Data Exploration

Andy Shen

10/27/2020

Traditional Techniques

```
rm(list = ls())
library(MASS) #lda, qda
library(class) #knn
library(tidyverse)
library(caret)
```

```
setwd("/Users/andyshen/Desktop/Git/Stats-101C-F20/Midterm Project")
train <- read.csv("training.csv", stringsAsFactors = TRUE)
test <- read.csv("test.csv", stringsAsFactors = TRUE)
train$class <- factor(train$class)
levels(train$class) <- c("NG", "OG", "TSG")
test$class <- factor(test$class)
levels(test$class) <- c("NG", "OG", "TSG")
```

```
set.seed(5732)
samp <- sample(1:nrow(train), floor(0.8 * nrow(train)), replace = FALSE)
train1 <- train[samp, ]
test_train <- train[-samp, ]
```

FamilyMemberCount, RVIS_percentile, N_Missense, intolerant_pNull, Gene_age, pLOF_Zscore

VEST_score

LDA

```
lda.mod <- lda(
  class ~ FamilyMemberCount + RVIS_percentile + N_Missense +
    intolerant_pNull + Gene_age + pLOF_Zscore, data = train1
)
preds <- predict(lda.mod, test_train, type = "response")$posterior
preds <- apply(preds, 1, which.max) - 1
tbl <- table(preds, test_train$class)
ter <- sum(diag(tbl)) / sum(tbl)
tbl
```

```
##
## preds  NG  OG TSG
##     0 559  21  22
##     1   2   5   0
##     2  10   4  13
```

Test error rate is 0.093.

QDA

```
qda.mod <- qda(
  class ~ FamilyMemberCount + RVIS_percentile + N_Missense +
    intolerant_pNull + Gene_age + pLOF_Zscore, data = train1
)
preds <- predict(qda.mod, test_train, type = "response")$posterior
preds <- apply(preds, 1, which.max) - 1
tbl <- table(preds, test_train$class)
ter <- sum(diag(tbl)) / sum(tbl)
tbl
```

```
##
## preds  NG  OG TSG
##     0 503  12  10
##     1  35  14  10
##     2  33   4  15
```

Test error rate is 0.164

KNN

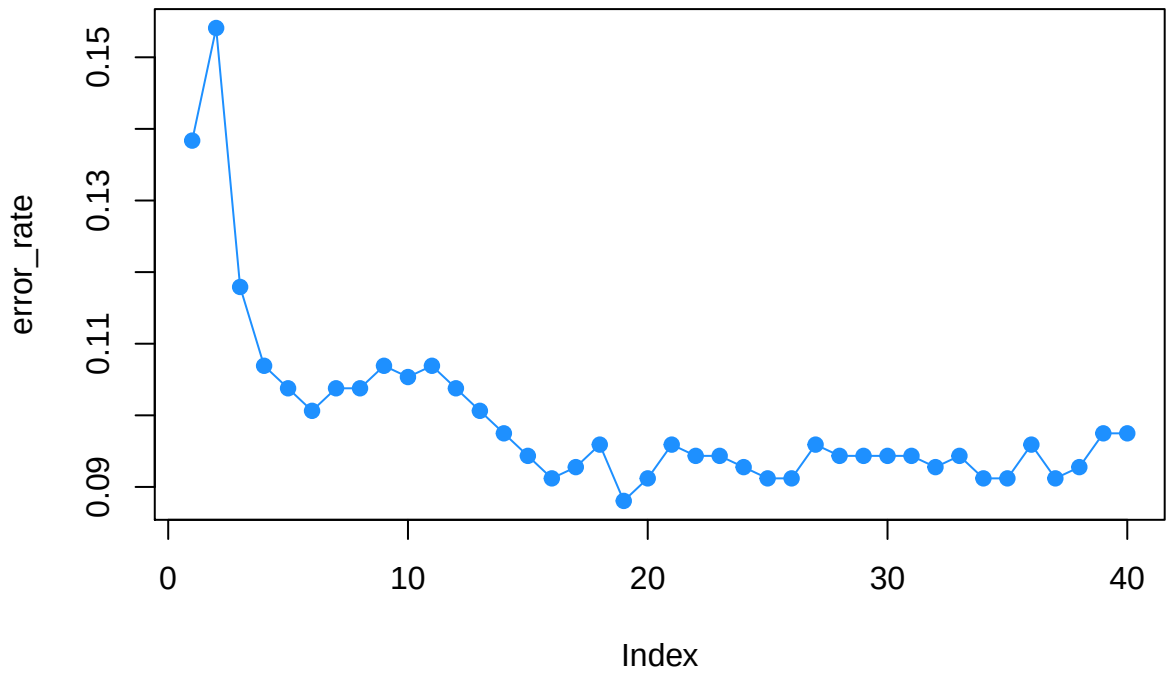
```
train1k <- train1 %>% dplyr::select(-class)
test_traink <- test_train %>% dplyr::select(-class)

for(col in 1:ncol(train1k)) {
  train1k[, col] <- train1k[, col] / max(train1k[, col])
  test_traink[, col] <- test_traink[, col] / max(test_traink[, col])
} #standardizing. ncol(test) == ncol(train)

trainx <- train1k %>% dplyr::select(
  FamilyMemberCount, RVIS_percentile, N_Missense, intolerant_pNull, Gene_age, pLOF_Zscore
)
trainy <- train1$class

testx <- test_traink %>% dplyr::select(
  FamilyMemberCount, RVIS_percentile, N_Missense, intolerant_pNull, Gene_age, pLOF_Zscore
)
testy <- test_train$class

rows <- 40
knn_mat <- matrix(NA, nrow = rows, ncol = length(testy))
error_rate <- rep(NA, rows)
for(i in 1:rows) {
  knn_mat[i,] <- knn(trainx, testx, trainy, k = i)
  tbl <- table("actual" = testy, "predicted" =  knn_mat[i,])
  error_rate[i] <- 1 - (sum(diag(tbl)) / sum(tbl))
}
plot(error_rate, type = "l", col = "dodgerblue")
points(error_rate, pch = 19, col = "dodgerblue")
```




```
best <- which.min(error_rate)
best_tbl <- table("actual" = testy, "predicted" =  knn_mat[best,])
```

```
best_tbl
```

```
##       predicted
## actual   1   2   3
##    NG  566   3   2
##    OG   23   6   1
##    TSG  24   3   8
```

```
ter <- sum(diag(best_tbl)) / sum(best_tbl)
```

Test error rate is 0.088. This is for K = 19.

Using caret

```
tc <- trainControl(
  method = "cv", number = 5, classProbs = TRUE, savePredictions = TRUE
) # setting up training technique
```

LDA

```
LDAfit <- caret::train(
  class ~ FamilyMemberCount + RVIS_percentile + N_Missense +
    intolerant_pNull + Gene_age + pLOF_Zscore,
  data = train1, method = "lda",
  preProc = c("center", "scale"),
  trControl = tc
)
```

QDA

```
QDAfit <- caret::train(
  class ~ FamilyMemberCount + RVIS_percentile + N_Missense +
    intolerant_pNull + Gene_age + pLOF_Zscore,
  data = train1, method = "qda",
  preProc = c("center", "scale"),
  trControl = tc
)
```

Logistic Regression

```
LRfit <- caret::train(
  class ~ FamilyMemberCount + RVIS_percentile + N_Missense +
    intolerant_pNull + Gene_age + pLOF_Zscore,
  data = train1, method = "glm", family = "binomial",
  preProc = c("center", "scale"),
  trControl = tc
)
```

```
## Warning: model fit failed for Fold1: parameter=none Error in method$fit(x = x, y = y, wts = wts, para
##   glm models can only use 2-class outcomes

## Warning: model fit failed for Fold2: parameter=none Error in method$fit(x = x, y = y, wts = wts, para
##   glm models can only use 2-class outcomes

## Warning: model fit failed for Fold3: parameter=none Error in method$fit(x = x, y = y, wts = wts, para
##   glm models can only use 2-class outcomes

## Warning: model fit failed for Fold4: parameter=none Error in method$fit(x = x, y = y, wts = wts, para
##   glm models can only use 2-class outcomes

## Warning: model fit failed for Fold5: parameter=none Error in method$fit(x = x, y = y, wts = wts, para
##   glm models can only use 2-class outcomes

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
## There were missing values in resampled performance measures.

## Something is wrong; all the Accuracy metric values are missing:
```

```
##     Accuracy       Kappa
##  Min.   : NA   Min.   : NA
##  1st Qu.: NA   1st Qu.: NA
##  Median : NA   Median : NA
##  Mean   :NaN   Mean   :NaN
##  3rd Qu.: NA   3rd Qu.: NA
##  Max.   : NA   Max.   : NA
##  NA's   :1     NA's   :1
```

```
## Error: Stopping
```

KNN

```
KNNfit <- caret::train(
  class ~ FamilyMemberCount + RVIS_percentile + N_Missense +
    intolerant_pNull + Gene_age + pLOF_Zscore, data = train1,
  method = "knn", preProc = c("center", "scale"), trControl = tc
)
ggplot(KNNfit) + theme_bw()
```