CIS 678 Machine Learning
Programming Project #1

Exploratory Data Analysis

Overview

This is intended to be a "warm-up" exercise. The main purpose of the assignment is to refresh your data management and programming skills; to reinforce experimental thinking; and to gain experience with interpretation and discussion of results.

Background

Authors and editors are often concerned with the "readability" of their articles and manuscripts: how difficult is a sentence to understand? is this document written at a level appropriate for its target audience? As a student author, you might wish to determine the cognitive complexity of your term paper or thesis. This leads to the question: are there any metrics for measuring the concept of readability?

One way to approach this problem is by computing the Flesch Index (FI), a numerical measure of the readability of English text. Originally invented for evaluating the difficulty of reading U.S. Army training manuals, it has since become ubiquitous.

Specifications

Acquire sample documents, analyze them using the Flesch Index, visually communicate your results, and discuss/interpret any experiments performed.

1. Pre-processing: read in and process the data

Several datafiles are posted to BlackBoard. They consist of text documents downloaded from Project Gutenberg, a free repository of digital documents. As you parse each file you will need to properly handle punctuation, non-alphabetic characters and whitespace. Be sure to document all design decisions. For example, the Flesch Index depends on knowing the number of sentences in a document – how do you define a "sentence" and how does your program determine that number?

2. Analysis: compute the Flesch Index (FI) of the data

The first step in the computation is to break the document down into syllables, words and sentences. There are two basic ideas:

- multisyllabic words are more complex than simple words
- sentences with a large number of words (requiring a higher cognitive load to keep everything in memory) are more complex than shorter sentences.

These two intuitions are captured in the succinct equation below:

$$\text{Flesch\_Index} = \quad 206.835 - 84.6\left(\frac{numSyllables}{numWords}\right) - 1.015\left(\frac{numWords}{numSentences}\right)$$

where *numSyllables* is the total number of syllables in the document, *numWords* is the total number of words and *numSentences* is the total number of sentences.

A high Flesch Index (~100) indicates text that is easy to read. In the equation above, notice that a higher ratio of syllables to words generates a larger term to be subtracted from the base, resulting in a lower index, thus indicating a document that is more difficult to read.

Here are some simple rules to help compute the required numeric values. There are numerous variations that you can experiment with, but be sure to apply your rules consistently.

Sentence:
Consider a sentence to have been encountered whenever you find a word that ends in a specific punctuation symbol: e.g. a period or question mark.

Word:
A word is a contiguous sequence of alphabetic characters. Typically, whitespace is used to define word boundaries.

Syllable:
The usual English definitions of vowels and consonants applies. A syllable is considered to have been encountered whenever you detect:

- Rule 1: a vowel at the start of a word *or*
- Rule 2: a vowel following a consonant in a word.
  With the exception of the word "the", a lone 'e' at the end of a word does not count as a syllable.

Note that the Wikipedia page includes a table for translating the Flesch Index into the corresponding grade level.

3. Visualization: display the data/results

Put some thought into visually understanding your data. For example, create a histogram showing the distribution of polysyllabic words in a document (i.e. what percentage of words contain *k* syllables?). Use a chart to show how different documents (or different metrics) compare. Be creative as you explore the data and your results.

Try to do your visualizations from within your program, using an available graphics library (e.g. matplotlib, seaborn, ggplot). The idea is to improve your skills at visually exploring and communicating.

4. Experimentation: what does the data tell you

Find your own data (the Project Gutenberg website contains thousands of documents in text format) and design an experiment. For example:

- Has readability changed over time (are today's books easier to read than those written in the 1700's)?
- How does a novel compare to a textbook?
- How do your results compare with those computed by Microsoft Word?

Provide a description of your experiments, an explanation of your results, anomalies detected and any conclusions you were able to reach.

Deliverables

- You must use Python and all computations must be performed by your program.
- Be sure to demonstrate good programming style and practices.
- Submit a **single PDF**, formatted as a Report. The report should describe your general approach, design decisions, problems encountered, experiments conducted and your results/analysis/conclusions. Include source-code, sample output and visuals.
- Be prepared to present and discuss your solution in class.