classecol: vignette

Thomas Frederick Johnson Github: GitTFJ

6 July 2020

classecol

classecol is a package to perform text classifications of twitter data (but may be useful for any social media text). It provides a series of functions to clean data, produces a sentiment matrix of some of the most popular methods, and has three models to classify text: biographical - to assess who the user is, environment - to assess if the tweet is related to their environment at the users stance on the enviornment, hunting - to assess if the tweet is related to hunting and determine their stance on hunting.

Intstructions to prepare package

This package is runs through R but is reliant on a python back-end (which dramitcally improves the speed of the classification). So before running any code, you will need to install a version of python - we recommend Python 3.6.9 which is what the package has been tested on; available here. You will also need to install a selection of packages on python: numpy, os, pandas, re, nltk, bs4, string, joblib, pickle, sklearn, keras, tensorflow, and time. Packages can be installed following these instructions. If at anypoint the error 'Module not found' appears, install this module/package following these ihe above instructions.

classecol is only available as a github repository so needs to be installed through github

```
library(devtools)
install_github("GitTFJ/classecol")
library(classecol)
library(reticulate)
```

classecol is not self-contained is alos reliant on a prtner repository which stores the python models and code. This section of code will download the accompanying repository and save it into a specified location

```
direc = paste(find.package("classecol"),"/models", sep = "")
download_models(direc)
```

In order to model the text file, we need to link R to the python program. 'reticulate' offers a function to perform this, but we found it performed incosistently, so require manuallly specifying pythons absolute filepath location. the file to search for is 'python.exe'.

You will also need to specify the location you have downloaded the models to and then send this location to python with the function 'r_to_py'

```
reticulate::use_python("C:/Users/mn826766/AppData/Local/Continuum/anaconda3/python.exe") #Specify your
direc = paste(direc, "/classecol-models-master/", sep = "")
model_directory = reticulate::r_to_py(direc)
```

Biographical classifier

At this point we are ready to prepare and classify the data. The biographical classifier 'bio_class()' works best with twitter data in its dirty form, so none of the text should be cleaned. However, it is neccasary to join the twitter name and description into one column named 'text' split with a space. if the column is named anything but 'text' with a dataframe named 'data' the download will fail.

```
df = data.frame(
  name = c(
    "Boris Johnson #StayAlert",
    "Thomas Frederick Johnson",
    "University of Reading"),
  description = c(
  "Prime Minister of the United Kingdom and @Conservatives leader.
  Member of Parliament for Uxbridge and South Ruislip. #StayAlert",
  "Researching carnivores and macroecology. Interested in #Ecology #Birds #LFC #Politics #DataScience",
  "Campus life and study at the University of Reading, UK. For news and comment follow @UniRdg_News"))
df$text = paste(df$name, df$description)
data = reticulate::r_to_py(df)
bio_class(
  type = "split",
  directory = direc)
```

```
## [1] "Person (not expert)"              "Expert"
## [3] "Organisation/Group/Company/Other"
```

Hunting classifier

The hunting classifier 'hunt_class()' works best with twitter data after a simple clean. if the column is named anything but 'text' with a dataframe named 'data' the download will fail.

```
df = data.frame(
  text = c(
    "I hate hunting",
    "Cant wait to go hunting",
    "Hunting for my car keys"))
df$text = classecol::clean(df$text, level = "simple")
data = reticulate::r_to_py(df)
hunt_class(
  type = "all",
  directory = direc)
```

```
## [1] "Relevant (against-hunting)" "Relevant (pro-hunting)"
## [3] "Irrelevant"
```

Environment classifier

The environment classifier 'env_class()' works best with twitter data after a full clean and also requires sentiment analysis on the text. if the column is named anything but 'text' with a dataframe named 'data' the download will fail.

```
df = data.frame(
  text = c(
    "I love walking in nature",
    "I am so sad we losing the rainforest. stop the destruction",
    "Tiger wins the PGA tour again!"))
df$text = classecol::clean(df$text, level = "full")
sm = as.matrix(cbind(
  valence(df$text),
  lang_eng(as.character(df$text)),
  senti_matrix(as.character(contract(df$text)))))
data = reticulate::r_to_py(df)
sent_mat = reticulate::r_to_py(sm)
env_class(
  type = "trim",
  directory = direc)
```

```
## [1] "Pro-wildlife (positive phrasing)" "Pro-wildlife (negative phrasing)"
## [3] "Irrelevant"
```