion Matrix Visualization:
A confusion matrix was generated to provide a detailed breakdown of correct and incorrect predictions across different categories. The confusion matrix was visualized as a heatmap using seaborn.