KENDRA OSBURN | 2-26-19 | IST707 | HW7 |

HANDWRITING RECOGNITION

Hotel Monteleone
Buttermilk
Ice
Jack Daniels
Ling Zang
Car Max
214 Rue Royale • New Orleans, LA
Telephone: (504) 523-3341 Fax: (504) 681-4413
Toll Free: 1-800-535-9595

Introduction Attempt One

Once upon a time, when you were very young -- not even a seedling in my tummy yet! -- your father and I would use this long, thin tool, called a pen, yes PEN. Can you spell that for mom? P - E - N. Exactly. We would use this *pen* to write -- yes, this is before writing on the computer -- we would write notes to one another. No, no not on your tablet, the pen was used with paper. Oh sweet girl, no, not like the paper from the toilet, no. Well, I guess kind of like the paper from the toilet? Just imagine a little sturdier. And about the same size as dad's big tablet. So yes, we would write one another little letters by hand, using pen and paper. It was really quite romantic. Unfortunately many of them got lost during the big fire... what was that? Oh, yes. The big fire that burnt mom and dad's first home. And grandma and grandpa's. Great memory! But yes, it was very sad. We used to live -- Honey can you grab that old map from your office? Yes the framed one. -- much farther from the ocean because the ocean used to be all the way out here. See this? That used to be the coast! Before the fires and the flooding, this was one of the most beautiful places in the whole world. I agree that it’s still very beautiful, but you, little one, you are what makes it beautiful -- right Honey?!

Thankfully, your mom has always been a digital hoarder, so she insisted on digitizing those letters forever ago. Let’s see if I can pull some up right now. I might not be able to read my writing! Honey do you remember cursive?! One of my most favorite days in elementary school -- yes, elementary school is similar to your primers, yes, but imagine going to a big building, a big SQUARE building with all the other kids. Not virtual school, like, I could reach out and touch my seat partner. Just like I’m touching you right now! Right Honey? Oh man, I miss the smell of those sticky classrooms -- and library books!! What’s a library book? You know what, let’s bookmark that for now. That just means to save our spot so we can return to it later. Yes, they are all very related. Dad will forever be sad he didn’t get to show you a real book. No, no. Not a book on the tablet. Like, a book book. With paper. Yes, like paper from the toilet.

Introduction Attempt Two

THE YEAR IS 2074, BOOKS AND PAPERS ARE SACRED AND HOUSED ONLY IN MUSEUMS.

In the early days of humanity, one of the ways society would communicate, both individually and as a group, was through writing. Before the printing press, everything was handwritten. NOTE: In this context handwritten doesn’t mean typed-by-hand. It means literally handwritten. Again, **not typed**. Written. Even after the invention of the printing press -- and even for a while after the invention of the typewriter, computer and holotab -- handwriting was still something that humans did. As technology increased, handwriting became more and more niche, used almost exclusively for art or gonzo historical analysis.

Analysis & Models

THE DATA (FROM KAGGLE)

The training data set, (train.csv), has 785 columns. The first column, called "label", is the digit that was drawn by the user. The rest of the columns contain the pixel-values of the associated image.

*Each pixel column in the training set has a name like pixelx, where x is an integer between 0 and 783, inclusive. To locate this pixel on the image, suppose that we have decomposed x as x = i * 28 + j, where i and j are integers between 0 and 27, inclusive. Then pixelx is located on row i and column j of a 28 x 28 matrix, (indexing by zero).*

These libraries were used: e1071, dyplr, naivebayes. The data was downloaded from kaggle.

```
## ====================================
## STEP 1: IMPORT LIBRARIES
## ====================================
library(e1071)
library(naivebayes)
library(dplyr)

## ====================================
## STEP 2: IMPORT DATA
## ====================================
trainNB <- read.csv("/Users/kosburn
1/syracuse/IST707/WK7/handwriting_train.csv", header = TRUE)
## testNB <-
read.csv("/Users/kosburn/syracuse/IST707/WK7/handwriting_test.csv", header
= TRUE)

## 2a. Working backup
originalTrainNB <- trainNB
## originalTestNB <-testNB

## ====================================
## STEP 3: PREPARE THE DATA
## ====================================

## 3a. REMOVE LABEL
trainNB_noLabel <- trainNB[,-1]
```

```
## 3b. TURN EVERYTHING != 0 into 1
trainNB_noLabel <- trainNB_noLabel %>%
  mutate_if(is.integer, as.numeric)
trainNB_noLabel[trainNB_noLabel != 0 ] <- 1
max(trainNB_noLabel)

## 3c. ADD LABEL BACK
trainNB_boollabel <- cbind("label"=trainNB$label, trainNB_noLabel)

## 3d. ALL TO FACTOR
trainNB_preSplit <- trainNB_boollabel %>%
  mutate_all(as.factor)

## 3e. REMOVE FACTOR WITH LEVEL 1
## TODO: UNDERSTAND THIS LINE
trainNB_preSplit <- (trainNB_preSplit[, sapply(trainNB_preSplit, nlevels) >
1])
workingTrain <- trainNB_preSplit


## ====================================
## STEP 4: SPLIT THE DATA
## ====================================

## 4a. SPLIT THE DATA EVENLY (like ham and mad) INTO 1-9
## TODO: REVISIT SUBSET
## subset(dataframe, colimsplittingon == 0)
Zero <- subset(workingTrain, label == 0)
One <- subset(workingTrain, label == 1)
Two <- subset(workingTrain, label == 2)
Three <- subset(workingTrain, label == 3)
Four <- subset(workingTrain, label == 4)
Five <- subset(workingTrain, label == 5)
Six <- subset(workingTrain, label == 6)
Seven <- subset(workingTrain, label == 7)
Eight <- subset(workingTrain, label == 8)
Nine <- subset(workingTrain, label == 9)

## GIVING UP ON LOOPS FOR NOW
## GET SAMPLE
get_sample <- function(df, n) {
  set.seed(42)
```

```r
  sample(nrow(df), n)
}

## GET TRAINSET
get_train_set <- function(df, columns) {
  set.seed(42)
  df[columns,]
}

## GET TEST SET
#Creating a function to create a testing df
get_test_set <- function(df, columns, n) {
  df <- df[-columns,]
  set.seed(42)
  df[sample(nrow(df), n),]
}


sample0 <- get_sample(Zero, 300)
train0 <- get_train_set(Zero, sample0)
test0 <- get_test_set(Zero, sample0, 100)

sample1 <- get_sample(One, 300)
train1 <- get_train_set(One, sample1)
test1 <- get_test_set(One, sample1, 100)

sample2 <- get_sample(Two, 300)
train2 <- get_train_set(Two, sample2)
test2 <- get_test_set(Two, sample2, 100)

sample3 <- get_sample(Three, 300)
train3 <- get_train_set(Three, sample3)
test3 <- get_test_set(Three, sample3, 100)

sample4 <- get_sample(Four, 300)
train4 <- get_train_set(Four, sample4)
test4 <- get_test_set(Four, sample4, 100)

sample5 <- get_sample(Five, 300)
train5 <- get_train_set(Five, sample5)
test5 <- get_test_set(Five, sample5, 100)
```

```
sample6 <- get_sample(Six, 300)
train6 <- get_train_set(Six, sample6)
test6 <- get_test_set(Six, sample6, 100)

sample7 <- get_sample(Seven, 300)
train7 <- get_train_set(Seven, sample7)
test7 <- get_test_set(Seven, sample7, 100)

sample8 <- get_sample(Eight, 300)
train8 <- get_train_set(Eight, sample8)
test8 <- get_test_set(Eight, sample8, 100)

sample9 <- get_sample(Nine, 300)
train9 <- get_train_set(Nine, sample9)
test9 <- get_test_set(Nine, sample9, 100)
## DO THAT A BUNCH OF TIME
## COMBINE TRAIN and TEST by ROWS (use rbind)

big_train_set <- rbind(train0, train1, train2, train3, train4, train5,
train6, train7, train8, train9)
big_test_set <- rbind(test0, test1, test2, test3, test4, test5, test6,
test7, test8, test9)

big_test_set_label <- big_test_set$label
big_test_set_nolabel <- big_test_set[,-1]

NB_e1071<-naiveBayes(label~., data=big_train_set, na.action = na.pass)
NB_e1071_Pred <- predict(NB_e1071, big_test_set_nolabel)
NB_e1071
table(NB_e1071_Pred,big_test_set_label)

##Visualize
#plot(NB_e1071, ylab = "Density", main = "Naive Bayes Plot")




## ************************************************
## MODELING: KNN TO PREDICT PERCENT PROFIT
## ************************************************
## 47% accuracy | DF USED: 'everything_USA'
```

```r
## ================================================
## STEP 1: LOAD THE DATA & LIBRARIES
## ================================================
library(ggplot2)
library(class)
library(dplyr)

model <- ('KNN')
# #Setting working directory
# # setwd("C:\\Users\\ho511\\Desktop\\IST 707\\Team Project\\CSVs")
# setwd("/Users/kosburn 1/syracuse/IST707/project/FINAL")
#
# #Reading in the Movies Spreadsheet
# kNN_train_percProf <- read.csv("train_perc_profit.csv", header = TRUE)
# kNN_test_percProf <- read.csv("test_perc_profit.csv", header= TRUE)

#Reading in the Movies Spreadsheet
# kNN_train_percProf <- read.csv("everything_train_perc_profit.csv", header
= TRUE)
# kNN_test_percProf <- read.csv("everything_test_perc_profit.csv", header=
TRUE)

df_train <- big_train_set
df_test <- big_test_set

# ## REMOVE ANY POSSIBLE Xs in df_train
# for(name in colnames(df_train)){
#   if(grepl('X', name)){
#     df_train <- df_train[,!colnames(df_train) %in% name]
#     print(name)
#   }
# }
# ## REMOVE ANY POSSIBLE Xs in df_test
# for(name in colnames(df_test)){
#   if(grepl('X', name)){
#     df_test <- df_test[,!colnames(df_test) %in% name]
#   }
# }


## ================================================
## STEP 2: PREP DATA FOR SPECIFIC MODEL
## ==================================================
#***************************************************
# kNN requires Numeric Data Only
# Remove the label from the training data
# Save Label in a train_label vector
#***************************************************

df_train_label <- df_train[,colnames(df_train) %in% ('label')]
df_test_label <- df_test[,colnames(df_test) %in% ('label')]

# #Removing all columns that are factors
# factorColumns <- df_train %>% Filter(f = is.factor) %>% names
# df_train <- df_train[,!colnames(df_train) %in% factorColumns]
# df_test <- df_test[,!colnames(df_test) %in% factorColumns]
#
# #Removing gross and profit
# df_train <- df_train[,!colnames(df_train) %in% c('gross', 'profit')]
# df_test <- df_test[,!colnames(df_test) %in% c('gross', 'profit')]
#
# #Changing everything to numeric from int
# df_train <- df_train %>% mutate_if(is.integer, as.numeric)
# df_test <- df_test %>% mutate_if(is.integer, as.numeric)

## ==================================================
## STEP 3: RUN THE MODEL
## ==================================================
# Setting k as the sqrt of the number of rows of the dataset
modelTitle <- (1)
modelname <- paste(model,"_", modelTitle, sep="")

(k <- round(sqrt(868)))
kNN <- class::knn(train = df_train, test = df_test, cl = df_train_label, k
= k, prob = TRUE)
(table <- table(kNN, df_test_label))

(miscalculation_rate <- 1 - sum(diag(table))/sum(table))
(accuracy_rate <- sum(diag(table))/sum(table))
exportDf <- data.frame(modelname, accuracy_rate, table ,accuracy)
exportDf
filenameCSV <- paste(modelname, ".csv", sep="")
write.csv(exportDf, file = filenameCSV)
```

```r
## ==================================================
## STEP 4: MAKE THE VIZ
## ==================================================
## nf stands for Number of Factors we are trying to predit
## nt stands for Number of Test objects within each factor
## TODO change percProf to varToTest

ggfilename <- paste(model, "_GG.png", sep="")
varToTest <- levels(df_train_label)
nf <- length(varToTest)
nt <- (nrow(df_test)/nf)
model <- replicate(nf, "vis")
accuracy <- c(diag(table)/nt*100)
model_accuracy <- data.frame(varToTest, model, accuracy)
(model_plot <- ggplot(model_accuracy, aes(x = varToTest, y = accuracy, fill
= rainbow(nf))) + geom_bar(stat = "identity") + theme(legend.position
="none") + scale_x_discrete(limit = varToTest))
(model_plot <- model_plot + ylab("Accuracy Percentage") + xlab("Percent
Profit") + ggtitle("Accuracy Percentage for KNN Model for Percent Profit")
+ ylim(0, 100))+ggsave(ggfilename, plot = last_plot(), device = NULL)
```