

Introduction to Toy Datasets

Applied Machine Learning for Educational Data Science

true

06/28/2021

Contents

Readability 1

Recidivism 3

[Updated: Mon, Jun 28, 2021 - 17:23:49]

There are two datasets we will analyze throughout the whole course. The first dataset has a continuous outcome and the second dataset has a binary outcome. We will apply several methods and algorithms to these two datasets during the course. This will give us an opportunity to compare and contrast the prediction outcomes from several models and methods on the same datasets. This section provides some background information and context for these two datasets.

Readability

The readability dataset comes from a recent [Kaggle Competition \(CommonLit Readability Prize\)](#). You can directly download the training dataset from the competition website, or you can import it from the course website.

```
readability <- read.csv('https://raw.githubusercontent.com/uo-datasci-specialization/c4-ml-fall-2021/main/readability.csv',
                        header=TRUE)

str(readability)
```

There is a total of 2834 observations. Each observation represents a reading passage. The most important variables are the `excerpt` and `target` columns. The `excerpt` column includes a plain text data and the `target` column includes a corresponding measure of readability for each excerpt.

```
readability[1,]$excerpt
readability[1,]$target
```

According to the data owner, 'the target value is the result of a Bradley-Terry analysis of more than 111,000 pairwise comparisons between excerpts. Teachers spanning grades 3-12 served as the raters for these comparisons.' A higher target value indicates a more difficult text to read. The purpose is to develop a model that predicts a readability score for a given text to identify an appropriate reading level.

We will not consider the standard error variable in our models although it has a strong relationship with the target outcome because the standard errors would not be available for new observations we would like to predict. There may be creative ways to make use of standard error in a multi-step prediction model (e.g., develop a separate prediction model for standard errors in the first step, and then use the predicted standard errors to predict target scores in the second step); however, we will not get into that in this course.

In the following weeks, we will cover how to generate features from plain text data and whether or not these features can successfully predict the target scores. These features will include [universal POS tags](#), [morphological features](#), [syntactic annotations](#), and some other simple text features (e.g., number of words, number of syllables). You will need to install the following packages for the following weeks:

- [udpipe](#)
- [quanteda](#)
- [quanteda.textmodels](#)

```
install.packages(pkgs = c('udpipe', 'quanteda', 'quanteda.textmodels'),  
                 dependencies = TRUE)
```

In addition, we will also be exposed a little bit to the world of Natural Language Processing (NLP) through some pre-trained language models (e.g., [RoBerta](#)). Our coverage of this material will be at the surface level. We will primarily cover how we can derive numerical sentence embedding from a pre-trained language model using Python through R. If you have time, [this series of Youtube videos](#) provide some background and accessible information about these models. In particular, Episode 2 will give a good idea about what these numerical embeddings are. For part of feature generation, we will use [reticulate](#), an R interface to Python, to access a number of Python modules.

You can run the following code in your computer to get prepared for the following weeks. Note that you only have to run the following code once to create a Python environment and install the necessary packages.

```
# Install and load the reticulate package  
  
install.packages(pkgs = 'reticulate',  
                 dependencies = TRUE)  
  
require(reticulate)  
  
# Install Miniconda  
  
install_miniconda()  
  
# Create a virtual Python environment  
  
virtualenv_create("my.python")  
  
# Install the Python modules  
  
conda_install(envname = 'my.python',  
              c('torch', 'transformers', 'numpy', 'nltk', 'tokenizers'),  
              pip = TRUE)
```

Once you create a virtual Python environment and install the packages using the code above, you can run the following code. If you are seeing the same output as below, you should be all set to explore some very exciting NLP tools using the Readability dataset.

```
require(reticulate)

# List the available Python environments

virtualenv_list()

# Import the modules

reticulate::import('torch')
reticulate::import('numpy')
reticulate::import('transformers')
reticulate::import('nltk')
reticulate::import('tokenizers')
```

```
[1] "my.python"
Module(torch)
Module(numpy)
Module(transformers)
Module(nltk)
Module(tokenizers)
```

Recidivism