Assignment #5: Predict the Author (Text Classification)

CSCI 374 Fall 2019 Oberlin College

Due: Tuesday November 19 at 3:00 PM

Background

In this assignment, your goal is to write a program capable of competing in a simplified game of Jeopardy!® where the only category asks for the author of a given passage originally written by a famous author. For your program, you will train and test Naïve Bayes as a text classifier using text downloaded from Project Gutenberg. In particular, you will download popular, famous books from 10 authors, train Naïve Bayes to learn the writing styles (indicated by word choices) of each author from those texts, then predict which author wrote 50 short passages (taken from different texts than those you used for training).

Students may work in groups of up to two members for this assignment.

Gitting Started

To begin this assignment, you please follow this link:

https://classroom.github.com/a/pisbwq_c

Data Set

You need to download the TXT files of the following books from Project Gutenberg at https://www.gutenberg.org/wiki/Main_Page:

1. *Pride and Prejudice* by Jane Austen
2. *Alice's Adventures in Wonderland* by Lewis Carroll
3. *Great Expectations* by Charles Dickens
4. *The Adventures of Sherlock Holmes* by Arthur Doyle
5. *The Odyssey* by Homer
6. *The Trial* by Franz Kafka
7. *The Republic* by Plato
8. *Anna Karenina* by Leo Tolstoy
9. *The War of the Worlds* by H.G. Wells

Additionally, you should also find two texts by another author of your choice. Pick one of those two texts (preferably the larger of the two) as a tenth book to include with the nine listed above. These nine books listed above (plus the one you chose as a tenth) will serve as the training set for your machine learning with Naïve Bayes. The other book you chose will be part of your testing set (described below).

Note 1: you should treat each book as a **single training instance** in Naïve Bayes. Otherwise, if you use paragraphs, sentences, etc. as training instances, then the label prior probabilities $P(author)$ will be skewed by the length of the books written by each author, which isn't helpful when our goal is to predict who authored a single passage.

Note 2: from these ten books used for training, you will want to manually remove the additional text added by Project Gutenberg located at the beginning and end of each file so that you are only learning from the original text by the author (or its translation by another author).

Program

Your assignment is to write a program called **authors** that behaves as follows:

1) It should take in one parameter:
    a. The path to a file containing the *test* set
2) First, the program should read in the ten previously identified books to use for training. You can hardcode the names of these files in your program, but please use *relative* paths (e.g., "Austen_PridePrejudice_1342.txt") and **do not** use *absolute* paths (e.g., "C:/Users/Adam/CSCI374/Austen_PridePrejudice_1342.txt"). Your program should look for the training books in the same directory as your program.
3) The text of each book should be preprocessed to make it appropriate for training with Naïve Bayes:
    a. The text should be split into a list of words (or multiple lists of words, one per paragraph/sentence/however you wish)
    b. Each word should be converted to lower case so that capitalization is ignored
    c. Stop words should be removed (e.g., a, an, the). You can choose your own stop words (feel free to search the internet for a list, just remember to cite your source in your code and README)
    d. Remaining words should be "stemmed" using the StemmingUtil code that I will provide on GitHub
4) The text of the test set should also be read in and preprocessed the same as with the training data in Step 3.
5) The stemmed words from each book should be fed into Naïve Bayes to learn models of the writing styles of each author (where the label for your data is the author of the text). You should use appropriate pseudocounts, as well as use the logarithms of probabilities (instead of multiplying the probabilities themselves, which can cause underflow issues).
6) The learned model should be evaluated using the provided test set (with your additions for your chosen tenth author, see below).
7) The confusion matrix counted during Step 6 should be output as a file with the name results_authors.csv

I will provide you with code to: (1) build a list of keywords from a String of text, and (2) create a list of stemmed words from a list of words.

Test File Format

The format of the test file (called TestSet_Passages.txt) is as follows:

```
################################################################################
Label
Passage
################################################################################
Label
Passage
################################################################################
etc.
```

You will need to be able to read in the test instances between the #### lines, where the first line is label (the **last name** of the actual author of the passage), and the second line is the passage to be tested.

You should add 5 passages from the second book (*not* the one used in training) written by your chosen author to this test set so that you can also evaluate the ability of your program to predict passages written by your chosen author.

Programming Languages

Due to the need for natural language processing capabilities such as word stemming that I will provide, you should complete this assignment using either the **Java** or **Python** programming languages.

Research Questions

Please use the results of your program to answer these questions and record your answers in a README file:

1) Evaluate the overall **accuracy** of your program:
    a. What accuracy did you observe? Calculate a 95% confidence interval around this accuracy.
    b. Consider another hypothetical program that simply randomly guesses the author from amongst the ten authors. What would its expected predictive accuracy be? What would be the 95% confidence interval around this random program's accuracy?
    c. Compare your program's observed accuracy to the hypothetical random program. Is your program significantly better? What does this imply about the learning performed by your program?
    d. What suggestions do you have for how we could further improve the accuracy of our text classification program? What other information could it incorporate, and/or how could the program be changed to be more accurate?

2) Evaluate the **recall** of your program:

    a. Calculate your program's recall for each author. As a reminder, the recall of your program is measured as:

    $$Recall_A = \frac{\#\text{ of passages by author } A \text{ correctly predicted}}{\#\text{ of passages by author } A}$$

    You **do not** need to calculate a 95% confidence interval since the number of instances is too small to give a meaningful interval.

    b. Compare the recalls between the ten authors. Which authors did your program best identify? Which authors did it most struggle to identify?

3) Evaluate the **precision** of your program:

    a. Calculate your program's precision for each author. As a reminder, the precision of your program is measured as:

    $$Precision_A = \frac{\#\text{ of passages by author } A \text{ correctly predicted}}{\#\text{ of passages predicted to be author } A}$$

    Once again, you **do not** need to calculate a 95% confidence interval since the number of instances is too small to give a meaningful interval.

    b. Compare the precisions between the ten authors. Which authors did your program predict correctly the most often? Which authors did it predict too frequently when it shouldn't have?

    c. Were there any trends observed between your recalls and precisions for the authors? Did a high or low recall tend to correlate with a high or low precision?

4) Investigate your results. For the authors for whom your program made incorrect predictions, were there any trends that you observed? That is, did your program tend to confuse two or more authors, thinking that they were similar? If so, does this confusion make sense given what you know about those authors (e.g., their time period, their location, etc.)? What other trends or important results did you observe? Why might these have happened?

README

Within a README file, you should include:

1) Your name(s),
2) Your answers to the questions above,
3) A short paragraph describing your experience during the assignment (what did you enjoy, what was difficult, etc.)
4) An estimation of how much time you spent on the assignment, and
5) An affirmation that you adhered to the honor code

Please remember to commit your solution code, results files, and README file to your repository on GitHub. **Please also commit your 10 books used for training, as well as your final test file.** You do not need to wait to commit your code until you are done with the assignment; it is good practice to do so not only after each coding session, but maybe after hitting important milestones or solving bugs during a coding session. ***Make sure to document your code***, explaining how you implemented the different components of the assignment.

Honor Code

Each student is allowed to work with a partner to complete this assignment. Groups are also allowed to collaborate with one another to discuss the abstract design and processes of their implementations. For example, please feel free to discuss the process of reading in the texts from Project Gutenberg and preprocessing the instances for Naïve Bayes. However, sharing code (either electronically or looking at each other's code) between groups is not permitted.

Grading Rubric

Your solution and README will be graded based on the following rubric:

Followed input and output directions: /5 points
Properly read in and preprocessed the data: /15 points
Correctly implemented the Naïve Bayes learning algorithm: /12 points
Used pseudocounts and logarithms of probabilities: /8 points
Correctly implemented classification: /10 points
Correctly answered the research questions: /40 points
Provided requested README information: /5 points
Appropriate code documentation: /5 points