kaggle

josef

September 14, 2018

library(ggplot2) # Data visualization  
library(readr)   
  
  
  
# import the train dataset  
train = read.csv("train.csv")  
head(train)

## id  
## 1 0000997932d777bf  
## 2 000103f0d9cfb60f  
## 3 000113f07ec002fd  
## 4 0001b41b1c6bb37e  
## 5 0001d958c54c6e35  
## 6 00025465d4725e87  
## comment\_text  
## 1 Explanation\nWhy the edits made under my username Hardcore Metallica Fan were reverted? They weren't vandalisms, just closure on some GAs after I voted at New York Dolls FAC. And please don't remove the template from the talk page since I'm retired now.89.205.38.27  
## 2 D'aww! He matches this background colour I'm seemingly stuck with. Thanks. (talk) 21:51, January 11, 2016 (UTC)  
## 3 Hey man, I'm really not trying to edit war. It's just that this guy is constantly removing relevant information and talking to me through edits instead of my talk page. He seems to care more about the formatting than the actual info.  
## 4 "\nMore\nI can't make any real suggestions on improvement - I wondered if the section statistics should be later on, or a subsection of ""types of accidents"" -I think the references may need tidying so that they are all in the exact same format ie date format etc. I can do that later on, if no-one else does first - if you have any preferences for formatting style on references or want to do it yourself please let me know.\n\nThere appears to be a backlog on articles for review so I guess there may be a delay until a reviewer turns up. It's listed in the relevant form eg Wikipedia:Good\_article\_nominations#Transport "  
## 5 You, sir, are my hero. Any chance you remember what page that's on?  
## 6 "\n\nCongratulations from me as well, use the tools well. Â Â· talk "  
## toxic severe\_toxic obscene threat insult identity\_hate  
## 1 0 0 0 0 0 0  
## 2 0 0 0 0 0 0  
## 3 0 0 0 0 0 0  
## 4 0 0 0 0 0 0  
## 5 0 0 0 0 0 0  
## 6 0 0 0 0 0 0

names(train)

## [1] "id" "comment\_text" "toxic" "severe\_toxic"   
## [5] "obscene" "threat" "insult" "identity\_hate"

dim(train)

## [1] 159571 8

# clean the text  
library(tm)

## Loading required package: NLP

##   
## Attaching package: 'NLP'

## The following object is masked from 'package:ggplot2':  
##   
## annotate

library(SnowballC)  
corpus = VCorpus(VectorSource(train$comment\_text))  
corpus = tm\_map(corpus, content\_transformer(tolower))  
corpus = tm\_map(corpus, removeNumbers)  
corpus = tm\_map(corpus, removePunctuation)  
corpus = tm\_map(corpus, removeWords, stopwords())  
#corpus = tm\_map(corpus, stemDocument)  
corpus = tm\_map(corpus, stripWhitespace)  
  
# Creating the Bag of Words model  
dtm = DocumentTermMatrix(corpus)  
dtm = removeSparseTerms(dtm, 0.994)  
dataset = as.data.frame(as.matrix(dtm))  
head(dataset)

names(dataset)

## [1] "â€“" "â€”" "â€¢" "able"   
## [5] "absolutely" "accept" "acceptable" "accepted"   
## [9] "according" "account" "accurate" "across"   
## [13] "act" "action" "actions" "actual"   
## [17] "actually" "add" "added" "adding"   
## [21] "addition" "address" "admin" "administrator"   
## [25] "administrators" "admins" "advice" "afd"   
## [29] "ago" "agree" "allow" "allowed"   
## [33] "almost" "alone" "along" "already"   
## [37] "also" "although" "always" "american"   
## [41] "among" "another" "answer" "anyone"   
## [45] "anything" "anyway" "apparently" "appear"   
## [49] "appears" "appreciate" "appropriate" "area"   
## [53] "arent" "argument" "arguments" "around"   
## [57] "article" "articles" "ask" "asked"   
## [61] "asking" "ass" "assume" "attack"   
## [65] "attacks" "attempt" "attention" "august"   
## [69] "author" "automatically" "available" "avoid"   
## [73] "aware" "away" "back" "bad"   
## [77] "ban" "band" "banned" "based"   
## [81] "become" "behavior" "believe" "best"   
## [85] "better" "bias" "biased" "big"   
## [89] "bit" "black" "block" "blocked"   
## [93] "blocking" "book" "books" "box"   
## [97] "bring" "british" "call" "called"   
## [101] "calling" "came" "can" "cant"   
## [105] "care" "case" "cases" "category"   
## [109] "cause" "certain" "certainly" "chance"   
## [113] "change" "changed" "changes" "changing"   
## [117] "check" "cheers" "citation" "citations"   
## [121] "cite" "cited" "city" "claim"   
## [125] "claims" "clear" "clearly" "close"   
## [129] "come" "comes" "coming" "comment"   
## [133] "comments" "common" "community" "company"   
## [137] "complete" "completely" "concerns" "conflict"   
## [141] "consensus" "consider" "considered" "contact"   
## [145] "content" "contest" "context" "continue"   
## [149] "contribs" "contribute" "contributing" "contributions"   
## [153] "copy" "copyright" "correct" "country"   
## [157] "couple" "course" "create" "created"   
## [161] "creating" "criteria" "criticism" "current"   
## [165] "currently" "date" "day" "days"   
## [169] "deal" "dear" "death" "debate"   
## [173] "decide" "decided" "definition" "delete"   
## [177] "deleted" "deleting" "deletion" "described"   
## [181] "description" "despite" "details" "didnt"   
## [185] "difference" "different" "directly" "disagree"   
## [189] "discuss" "discussed" "discussion" "dispute"   
## [193] "disruptive" "doesnt" "done" "dont"   
## [197] "doubt" "due" "earlier" "early"   
## [201] "edit" "edited" "editing" "editor"   
## [205] "editors" "edits" "either" "else"   
## [209] "email" "encyclopedia" "end" "english"   
## [213] "enjoy" "enough" "entire" "entirely"   
## [217] "entry" "error" "especially" "etc"   
## [221] "even" "ever" "every" "everyone"   
## [225] "everything" "evidence" "exactly" "example"   
## [229] "except" "exist" "existing" "explain"   
## [233] "explaining" "explanation" "external" "fact"   
## [237] "facts" "fair" "faith" "false"   
## [241] "family" "far" "feel" "field"   
## [245] "file" "film" "finally" "find"   
## [249] "fine" "first" "five" "fix"   
## [253] "follow" "following" "form" "found"   
## [257] "four" "free" "friend" "friends"   
## [261] "fuck" "fucking" "full" "future"   
## [265] "game" "gave" "general" "generally"   
## [269] "get" "gets" "getting" "give"   
## [273] "given" "gives" "giving" "god"   
## [277] "goes" "going" "gone" "good"   
## [281] "google" "got" "government" "great"   
## [285] "group" "guess" "guidelines" "guy"   
## [289] "guys" "hand" "happen" "happened"   
## [293] "happy" "hard" "hate" "havent"   
## [297] "head" "heard" "hell" "hello"   
## [301] "help" "helpful" "hes" "hey"   
## [305] "high" "highly" "historical" "history"   
## [309] "hope" "hours" "however" "human"   
## [313] "idea" "ill" "image" "images"   
## [317] "important" "improve" "inappropriate" "include"   
## [321] "included" "including" "inclusion" "incorrect"   
## [325] "indeed" "indicate" "info" "information"   
## [329] "instead" "interest" "interested" "interesting"   
## [333] "internet" "involved" "irrelevant" "isnt"   
## [337] "issue" "issues" "ive" "january"   
## [341] "job" "john" "july" "june"   
## [345] "just" "keep" "kind" "know"   
## [349] "knowledge" "known" "knows" "lack"   
## [353] "language" "large" "last" "later"   
## [357] "law" "lead" "learn" "least"   
## [361] "leave" "left" "less" "let"   
## [365] "lets" "level" "life" "like"   
## [369] "likely" "line" "link" "linked"   
## [373] "links" "list" "listed" "little"   
## [377] "live" "living" "long" "longer"   
## [381] "look" "looked" "looking" "looks"   
## [385] "lost" "lot" "love" "made"   
## [389] "main" "major" "majority" "make"   
## [393] "makes" "making" "man" "manual"   
## [397] "many" "march" "material" "matter"   
## [401] "may" "maybe" "mean" "meaning"   
## [405] "means" "meant" "media" "members"   
## [409] "mention" "mentioned" "merely" "message"   
## [413] "messages" "might" "mind" "mine"   
## [417] "mistake" "months" "move" "moved"   
## [421] "much" "multiple" "music" "must"   
## [425] "name" "names" "national" "necessary"   
## [429] "need" "needed" "needs" "neither"   
## [433] "neutral" "never" "new" "news"   
## [437] "next" "nice" "none" "nonsense"   
## [441] "notability" "notable" "note" "nothing"   
## [445] "notice" "noticed" "now" "npov"   
## [449] "number" "obvious" "obviously" "official"   
## [453] "often" "okay" "old" "one"   
## [457] "ones" "online" "open" "opinion"   
## [461] "opinions" "order" "original" "others"   
## [465] "otherwise" "outside" "page" "pages"   
## [469] "paragraph" "part" "particular" "particularly"   
## [473] "party" "past" "people" "per"   
## [477] "perhaps" "period" "person" "personal"   
## [481] "personally" "picture" "piece" "pillars"   
## [485] "place" "placed" "play" "please"   
## [489] "point" "points" "policies" "policy"   
## [493] "political" "position" "possible" "possibly"   
## [497] "post" "posted" "posting" "pov"   
## [501] "power" "present" "pretty" "previous"   
## [505] "probably" "problem" "problems" "process"   
## [509] "produce" "project" "proof" "proper"   
## [513] "proposed" "prove" "provide" "provided"   
## [517] "public" "published" "purpose" "put"   
## [521] "putting" "quality" "question" "questions"   
## [525] "quite" "quote" "rather" "read"   
## [529] "readers" "reading" "real" "really"   
## [533] "reason" "reasons" "recent" "recently"   
## [537] "record" "redirect" "refer" "reference"   
## [541] "references" "regarding" "regards" "related"   
## [545] "release" "relevant" "reliable" "remember"   
## [549] "removal" "remove" "removed" "removing"   
## [553] "reply" "report" "request" "requesting"   
## [557] "research" "respect" "respond" "response"   
## [561] "rest" "result" "revert" "reverted"   
## [565] "reverting" "review" "right" "rights"   
## [569] "rule" "rules" "run" "said"   
## [573] "sandbox" "saw" "say" "saying"   
## [577] "says" "school" "science" "search"   
## [581] "second" "section" "sections" "see"   
## [585] "seem" "seems" "seen" "sense"   
## [589] "sentence" "separate" "series" "serious"   
## [593] "seriously" "set" "several" "shit"   
## [597] "short" "shouldnt" "show" "shows"   
## [601] "side" "sign" "significant" "similar"   
## [605] "simple" "simply" "since" "single"   
## [609] "site" "sites" "situation" "small"   
## [613] "someone" "something" "sometimes" "soon"   
## [617] "sorry" "sort" "source" "sourced"   
## [621] "sources" "speak" "specific" "specifically"   
## [625] "speedy" "standard" "start" "started"   
## [629] "state" "stated" "statement" "statements"   
## [633] "states" "status" "stay" "still"   
## [637] "stop" "story" "stuff" "stupid"   
## [641] "style" "subject" "subjects" "suggest"   
## [645] "summary" "support" "supposed" "sure"   
## [649] "system" "tag" "tagged" "tags"   
## [653] "take" "taken" "taking" "talk"   
## [657] "talking" "tell" "template" "term"   
## [661] "terms" "test" "text" "thank"   
## [665] "thanks" "thats" "theory" "therefore"   
## [669] "theres" "theyre" "thing" "things"   
## [673] "think" "thinking" "third" "though"   
## [677] "thought" "three" "thus" "tildes"   
## [681] "time" "times" "title" "today"   
## [685] "together" "told" "took" "top"   
## [689] "topic" "totally" "towards" "tried"   
## [693] "true" "truth" "try" "trying"   
## [697] "tutorial" "two" "type" "unblock"   
## [701] "understand" "understanding" "unfortunately" "united"   
## [705] "university" "unless" "uploaded" "upon"   
## [709] "use" "used" "useful" "user"   
## [713] "username" "users" "uses" "using"   
## [717] "usually" "utc" "valid" "vandalism"   
## [721] "vandalize" "various" "version" "video"   
## [725] "view" "views" "violation" "wait"   
## [729] "want" "wanted" "wants" "war"   
## [733] "warning" "wasnt" "way" "web"   
## [737] "website" "week" "welcome" "well"   
## [741] "went" "whatever" "whats" "whether"   
## [745] "white" "whole" "wiki" "wikipedia"   
## [749] "wikipedian" "wikipedias" "will" "wish"   
## [753] "within" "without" "wont" "word"   
## [757] "words" "work" "worked" "working"   
## [761] "works" "world" "worth" "wouldnt"   
## [765] "write" "writing" "written" "wrong"   
## [769] "wrote" "yeah" "year" "years"   
## [773] "yes" "yet" "youd" "youll"   
## [777] "youre" "youve"

dim(dataset)

## [1] 159571 778

dataset$toxic = NULL  
dataset$severe\_toxic = NULL  
dataset$obscene = NULL  
dataset$threat = NULL  
dataset$insult = NULL  
dataset$identity\_hate = NULL  
dim(dataset)

## [1] 159571 778

# attachh the output variables  
dataset$toxic = train$toxic  
dataset$severe\_toxic = train$severe\_toxic  
dataset$obscene = train$obscene  
dataset$threat = train$threat  
dataset$insult = train$insult  
dataset$identity\_hate = train$identity\_hate  
dim(dataset)

## [1] 159571 784

# create a multiclass variable  
attach(dataset)

## The following object is masked from package:base:  
##   
## version

dataset$target = ifelse(toxic==1,0,NA)  
dataset$target = ifelse(severe\_toxic==1,1,dataset$target)  
dataset$target = ifelse(obscene==1,2,dataset$target)  
dataset$target = ifelse(threat==1,3,dataset$target)  
dataset$target = ifelse(insult==1,4,dataset$target)  
dataset$target = ifelse(identity\_hate==1,5,dataset$target)  
dataset$target = ifelse(is.na(dataset$target),6,dataset$target)  
  
table(dataset$target)

##   
## 0 1 2 3 4 5 6   
## 5666 41 2233 163 6717 1405 143346

#remove na rows  
  
dataset = dataset[-which(dataset$target==6),]  
table(dataset$target)

##   
## 0 1 2 3 4 5   
## 5666 41 2233 163 6717 1405

dim(dataset)

## [1] 16225 785

# import the test datasets  
test = read.csv("test.csv")  
test\_labels = read.csv("test\_labels.csv")  
  
# clean the text  
  
corpus = VCorpus(VectorSource(test$comment\_text))  
corpus = tm\_map(corpus, content\_transformer(tolower))  
corpus = tm\_map(corpus, removeNumbers)  
corpus = tm\_map(corpus, removePunctuation)  
corpus = tm\_map(corpus, removeWords, stopwords())  
#corpus = tm\_map(corpus, stemDocument)  
corpus = tm\_map(corpus, stripWhitespace)  
  
# Creating the Bag of Words model  
dtm = DocumentTermMatrix(corpus)  
dtm = removeSparseTerms(dtm, 0.994)  
test\_dataset = as.data.frame(as.matrix(dtm))  
dim(test\_dataset)

## [1] 153164 685

# create a multiclass variable  
attach(test\_labels)

## The following objects are masked from dataset:  
##   
## identity\_hate, insult, obscene, severe\_toxic, threat, toxic

test\_dataset$target = ifelse(toxic==1,0,NA)  
test\_dataset$target = ifelse(severe\_toxic==1,1,test\_dataset$target)  
test\_dataset$target = ifelse(obscene==1,2,test\_dataset$target)  
test\_dataset$target = ifelse(threat==1,3,test\_dataset$target)  
test\_dataset$target = ifelse(insult==1,4,test\_dataset$target)  
test\_dataset$target = ifelse(identity\_hate==1,5,test\_dataset$target)  
test\_dataset$target = ifelse(is.na(test\_dataset$target),6,test\_dataset$target)  
table(test\_dataset$target)

##   
## 0 2 3 4 5 6   
## 1710 931 65 2825 712 146921

#remove na rows  
  
test\_dataset = test\_dataset[-which(test\_dataset$target==6),]  
table(test\_dataset$target)

##   
## 0 2 3 4 5   
## 1710 931 65 2825 712

dim(test\_dataset)

## [1] 6243 686

# match the variables of the traning\_set with those of the test\_dataset  
dim(test\_dataset)

## [1] 6243 686

dim(dataset)

## [1] 16225 785

common\_cols = intersect(colnames(dataset),colnames(test\_dataset))  
  
# create a dataset with 666 common variables with the test\_dataset and train dataset.  
  
  
dataset1= dataset[common\_cols]  
dim(dataset1)

## [1] 16225 666

test\_dataset1 = test\_dataset[common\_cols]  
dim(test\_dataset1)

## [1] 6243 666

# Encoding the target feature as factor  
dataset1$target = dataset$target  
dataset1$target = factor(dataset1$target)  
# Splitting the dataset into the Training set and Test set  
#library(caTools)  
library(caret)

## Loading required package: lattice

set.seed(123)  
split = createDataPartition(dataset1$target, p=0.7, list = FALSE)  
train\_set = dataset1[split,]  
test\_set = dataset1[-split,]  
dim(train\_set)

## [1] 11361 666

dim(test\_set)

## [1] 4864 666

# Applying PCA  
library(e1071)  
pca = preProcess(x = train\_set, method = 'pca', pcaComp = 5)  
train\_set = predict(pca, train\_set)  
test\_set = predict(pca, test\_set)  
test\_dataset2 = predict(pca, test\_dataset1)  
  
dim(train\_set)

## [1] 11361 6

dim(test\_set)

## [1] 4864 6

dim(test\_dataset2)

## [1] 6243 6

# Fitting Neural netweork to the Training set  
  
set.seed(300)  
ctrl = trainControl(method="cv",number=3)  
nn\_classifier = train(target~.,data=train\_set,method="nnet",trControl=ctrl,na.action=na.omit)

## iter 100 value 14235.319125  
## final value 14235.319125   
## stopped after 100 iterations

y\_pred <- predict(nn\_classifier, newdata=test\_set[-1])  
#head(y\_pred)  
# Making the Confusion Matrix  
cm = table(test\_set[, 1], y\_pred)  
confusionMatrix(cm)

## Confusion Matrix and Statistics  
##   
## y\_pred  
## 0 1 2 3 4 5  
## 0 836 0 0 0 863 0  
## 1 2 0 0 0 10 0  
## 2 267 0 0 0 402 0  
## 3 8 0 0 0 40 0  
## 4 632 0 0 0 1383 0  
## 5 99 0 0 0 322 0  
##   
## Overall Statistics  
##   
## Accuracy : 0.4562   
## 95% CI : (0.4421, 0.4703)  
## No Information Rate : 0.6209   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.1091   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: 0 Class: 1 Class: 2 Class: 3 Class: 4 Class: 5  
## Sensitivity 0.4534 NA NA NA 0.4579 NA  
## Specificity 0.7142 0.997533 0.8625 0.990132 0.6573 0.91345  
## Pos Pred Value 0.4921 NA NA NA 0.6864 NA  
## Neg Pred Value 0.6815 NA NA NA 0.4254 NA  
## Prevalence 0.3791 0.000000 0.0000 0.000000 0.6209 0.00000  
## Detection Rate 0.1719 0.000000 0.0000 0.000000 0.2843 0.00000  
## Detection Prevalence 0.3493 0.002467 0.1375 0.009868 0.4143 0.08655  
## Balanced Accuracy 0.5838 NA NA NA 0.5576 NA

# Fitting a naive baeys model to the Training set  
nb\_classifier = train(target~.,data=train\_set,method="nb",trControl=ctrl,na.action=na.omit)

with  
## observation 3635

# Fitting a random forest to the Training set  
#rf\_classifier = train(target~.,data=train\_set,method="rf",trControl=ctrl,na.action=na.omit)  
  
  
# Predicting the Test set results  
#which(colnames(test\_set)=="target")  
y\_pred <- predict(nb\_classifier, newdata=test\_set[-1])

with  
## observation 4813

#head(y\_pred)  
# Making the Confusion Matrix  
cm = table(test\_set[, 1], y\_pred)  
confusionMatrix(cm)

## Confusion Matrix and Statistics  
##   
## y\_pred  
## 0 1 2 3 4 5  
## 0 842 4 9 0 844 0  
## 1 5 0 0 0 7 0  
## 2 260 1 1 0 407 0  
## 3 14 0 0 0 34 0  
## 4 695 4 5 0 1311 0  
## 5 108 0 2 0 311 0  
##   
## Overall Statistics  
##   
## Accuracy : 0.4428   
## 95% CI : (0.4288, 0.4569)  
## No Information Rate : 0.5991   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.0913   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: 0 Class: 1 Class: 2 Class: 3 Class: 4  
## Sensitivity 0.4376 0.000000 0.0588235 NA 0.4499  
## Specificity 0.7085 0.997528 0.8621828 0.990132 0.6390  
## Pos Pred Value 0.4956 0.000000 0.0014948 NA 0.6506  
## Neg Pred Value 0.6581 0.998145 0.9961859 NA 0.4373  
## Prevalence 0.3956 0.001850 0.0034951 0.000000 0.5991  
## Detection Rate 0.1731 0.000000 0.0002056 0.000000 0.2695  
## Detection Prevalence 0.3493 0.002467 0.1375411 0.009868 0.4143  
## Balanced Accuracy 0.5731 0.498764 0.4605032 NA 0.5444  
## Class: 5  
## Sensitivity NA  
## Specificity 0.91345  
## Pos Pred Value NA  
## Neg Pred Value NA  
## Prevalence 0.00000  
## Detection Rate 0.00000  
## Detection Prevalence 0.08655  
## Balanced Accuracy NA

# predict on the test dataset  
y\_pred = predict(nb\_classifier, newdata=test\_dataset2,type="prob")

#y\_pred = predict(xgb\_classifier, newdata=data.matrix(test\_dataset2),reshape=TRUE)  
#y\_pred = as.matrix(y\_pred)  
#head(y\_pred)