University of Macau

澳門大學
UNIVERSIDADE DE MACAU
UNIVERSITY OF MACAU

CISC3025 Natural Language Processing Project #2

by

WONG KAI YUAN

Student No: DC026157

Project Supervisor

Prof Derek, Fai Wong

08 March 2024

Table of Contents

1) Introduction

This report is about my journey in creating and testing a simple yet powerful computer program, known as a Naïve Bayes (NB) classifier, to sort text into categories using a dataset from Reuters news articles. My goal was to see how well this program could organize texts into five specific topics ('crude', 'grain', 'money-fx', 'acq', and 'earn'), by looking at the most common words in each article and making smart guesses on where they fit best. Thereby contributing to the broader field of natural language processing (NLP) and machine learning (ML).

Part of the Reuters collection, which is filled with news stories sorted into various topics, to try my text sorting tool. To keep things manageable, only focused on five topics, which still provided me with a good challenge. My main tasks were to pre-process the text data effectively, select informative features through frequency analysis, compute the necessary probabilities for the NB algorithm, classify unseen documents accurately, and evaluate the model's performance using the F1 score.

I'm happy to share that I built a full system for sorting text, starting from cleaning up the data all the way to checking how good my program was. The Naïve Bayes classifier did a great job, showing that it's a reliable tool for putting documents into the right categories. This project didn't just prove that the approach works well in in text classification task; it also gave me valuable insights into what makes the Naïve Bayes method effective and where it might have its limits when dealing with real-life information.

The rest of this report will give you a closer look at how I went about this project, the hurdles I faced and how I overcame them, and a deep dive into my findings. I'll share what worked, what didn't, and what I took away from the whole experience.

2) Description & Results

A) Methodology and Approach

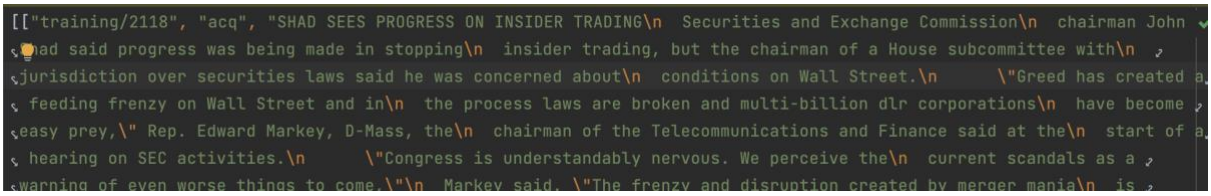
The project's methodology followed several key phases: data preprocessing, feature selection, probability calculation, document classification, and performance evaluation. Each step was built on the previous one, using the Naïve Bayes algorithm to do text classification.

Data Preprocessing: The preprocessing step was crucial for preparing the data for analysis. Using Python and the Natural Language Toolkit (NLTK), I implemented the following operations in the **preprocess** function:

- Conversion of all letters to lowercase to standardize the text.
- Removal of punctuation and special characters, utilizing regular expressions, to reduce noise in the text.
- Tokenization of sentences into words using **nltk.word_tokenize()** to break down the text into manageable pieces.
- Stemming of words to their root form with **nltk.PorterStemmer** to consolidate different forms of a word into a single item.

This preprocessing ensured that the classifier could recognize variations of the same word as identical, simplifying and streamlining the subsequent analysis steps.

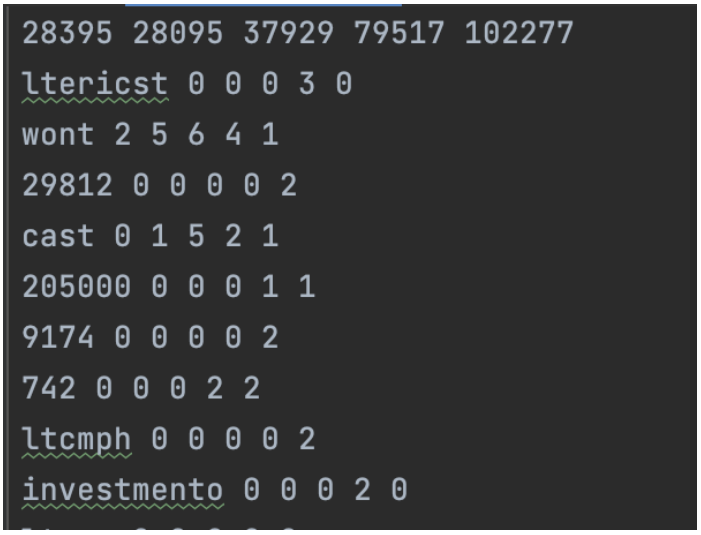
Before Preprocessing :



After Preprocessing :

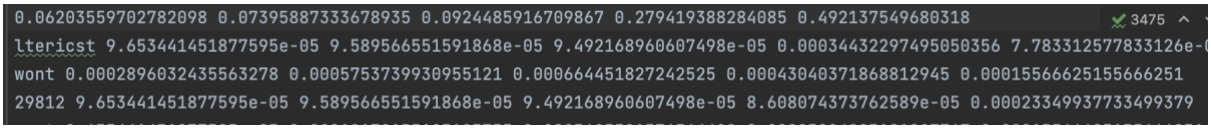
Feature Selection: Given the vast number of unique words in the corpus, a method for selecting informative features was necessary to manage computational resources effectively and improve model accuracy. The **count_word** function calculated the frequency of each word across the corpus. Then, in the **feature_selection** function, I chose the top 10,000 words by frequency as features. This threshold was a balance between computational efficiency and retaining meaningful information for classification.

Output of feature selection *:*



Probability Calculation: The **calculate_probability** function handled the computation of both prior probabilities for each class and conditional probabilities for each feature word given a class. I applied Laplace smoothing to address the issue of zero probabilities for unseen words, ensuring that every word had a non-zero chance of being associated with each category.

Output of Probability Calculation *:*



Document Classification: In the **classify** function, I employed the Naïve Bayes algorithm to calculate the posterior probability of each class for a given document and assigned the document to the class with the highest probability. This step was crucial for the practical application of the classifier to unseen data.

***Output of classification**:*

```
test/14975 earn
test/21067 crude
test/20081 money-fx
test/21040 acq
test/16228 earn
test/18689 crude
test/21462 acq
test/16833 acq
test/15255 acq
test/20548 earn
test/19325 acq
```

Performance Evaluation: Finally, the model's performance was evaluated using the F1 score, a harmonic mean of precision and recall. This metric was chosen for its balance between the classifier's ability to correctly identify documents of a certain class (precision) and its ability to identify all documents of that class within the dataset (recall).

B) Challenges and Solutions

One big problem was dealing with the huge number of different words, which made the program complicated and slow. I solved this by only focusing on the most common words (10000), which made things much simpler and faster without losing much accuracy. This strategy was implemented in the **feature_selection** function.

Another issue was that sometimes, the program could end up thinking a word had zero chance of appearing in any topic, especially if that word wasn't in the training data. To fix this, I used a method called Laplace smoothing within the **calculate_probability** function, which made sure every word had at least a tiny chance of appearing in each topic.

C) Achievements and Results

The Naïve Bayes classifier achieved an F1 score of 0.9133, indicating a high level of accuracy in classifying documents into the correct categories. This result underscores the effectiveness of the Naïve Bayes algorithm for text classification tasks, particularly when combined with careful pre-processing and feature selection.

```
F1 Score: 0.9133470440969571

Process finished with exit code 0
```

3) Conclusion

The journey of developing and testing a Naïve Bayes classifier on the Reuters news articles has been enlightening, showing just how powerful and practical text sorting can be in the tech-savvy world of natural language processing. The standout result of this project was creating a program that could accurately and reliably do text classification task, as highlighted by an impressive F1 score.

A) Accomplishments and Learnings

The primary accomplishment of this project is the comprehensive implementation of a text classification pipeline, from preprocessing and feature selection to classification and evaluation. Through this process, we learned the importance of each step in contributing to the overall performance of the classifier. Specifically, the preprocessing and feature selection phases were crucial in managing the dataset's complexity and ensuring that the Naïve Bayes algorithm could perform efficiently and effectively.

I also saw first-hand the power of probability in dealing with the unpredictable nature of language. Using Naïve Bayes, I could make smart guesses about text categories based on word use, and techniques like Laplace smoothing helped me navigate the tricky waters of words that didn't show up in the training data.

B) Reflections and Future Work

Reflecting on the project, one area for potential improvement is the exploration of alternative feature selection methods. While frequency-based selection proved effective, examining other techniques, such as TF-IDF (Term Frequency-Inverse Document Frequency) or mutual information, might reveal additional insights into feature relevance and classification performance.

Additionally, experimenting with different models, such as Support Vector Machines (SVM) or deep learning approaches, could provide comparative benchmarks and further enhance the text classification capabilities.

For future work, expanding the scope to include a larger set of categories from the Reuters dataset—or applying the methodology to other text corpora—would test the model's scalability and adaptability to different domains.

In conclusion, this project not only achieved its goal of implementing a highly accurate Naïve Bayes classifier for text classification but also contributed valuable learnings and identified promising directions for future research in the field. The experience underscores the potential of machine learning and natural language processing to extract meaningful insights from textual data, paving the way for advancements in information retrieval, content analysis, and beyond.