Introduction to Toy Datasets

Applied Machine Learning for Educational Data Science

true

06/28/2021

Contents

[Updated: Fri, Jul 02, 2021 - 14:41:29]

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/ma
                        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 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 virtual 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

The Recidivism dataset comes from The National Institute of Justice's (NIJ) Recidivism Forecasting Challenge. The challenge aims to increase public safety and improve the fair administration of justice across the United States. This challenge had three stages of prediction, and all three stages require to model a binary outcome (recidivated vs. not recidivated in Year 1, Year 2, and Year 3). In this class, we will only work on the second stage and develop a model for predicting the probability that an individual will be recidivated in the second year after initial release.

You can directly download the training dataset from the competition website, or you can import it from the course website. Either way, please make sure you read the Terms of Use at this link before working with this dataset.

Note that the competition website has also test datasets for Year 1, Year 2, and Year 3; however, we will not use them because they don't have the outcome. These datasets are being used for competition. Participants were supposed to assign a probability at each stage and submit their predictions for internal performance evaluation. Therefore, we will not be able utilize these test datasets.

```
recidivism_train <- read.csv('https://raw.githubusercontent.com/uo-datasci-specialization/c4-ml-fall-20

str(recidivism_train)
```

There are 18,028 observations in the training set and 53 variables including a unique ID variable, and four potential target variables (Recidivism in Year 1, Recidivism in Year 2, and Recidivism in Year 3). The remaining 49 variables are potential predictive features and include variables such as gender, race, age at release, gang affiliation, etc. A full list of these variables can be found at this link.

We will work on developing a model to predict the target variable `Recidivism_Arrest_Year2` using the 49 potential predictive variables. Before moving forward, we have to remove the individuals who had already been recidivated in Year 1. As you can see below, about 29.8% of the individuals were recidivated in Year 1. I am removing these individuals from the original training dataset and save the new dataset for later use in class.

```r
table(recidivism_train$Recidivism_Arrest_Year1)

recidivism_train2 <- recidivism_train[recidivism_train$Recidivism_Arrest_Year1 == 'false',]

write.csv(recidivism_train2,
          here('data/recidivism_train_y1 removed.csv'),
          row.names = FALSE)
```