Wikipedia-KNN

Imagine we try to propose categories for a Wikipedia article that lacks this information. Develop an algorithm and evaluate its performance on existing data

1- Load the CSV file /home/bigdata/9/enwiki-categories.csv containing article name and categories

```
#install.packages("SnowballC")
#install.packages("sqldf")
library(NLP)
library(tm) # Text mining: Corpus and Document Term Matrix
library(class) # KNN model
library(SnowballC) # Stemming words
library(NLP)

X = read.csv('categories.csv',colClasses=c("NULL","NULL",NA,NA,"NULL"), header = FALSE)
Y = read.csv('categories.csv',colClasses=c("NULL","NULL","NULL","NULL",NA), header = FALSE)
colnames(X) = c("title","content")
corpus = Corpus(VectorSource(X$content))
# Clean corpus
corpus <- tm_map(corpus, content_transformer(tolower))
corpus <- tm_map(corpus, removeNumbers)
corpus <- tm_map(corpus, removePunctuation)
corpus <- tm_map(corpus, stripWhitespace)
corpus <- tm_map(corpus, removeWords, stopwords("english"))
corpus <- tm_map(corpus, stemDocument, language = "english")

dtm <- DocumentTermMatrix(corpus)

# Transform dtm to matrix to data frame - df is easier to work with
mat.df <- as.data.frame(data.matrix(dtm), stringsAsfactors = FALSE)

# Column bind category (known classification)
mat.df <- cbind(mat.df, Y$V5)

# Change name of new column to "category"
colnames(mat.df)[ncol(mat.df)] <- "category"
# mat.df[15493]
# ncol(mat.df)
# colnames(mat.df)[1]
```

2- Split the articles (or a subset of them) into training, validation and test set.

```
set.seed(7)
ss <- sample(1:3,size=nrow(mat.df),replace=TRUE,prob=c(0.6,0.2,0.2))

# Isolate classifier
cl <- mat.df[, "category"]

# Create model data and remove "category"
modeldata <- mat.df[,!colnames(mat.df) %in% "category"]
```

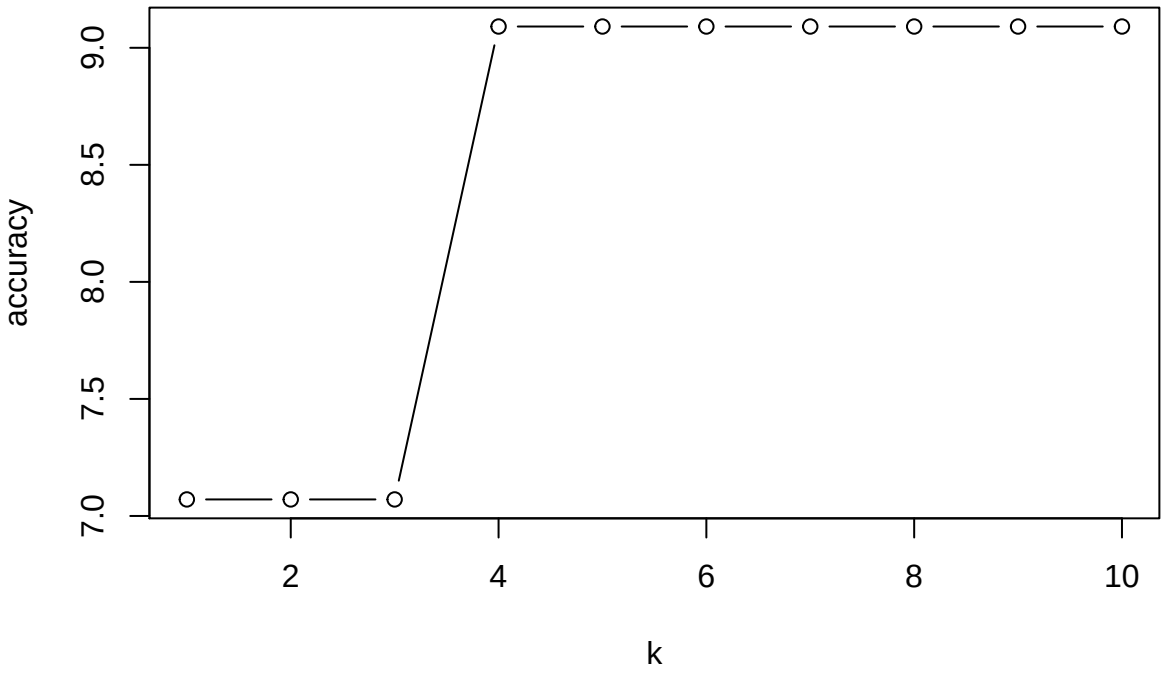
Train a k-nearest neighbor classifier with the training data and one other model of your choice. Features for an article could be the relative word frequency, i.e., two articles are similar if all individual word frequencies

are similar. You can also user other metrics of your choice. Check the accuracy of the predictions on the validation sets and tune your models.

```
accuracy <- rep(0, 10)
k <- 1:10
for(x in k){
# Create model: training set, test set, training set classifier
knn.pred <- knn(modeldata[ss==1, ], modeldata[ss==2, ], cl[ss==1], k  = x )

# Confusion matrix
conf.mat <- table("Predictions" = knn.pred, Actual = cl[ss==2])

# Accuracy
accuracy[x] <- (sum(diag(conf.mat))/length(cl[ss==2]) * 100)
}
plot(k, accuracy, type = 'b')
```




```
accuracy
```

```
##  [1] 7.070707 7.070707 7.070707 9.090909 9.090909 9.090909 9.090909
##  [8] 9.090909 9.090909 9.090909
```

Assess the expected accuracy of your tuned models on the test set

```
knn.pred <- knn(modeldata[ss==1 || ss == 2, ], modeldata[ss==3, ], cl[ss==1 || ss == 2], k  = 4 )
# Accuracy
# Confusion matrix
conf.mat <- table("Predictions" = knn.pred, Actual = cl[ss==3])
accuracy <- (sum(diag(conf.mat))/length(cl[ss==3]) * 100)
accuracy
```

```
## [1] 20
```

```
# Create data frame with test data and predicted category
df.pred <- cbind.data.frame(knn.pred, cl[ss==3])
```

```
write.table(df.pred, file="output.csv", sep=";")
```

Inspect a few (random) articles and compare the suggestions of your classifier with the available data.

Actual : [‘Lists of minor planets by number’] Predicted: [‘Lists of minor planets by number’]

Actual : [‘Federal departments of Switzerland’, ‘Federal Department of Defence, Civil Protection and Sports’, ‘Military of Switzerland’, ‘Sport in Switzerland’, ‘1848 establishments in Switzerland’]

Predicted:

[‘Hong Kong Federation of Students’, ‘Politics of Hong Kong’, "Students’ unions in Hong Kong", ‘Universities in Hong Kong’, "Groups of students’ unions", ‘1958 establishments in Hong Kong’, ‘Student organizations established in 1958’]

Actual : “[‘1923 births’, ‘1985 deaths’, ‘New Zealand cricketers’, ‘Auckland cricketers’, ‘Sportspeople from Auckland’ Predicted: [‘Williams pinball machines’, ‘1985 pinball machines’]

Actual : [] Predicted: [‘1795 births’, ‘1882 deaths’, ‘People from Loire (department)’, ‘French cardinals’]

Test text