Data Exploration

Andy Shen

10/27/2020

```
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)
```

```
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   0   1   2
##     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   0   1   2
##     0 503  12  10
##     1  35  14  10
##     2  33   4  15
```

Test error rate is 0.164

KNN

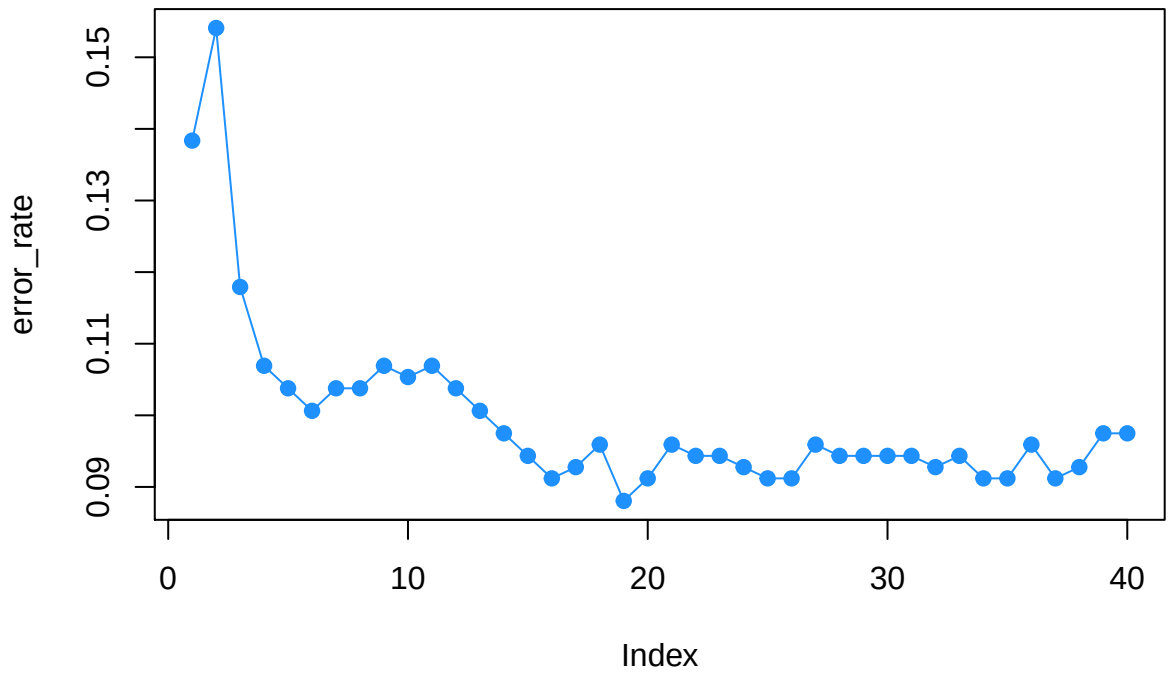
```
train1k <- train1 %>% dplyr::select(-class)
test_traink <- test_train %>% dplyr::select(-class)

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
##      0 566   3   2
##      1  23   6   1
##      2  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)
knn_c <- caret::train(
  class ~ FamilyMemberCount + RVIS_percentile + N_Missense +
    intolerant_pNull + Gene_age + pLOF_Zscore, data = train1,
  method = "knn"
)
knn_c
```

```
## k-Nearest Neighbors
##
## 2541 samples
##    6 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 2541, 2541, 2541, 2541, 2541, 2541, ...
## Resampling results across tuning parameters:
##
##   k  RMSE       Rsquared   MAE
##   5  0.4886163  0.1119551  0.2057785
##   7  0.4751973  0.1256874  0.2052157
##   9  0.4667617  0.1376250  0.2050066
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 9.
```