Introduction to Toy Datasets

Applied Machine Learning for Educational Data Science

true

06/28/2021

Contents

[Updated: Sat, Oct 23, 2021 - 12:46:20]

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 lower target value indicates a more difficult text to read. The highest target score is equivalent of the 3rd grade level while the lowest target score is equivalent of the 12th grade level. 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).

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.

You will need to install the following packages for the following weeks:

- udpipe
- quanteda
- quanteda.textstats
- text

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

# Make sure to install the developer version of the text package from Github

# install.packages("devtools")

devtools::install_github("oscarkjell/text")
```

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 install the necessary packages.

```
# Install and load the reticulate package

install.packages(pkgs = 'reticulate',
                 dependencies = TRUE)

require(reticulate)

# Install Miniconda

install_miniconda()

# Install the Python modules

conda_install(envname = 'r-reticulate', 'torch', pip = TRUE)


conda_install(envname = 'r-reticulate','transformers',pip = TRUE)
```

```
conda_install(envname = 'r-reticulate','nltk',pip = TRUE)


conda_install(envname = 'r-reticulate','tokenizers',pip = TRUE)
```

Once you install the Python 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)

# Import the modules

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

# Load udpipe

require(udpipe)

# Load quanteda

require(quanteda)

# Load quanteda text stats

require(quanteda.textstats)

# Load the text package

require(text)
```

```
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.

```
recidivism <- read.csv('https://raw.githubusercontent.com/uo-datasci-specialization/c4-ml-fall-2021/mai

str(recidivism)
```

There are 25,835 observations in the training set and 54 variables including a unique ID variable, four outcome variables (Recidivism in Year 1, Recidivism in Year 2, and Recidivism in Year 3, Recidivism within 3 years), and a filter variable to indicate whether an observation was included in the training dataset or test dataset. The remaining 48 variables are potential predictive features. A full list of these variables can be found at this link.

We will work on developing a model to predict the outcome variable `Recidivism_Arrest_Year2` using the 48 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 29.9% of the individuals were recidivated in Year 1. I am removing these individuals from the original dataset

```
table(recidivism$Recidivism_Arrest_Year1)
recidivism2 <- recidivism[recidivism$Recidivism_Arrest_Year1 == FALSE,]
```

I will also do recoding of some variables before saving the new dataset for later use in class.

- First, some variables in the dataset are coded as TRUE and FALSE. When these variables are imported into R, R automatically recognizes them as logical variables. I will recode all these variables such that FALSE = 0 and TRUE = 1.

```
# Find the columns recognized as logical

  cols <- sapply(recidivism, is.logical)

# Convert them to numeric 0s and 1s

  recidivism2[,cols] <- lapply(recidivism2[,cols], as.numeric)
```

- Second, the highest value for some variables are coded as **3 or more**, **4 or more**, **10 or more**, etc. These variables can be considered as numeric, but R recognizes them as character vectors due to phrase **or more** for the highest value. We will recode these variables so 'X or more' will be equal to X.

```
require(dplyr)

# ?recode for more info

# Dependents

  recidivism2$Dependents <- recode(recidivism2$Dependents,
                                   '0'=0,
                                   '1'=1,
                                   '2'=2,
                                   '3 or more'=3)
```

```
# Prior Arrest Episodes Felony

  recidivism2$Prior_Arrest_Episodes_Felony <- recode(recidivism2$Prior_Arrest_Episodes_Felony,
                                                     '0'=0,
                                                     '1'=1,
                                                     '2'=2,
                                                     '3'=3,
                                                     '4'=4,
                                                     '5'=5,
                                                     '6'=6,
                                                     '7'=7,
                                                     '8'=8,
                                                     '9'=9,
                                                     '10 or more'=10)
# Prior Arrest Episods Misd

  recidivism2$Prior_Arrest_Episodes_Misd <- recode(recidivism2$Prior_Arrest_Episodes_Misd,
                                                   '0'=0,
                                                   '1'=1,
                                                   '2'=2,
                                                   '3'=3,
                                                   '4'=4,
                                                   '5'=5,
                                                   '6 or more'=6)

# Prior Arrest Episodes Violent

  recidivism2$Prior_Arrest_Episodes_Violent <- recode(recidivism2$Prior_Arrest_Episodes_Violent,
                                                      '0'=0,
                                                      '1'=1,
                                                      '2'=2,
                                                      '3 or more'=3)

# Prior Arrest Episods Property

  recidivism2$Prior_Arrest_Episodes_Property <- recode(recidivism2$Prior_Arrest_Episodes_Property,
                                                       '0'=0,
                                                       '1'=1,
                                                       '2'=2,
                                                       '3'=3,
                                                       '4'=4,
                                                       '5 or more'=5)

# Prior Arrest Episods Drug

  recidivism2$Prior_Arrest_Episodes_Drug <- recode(recidivism2$Prior_Arrest_Episodes_Drug,
                                                   '0'=0,
                                                   '1'=1,
                                                   '2'=2,
                                                   '3'=3,
                                                   '4'=4,
                                                   '5 or more'=5)
# Prior Arrest Episods PPViolationCharges
```

```
  recidivism2$Prior_Arrest_Episodes_PPViolationCharges <- recode(recidivism2$Prior_Arrest_Episodes_PPVi
                                                                 '0'=0,
                                                                 '1'=1,
                                                                 '2'=2,
                                                                 '3'=3,
                                                                 '4'=4,
                                                                 '5 or more'=5)

# Prior Conviction Episodes Felony

  recidivism2$Prior_Conviction_Episodes_Felony <- recode(recidivism2$Prior_Conviction_Episodes_Felony,
                                                         '0'=0,
                                                         '1'=1,
                                                         '2'=2,
                                                         '3 or more'=3)

# Prior Conviction Episodes Misd

  recidivism2$Prior_Conviction_Episodes_Misd <- recode(recidivism2$Prior_Conviction_Episodes_Misd,
                                                       '0'=0,
                                                       '1'=1,
                                                       '2'=2,
                                                       '3'=3,
                                                       '4 or more'=4)

# Prior Conviction Episodes Prop

  recidivism2$Prior_Conviction_Episodes_Prop <- recode(recidivism2$Prior_Conviction_Episodes_Prop,
                                                       '0'=0,
                                                       '1'=1,
                                                       '2'=2,
                                                       '3 or more'=3)

# Prior Conviction Episodes Drug

  recidivism2$Prior_Conviction_Episodes_Drug <- recode(recidivism2$Prior_Conviction_Episodes_Drug,
                                                       '0'=0,
                                                       '1'=1,
                                                       '2 or more'=2)

# Delinquency Reports

  recidivism2$Delinquency_Reports <- recode(recidivism2$Delinquency_Reports,
                                            '0'=0,
                                            '1'=1,
                                            '2'=2,
                                            '3'=3,
                                            '4 or more'=4)

# Program Attendances

  recidivism2$Program_Attendances <- recode(recidivism2$Program_Attendances,
                                            '0'=0,
```

```
                                          '1'=1,
                                          '2'=2,
                                          '3'=3,
                                          '4'=4,
                                          '5'=5,
                                          '6'=6,
                                          '7'=7,
                                          '8'=8,
                                          '9'=9,
                                          '10 or more'=10)

# Program Unexcused Absences

  recidivism2$Program_UnexcusedAbsences <- recode(recidivism2$Program_UnexcusedAbsences,
                                                  '0'=0,
                                                  '1'=1,
                                                  '2'=2,
                                                  '3 or more'=3)

# Residence Changes

  recidivism2$Residence_Changes <- recode(recidivism2$Residence_Changes,
                                          '0'=0,
                                          '1'=1,
                                          '2'=2,
                                          '3 or more'=3)
#################################################################

str(recidivism2)
```

Now, we can write the final version of the dataset for later use.

```
write.csv(recidivism2,
          here('data/recidivism_y1 removed.csv'),
          row.names = FALSE)
```