HW 11 (KNN)

Halid Kopanski

7/4/2021

Contents

Data Read in and cleaning

```
# pulling in and cleaning the data.
titanicData <-read_csv("titanic.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   pclass = col_double(),
##   survived = col_double(),
##   name = col_character(),
##   sex = col_character(),
##   age = col_double(),
##   sibsp = col_double(),
##   parch = col_double(),
##   ticket = col_character(),
##   fare = col_double(),
##   cabin = col_character(),
##   embarked = col_character(),
##   boat = col_character(),
##   body = col_double(),
##   home.dest = col_character()
## )
```

```
titanicData <-filter(titanicData,!is.na(survived)& !is.na(fare)& !is.na(age))
titanicData$survived <-as.factor(titanicData$survived)

# Creating training and test datasets
set.seed(1)
training <-sample(1:nrow(titanicData), size =nrow(titanicData)*0.8)
testing <- dplyr::setdiff(1:nrow(titanicData), training)
titanicDataTrain <- titanicData[training, ]
titanicDataTest <- titanicData[testing, ]
```

Short EDA

```
print(titanicDataTrain)
```

```
## # A tibble: 836 x 14
##    pclass survived name      sex     age sibsp parch ticket  fare cabin embarked
##     <dbl> <fct>    <chr>     <chr> <dbl> <dbl> <dbl> <chr>  <dbl> <chr> <chr>
##  1      3 0        Vande Wa~ male     28     0     0 345770  9.5  <NA>  S
##  2      3 0        Cribb, M~ male     44     0     1 371362 16.1  <NA>  S
##  3      1 1        Harder, ~ fema~    25     1     0 11765  55.4  E50   C
##  4      3 1        Persson,~ male     25     1     0 347083  7.78 <NA>  S
##  5      2 0        Norman, ~ male     28     0     0 218629 13.5  <NA>  S
##  6      2 1        Beane, M~ fema~    19     1     0 2908   26    <NA>  S
##  7      1 0        White, M~ male     54     0     1 35281  77.3  D26   S
##  8      3 0        Attalah,~ male     30     0     0 2694    7.22 <NA>  C
##  9      2 0        Chapman,~ fema~    29     1     0 SC/AH~ 26    <NA>  S
## 10      1 0        Brady, M~ male     41     0     0 113054 30.5  A21   S
## # ... with 826 more rows, and 3 more variables: boat <chr>, body <dbl>,
## #   home.dest <chr>
```

```
summary(titanicDataTrain)
```

```
##      pclass      survived     name               sex
##  Min.   :1.000   0:490    Length:836         Length:836
##  1st Qu.:1.000   1:346    Class :character   Class :character
##  Median :2.000            Mode  :character   Mode  :character
##  Mean   :2.233
##  3rd Qu.:3.000
##  Max.   :3.000
##
##       age              sibsp            parch           ticket
##  Min.   : 0.1667   Min.   :0.0000   Min.   :0.0000   Length:836
##  1st Qu.:21.0000   1st Qu.:0.0000   1st Qu.:0.0000   Class :character
##  Median :28.0000   Median :0.0000   Median :0.0000   Mode  :character
##  Mean   :29.3458   Mean   :0.5144   Mean   :0.4246
##  3rd Qu.:37.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :80.0000   Max.   :8.0000   Max.   :6.0000
##
##       fare            cabin             embarked            boat
##  Min.   :  0.00   Length:836         Length:836         Length:836
```

```
##  1st Qu.:  8.05   Class :character   Class :character   Class :character
##  Median : 15.02   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 35.24
##  3rd Qu.: 32.50
##  Max.   :512.33
##
##       body        home.dest
##  Min.   :  1.0   Length:836
##  1st Qu.: 79.0   Class :character
##  Median :149.0   Mode  :character
##  Mean   :159.4
##  3rd Qu.:249.0
##  Max.   :328.0
##  NA's   :747
```

KNN Fits

```
# Setting up the train control, in this case we will run a k fold of 10, 3 times
trctrl <- trainControl(method = "repeatedcv",
                       number = 10,
                       repeats = 3)

set.seed(2020)

# Using the argument tuneGrid, we will train models using k values of 2 to 30.
# The data will be standardized using the preprocess argument.
# For this case, we will only use the predictors age and fare.

knn_fit1 <- train(survived ~ .,
                  data = select(titanicDataTrain, survived, age, fare),
                  method = "knn",
                  trControl = trctrl,
                  preProcess = c("center", "scale"),
                  tuneGrid = data.frame(k = 2:30))
```

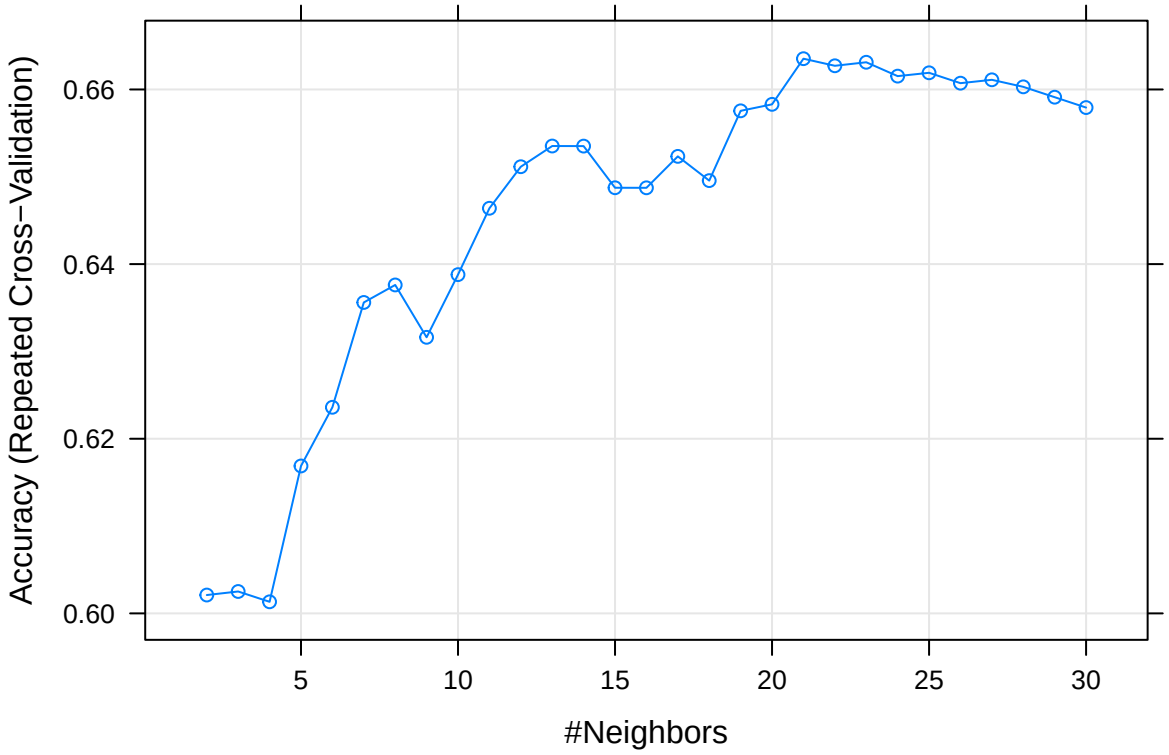
Additional Fits

```
# Some additional fits using a higher number of predictors. These will
# just be used to compare against the original fit.

knn_fit2 <- train(survived ~ .,
                  data = select(titanicDataTrain, survived, age, fare, sex),
                  method = "knn",
                  trControl = trctrl,
                  preProcess = c("center", "scale"),
                  tuneGrid = data.frame(k = 2:30))

knn_fit3 <- train(survived ~ .,
                   data = select(titanicDataTrain, survived,
                                 age, fare, sex, pclass),
```

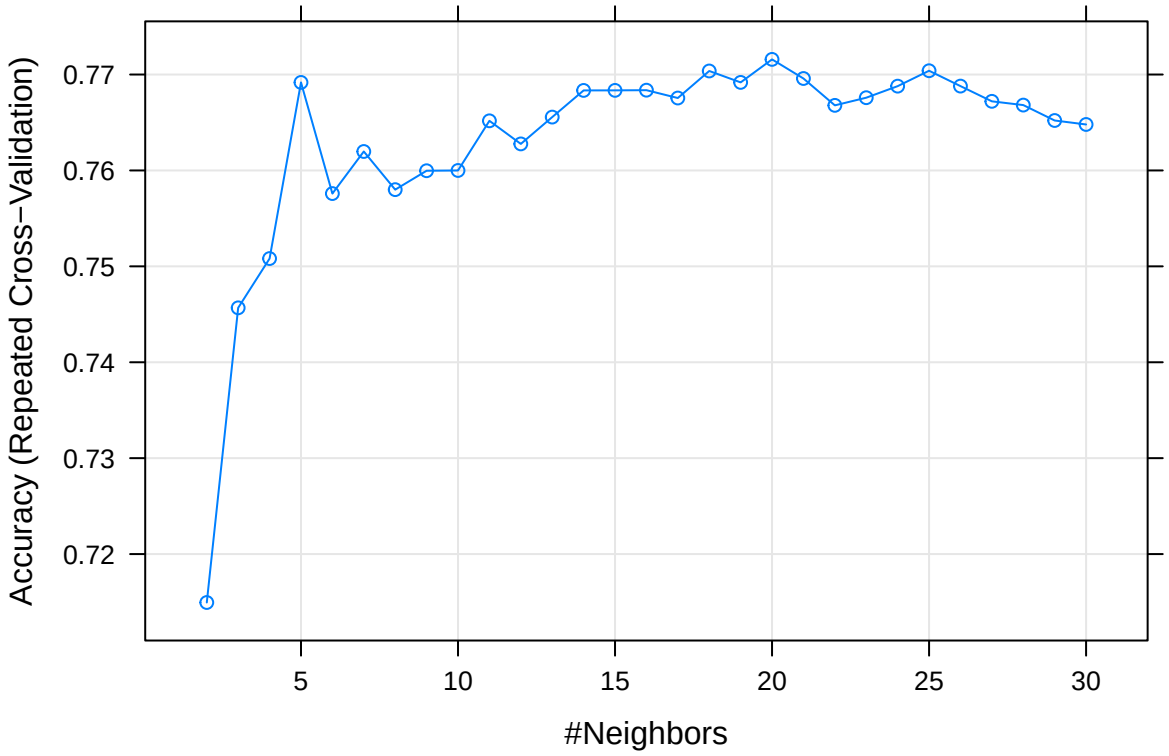
```
                    method = "knn",
                    trControl = trctrl,
                    preProcess = c("center", "scale"),
                    tuneGrid = data.frame(k = 2:30))
```

Plots

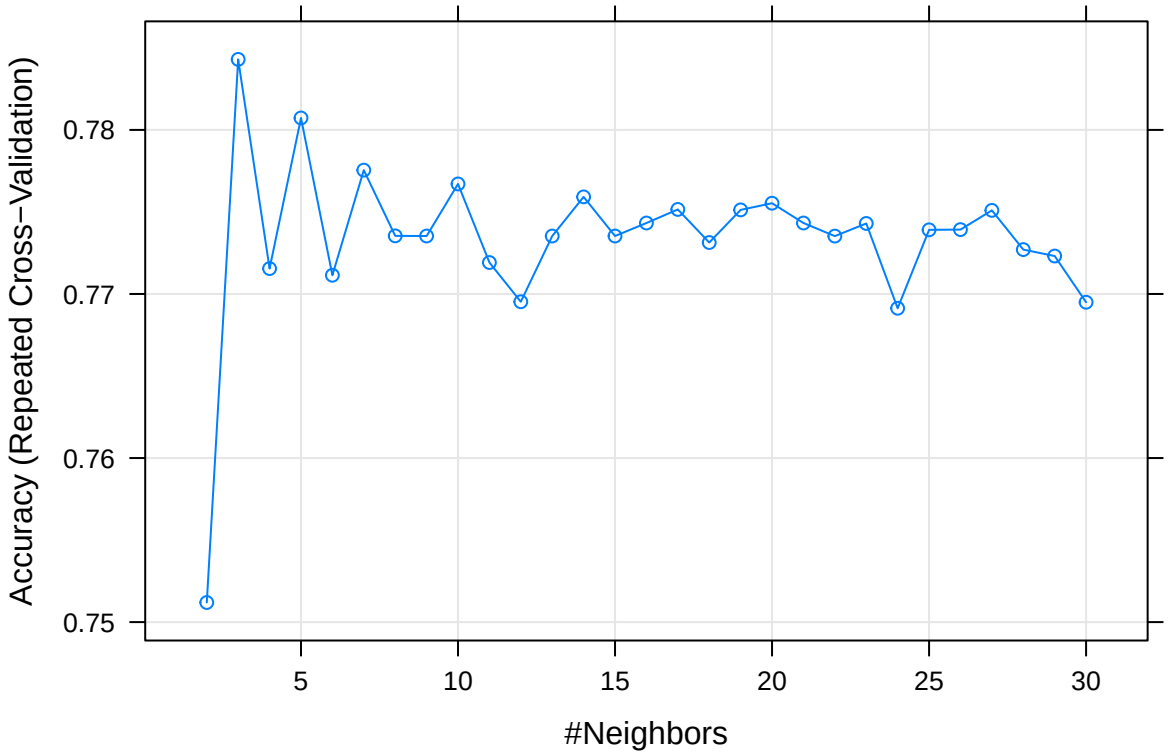
```
# Plotting the three fits from the previous step, we can see that higher
# k values are favored to a certain point.
plot(knn_fit1)
```




```
plot(knn_fit2)
```




```
plot(knn_fit3)
```



In the above plots, we can see that the best KNN model using age and fare used a k value of 21. Meaning, an individual classification is determined by the classification of the nearest 21 neighbors. The other two models needed k values of 20 and 3.

```
set.seed(2020)
# Run the best knn fit on the test data
knn_pred <- predict(knn_fit1, newdata = titanicDataTest)

# Create a compare set where none one survived
comparator <- sum(rep(0, nrow(titanicDataTest)) !=
                    titanicDataTest$survived) / nrow(titanicDataTest)

misclass <- sum(knn_pred != titanicDataTest$survived) / nrow(titanicDataTest)

# As we can see, the knn model using only age and fare did better
# than just assuming none survived.

sprintf("This is the misclassification rate for knn_fit1: %0.3f", misclass)
```

```
## [1] "This is the misclassification rate for knn_fit1: 0.292"
```

```
sprintf("This is the comparison assuming none survived: %0.3f", comparator)
```

```
## [1] "This is the comparison assuming none survived: 0.388"
```

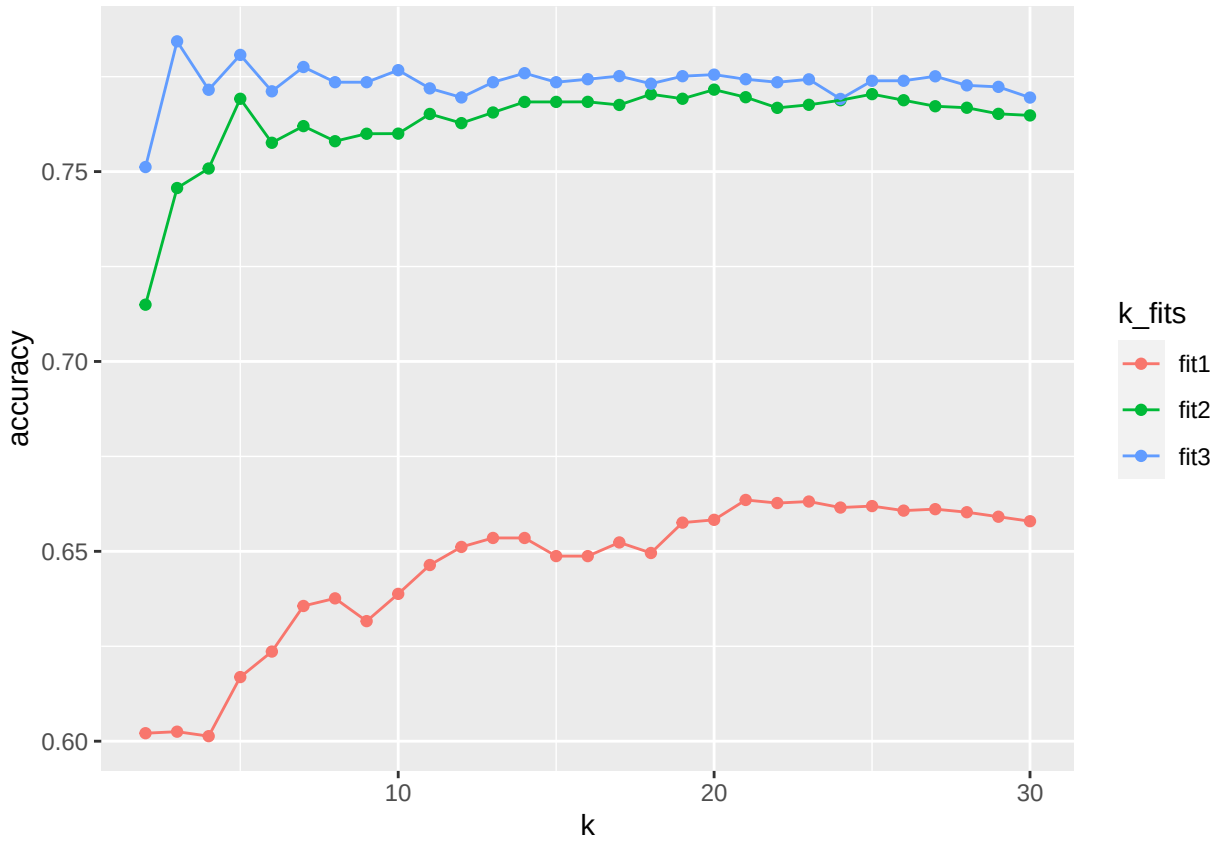
```r
# Comparing to the other two models, we can see that misclassifications can be
# reduced by added more information. In this case, sex is a better predictor
# than class. Class actually increased misclassifications.

misclass2 <- sum(predict(knn_fit2, newdata = titanicDataTest) !=
                   titanicDataTest$survived) / nrow(titanicDataTest)

misclass3 <- sum(predict(knn_fit3, newdata = titanicDataTest) !=
                   titanicDataTest$survived) / nrow(titanicDataTest)

sprintf("Adding more predictors increased accuracy: %0.3f",
        c(misclass2, misclass3))
```

```
## [1] "Adding more predictors increased accuracy: 0.201"
## [2] "Adding more predictors increased accuracy: 0.258"
```

Comparing Models

Here we can see how much of a boost in accuracy adding predictors can give to a KNN model. In the third model even though the test accuracy drops, the value of k required by the model is smaller. The best model k values are 21, 20, and 3 for models using (age, fare), (age, fare, sex), and (age, fare, sex, pclass) respectively.

```r
knn_fits <- as_tibble(data.frame(k = 2:30, fit1 = knn_fit1$results[[2]],
                                 fit2 = knn_fit2$results[[2]],
                                 fit3 = knn_fit3$results[[2]]))

knn_fits %>% gather(key = k_fits, value = accuracy, fit1, fit2, fit3) %>%
  ggplot(aes(x = k, y = accuracy, col = k_fits)) + geom_point() + geom_line()
```


accuracy
0.75
0.70
0.65
0.60
10
20
30
k
k_fits
fit1
fit2
fit3