HW_5_Geary

Marion Geary

2/15/2022

```
library(tidymodels)
setwd("/Users/Marion/Desktop/math386/hw/hw-4")
load('rad.Rdata')
```

Exercise 1

```
rad <- rad %>% mutate(Sex = as.factor(Sex))

knn_model <- nearest_neighbor(neighbors = 5, weight_func = "epanechnikov", dist_power = 2, mode = "clas

set.seed(12)
rad_split <- rad %>%
  initial_split(prop = .8)
rad_test <- testing(rad_split)
rad_train <- training(rad_split)

rad_recipe <- recipe(BinaryDiagnosis ~ ., data = rad) %>%
  step_dummy(Sex) %>%
  step_normalize(all_predictors())

rad_wkflow <- workflow() %>%
  add_model(knn_model) %>%
  add_recipe(rad_recipe)

set.seed(12)
rad_folds <- vfold_cv(rad_train, v = 10, repeats = 5)
```

```
my_metrics <- metric_set(sens, yardstick::spec, accuracy)

rad_pred <- control_resamples(save_pred = TRUE)

set.seed(12)
rad_res <- rad_wkflow %>% fit_resamples(resamples = rad_folds, control = rad_pred, metrics = my_metrics
```

Exercise 2

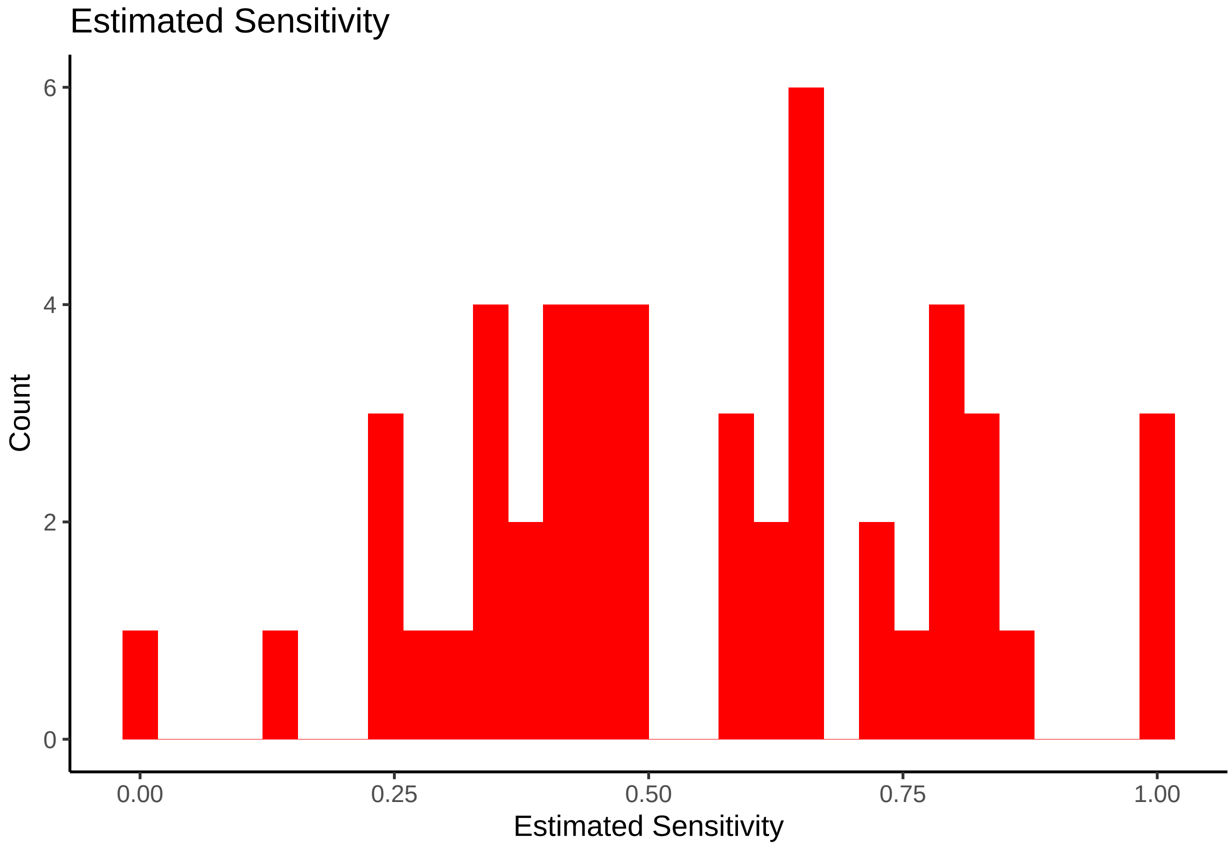
```
collect_metrics(rad_res, event_level = "second")
```

```
## # A tibble: 3 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.606    50  0.0175 Preprocessor1_Model1
## 2 sens     binary     0.556    50  0.0327 Preprocessor1_Model1
## 3 spec     binary     0.651    50  0.0214 Preprocessor1_Model1
```

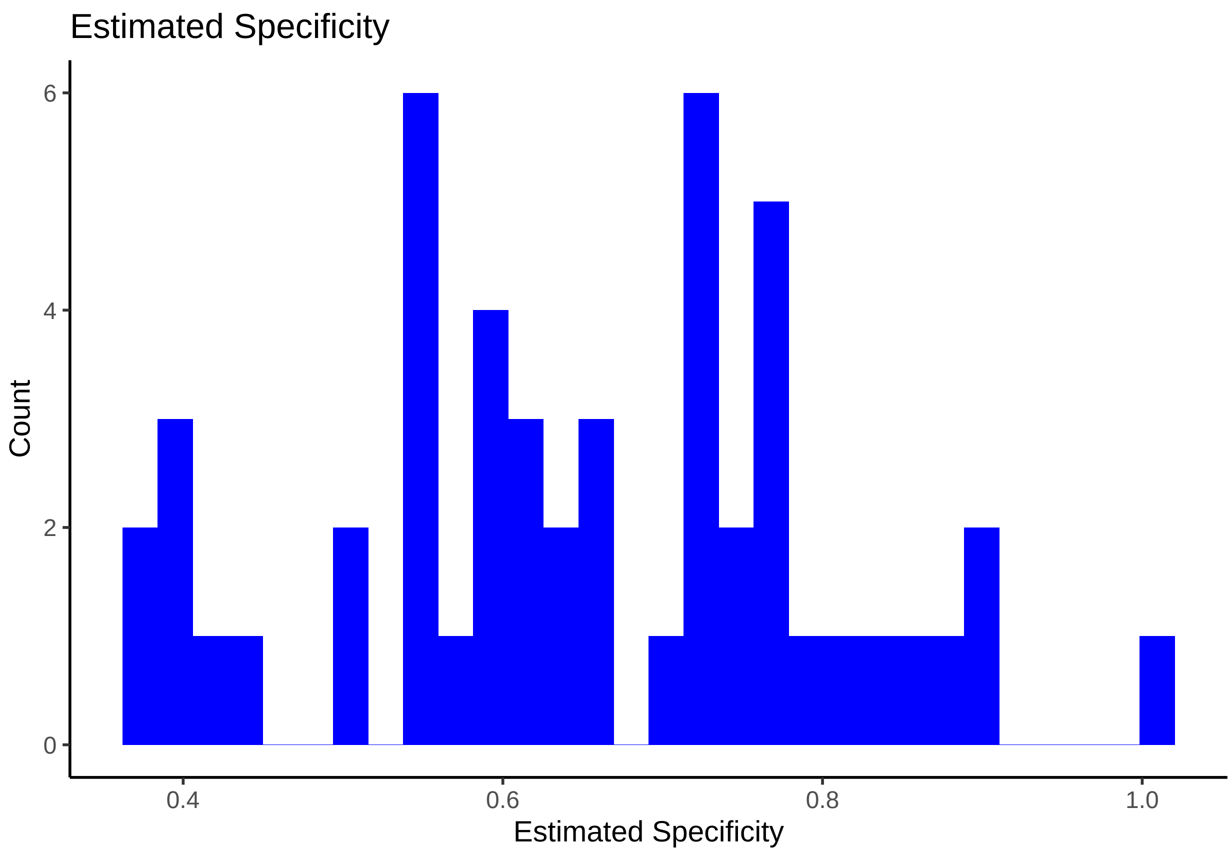
Compared to HW 4, all metrics are higher for this resampled model. The `accuracy` has improved the most, from 0.475 to 0.606. The sensitivity improved from 0.542 to 0.556. Specificity increased from 0.375 to 0.651. These show how resampling improves the model.

Exercise 3

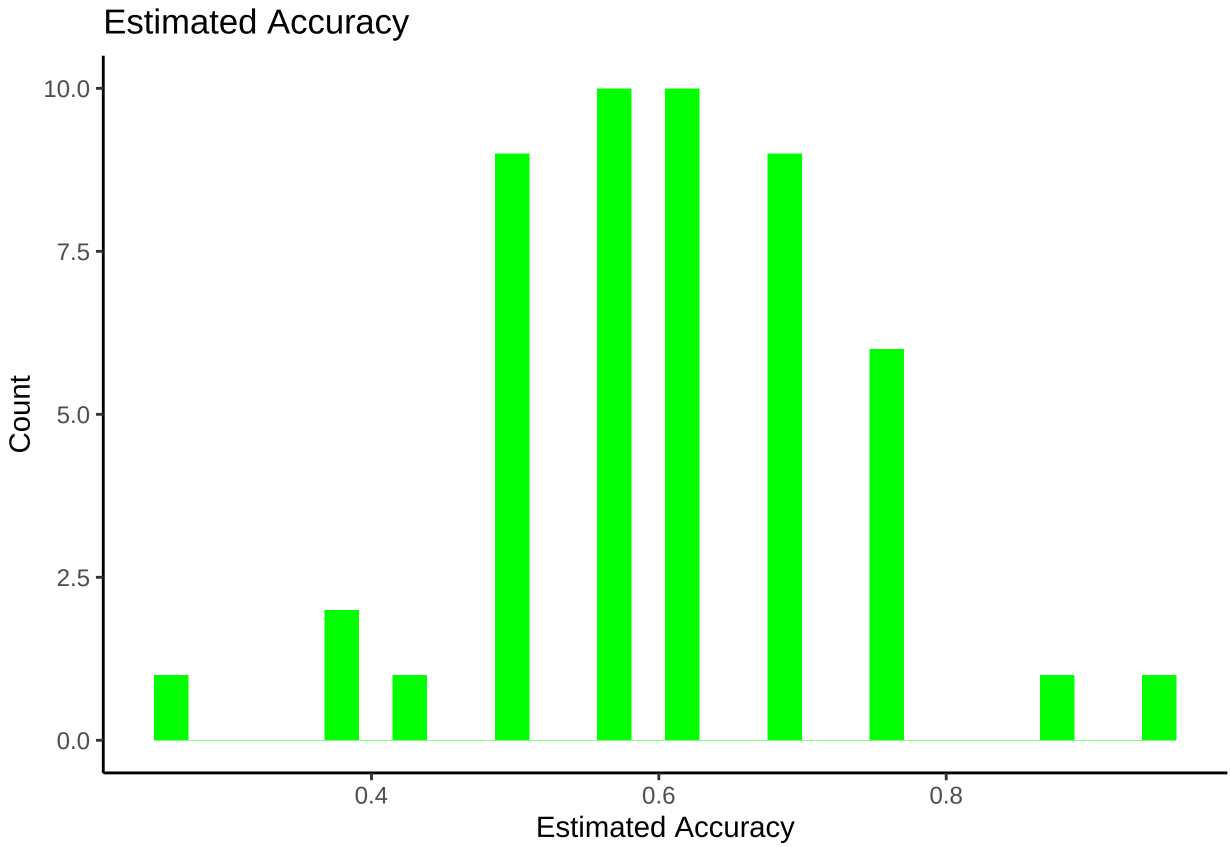
```
ggplot(collect_metrics(rad_res, summarize = F) %>%
         filter(.metric == "sens"), aes(x = .estimate)) + geom_histogram(fill = "red") + theme_classic(
```



```
ggplot(collect_metrics(rad_res, summarize = F) %>%
         filter(.metric == "spec"), aes(x = .estimate)) + geom_histogram(fill = "blue") + theme_classic
```



```
ggplot(collect_metrics(rad_res, summarize = F) %>%
         filter(.metric == "accuracy"), aes(x = .estimate)) + geom_histogram(fill = "green") + theme_cla
```



The estimated sensitivity graph shows that the sensitiviy has a roughly normal distribution centered around the mean, and a range from 0 to 1. The estimated specificity is semi-normal, with the most values between 0.6 and 0.8. The distribution has a smaller standard deviation, with no values below 0.3. The estimated accuracy graph has a roughly normal distribution with most of the values falling close to the mean. The values range from 0 to 1, but few folds have those extreme values.

Exercise 4

```
k_grid <- tibble(neighbors = seq(2, 20, by = 2))

knn_model <- nearest_neighbor(neighbors = tune(), weight_func = "epanechnikov", dist_power = 2, mode =

rad_wkflow <- workflow() %>%
  add_model(knn_model) %>%
  add_recipe(rad_recipe)

rad_res_2 <- rad_wkflow %>% tune_grid(resamples = rad_folds, grid = k_grid, metrics = my_metrics)

collect_metrics(rad_res_2, event_level = "second")
```

```
## # A tibble: 30 x 7
##    neighbors .metric  .estimator  mean     n std_err .config
##        <dbl> <chr>    <chr>      <dbl> <int>   <dbl> <chr>
##  1         2 accuracy binary     0.592    50  0.0167 Preprocessor1_Model01
##  2         2 sens     binary     0.567    50  0.0296 Preprocessor1_Model01
##  3         2 spec     binary     0.615    50  0.0211 Preprocessor1_Model01
##  4         4 accuracy binary     0.601    50  0.0165 Preprocessor1_Model02
##  5         4 sens     binary     0.547    50  0.0318 Preprocessor1_Model02
##  6         4 spec     binary     0.647    50  0.0202 Preprocessor1_Model02
##  7         6 accuracy binary     0.614    50  0.0173 Preprocessor1_Model03
##  8         6 sens     binary     0.564    50  0.0312 Preprocessor1_Model03
##  9         6 spec     binary     0.656    50  0.0215 Preprocessor1_Model03
## 10         8 accuracy binary     0.615    50  0.0182 Preprocessor1_Model04
## # ... with 20 more rows
```

```
show_best(rad_res_2, metric = "spec")
```

```
## # A tibble: 5 x 7
##   neighbors .metric .estimator  mean     n std_err .config
##       <dbl> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1         8 spec    binary     0.661    50  0.0209 Preprocessor1_Model04
## 2         6 spec    binary     0.656    50  0.0215 Preprocessor1_Model03
## 3        10 spec    binary     0.647    50  0.0222 Preprocessor1_Model05
## 4         4 spec    binary     0.647    50  0.0202 Preprocessor1_Model02
## 5        16 spec    binary     0.643    50  0.0247 Preprocessor1_Model08
```

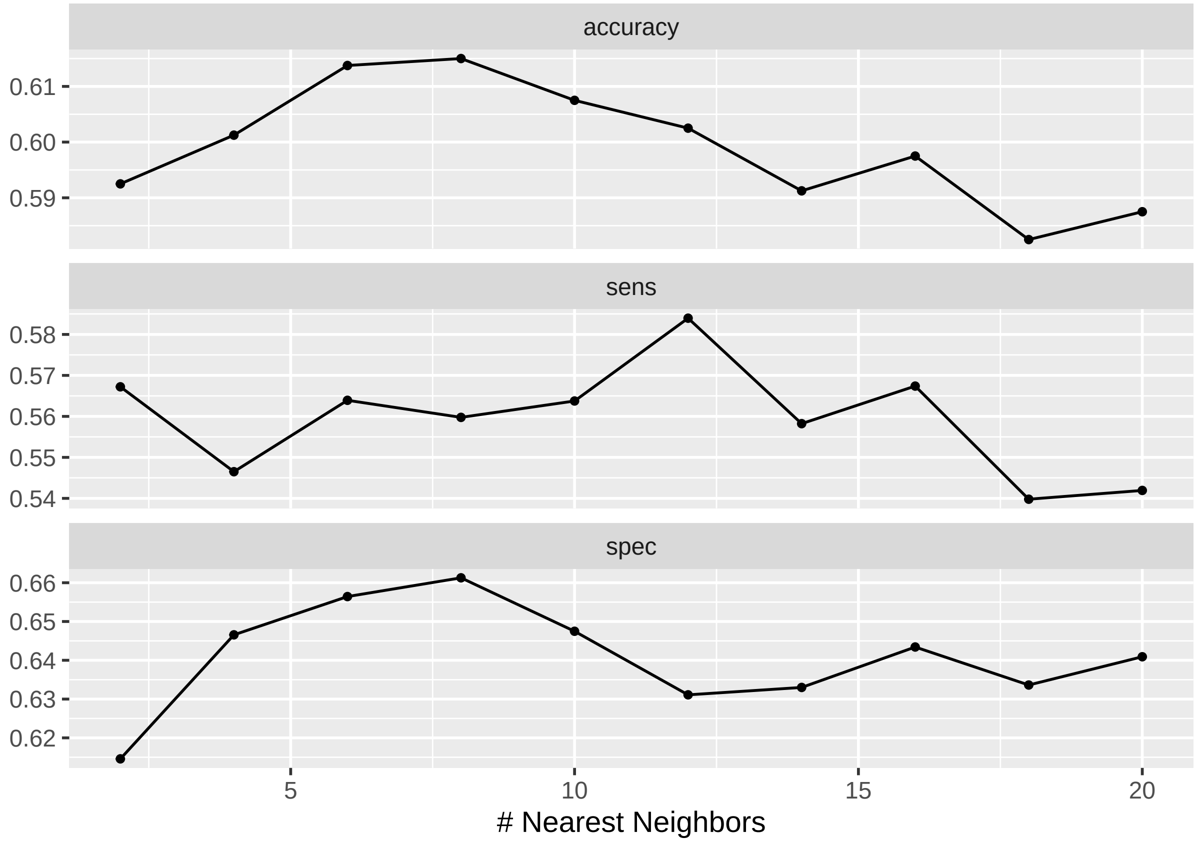
```
show_best(rad_res_2, metric = "sens")
```

```
## # A tibble: 5 x 7
##   neighbors .metric .estimator  mean     n std_err .config
##       <dbl> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1        12 sens    binary     0.584    50  0.0283 Preprocessor1_Model06
## 2        16 sens    binary     0.567    50  0.0276 Preprocessor1_Model08
## 3         2 sens    binary     0.567    50  0.0296 Preprocessor1_Model01
## 4         6 sens    binary     0.564    50  0.0312 Preprocessor1_Model03
## 5        10 sens    binary     0.564    50  0.0304 Preprocessor1_Model05
```

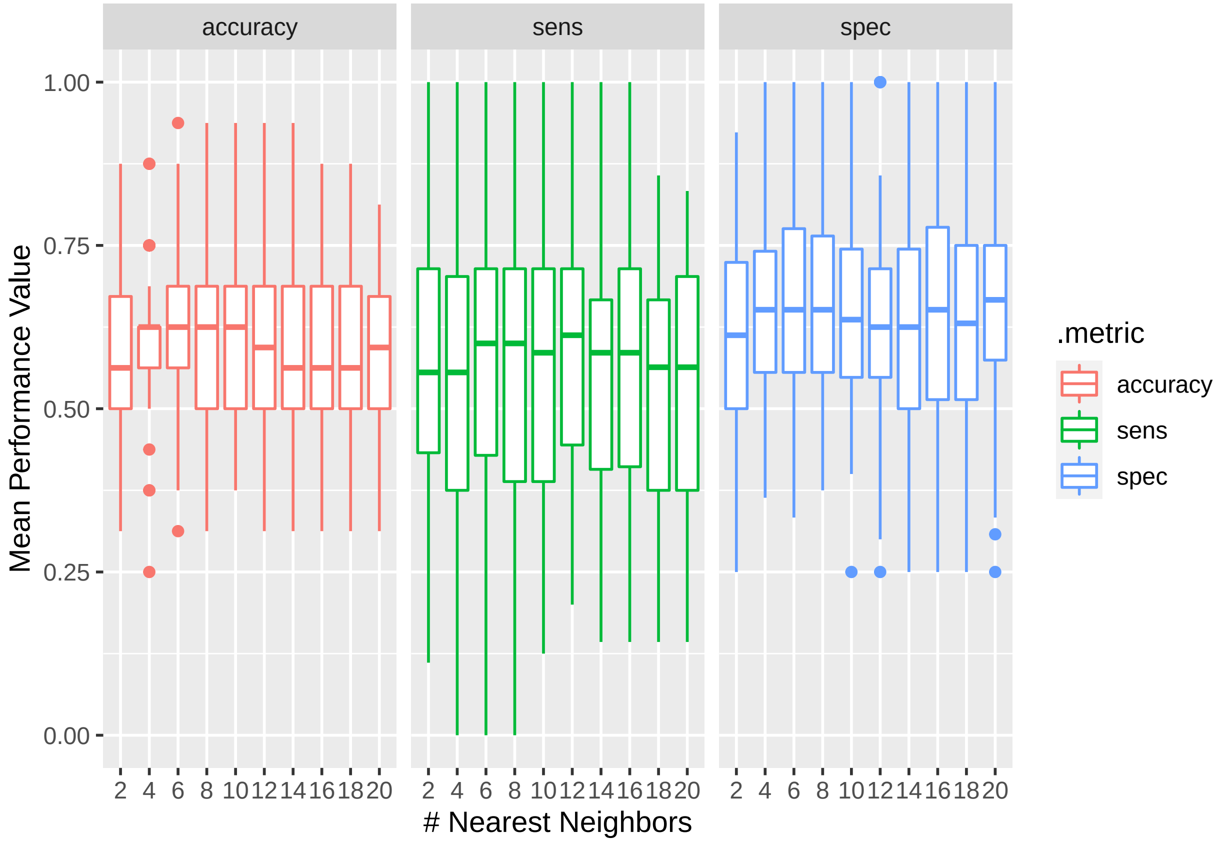
```
show_best(rad_res_2, metric = "accuracy")
```

```
## # A tibble: 5 x 7
##   neighbors .metric  .estimator  mean     n std_err .config
##       <dbl> <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1         8 accuracy binary     0.615    50  0.0182 Preprocessor1_Model04
## 2         6 accuracy binary     0.614    50  0.0173 Preprocessor1_Model03
## 3        10 accuracy binary     0.608    50  0.0185 Preprocessor1_Model05
## 4        12 accuracy binary     0.602    50  0.0178 Preprocessor1_Model06
## 5         4 accuracy binary     0.601    50  0.0165 Preprocessor1_Model02
```

```
autoplot(rad_res_2)
```



```
all_tun_res <- collect_metrics(rad_res_2, event_level = "second", summarize = F) %>% mutate(neighbors =

ggplot(all_tun_res) +
  geom_boxplot(aes(x = neighbors, y = .estimate, color = .metric)) +
  labs(x = "# Nearest Neighbors", y = "Mean Performance Value", main = "Resampling Estimates for Tuning
```



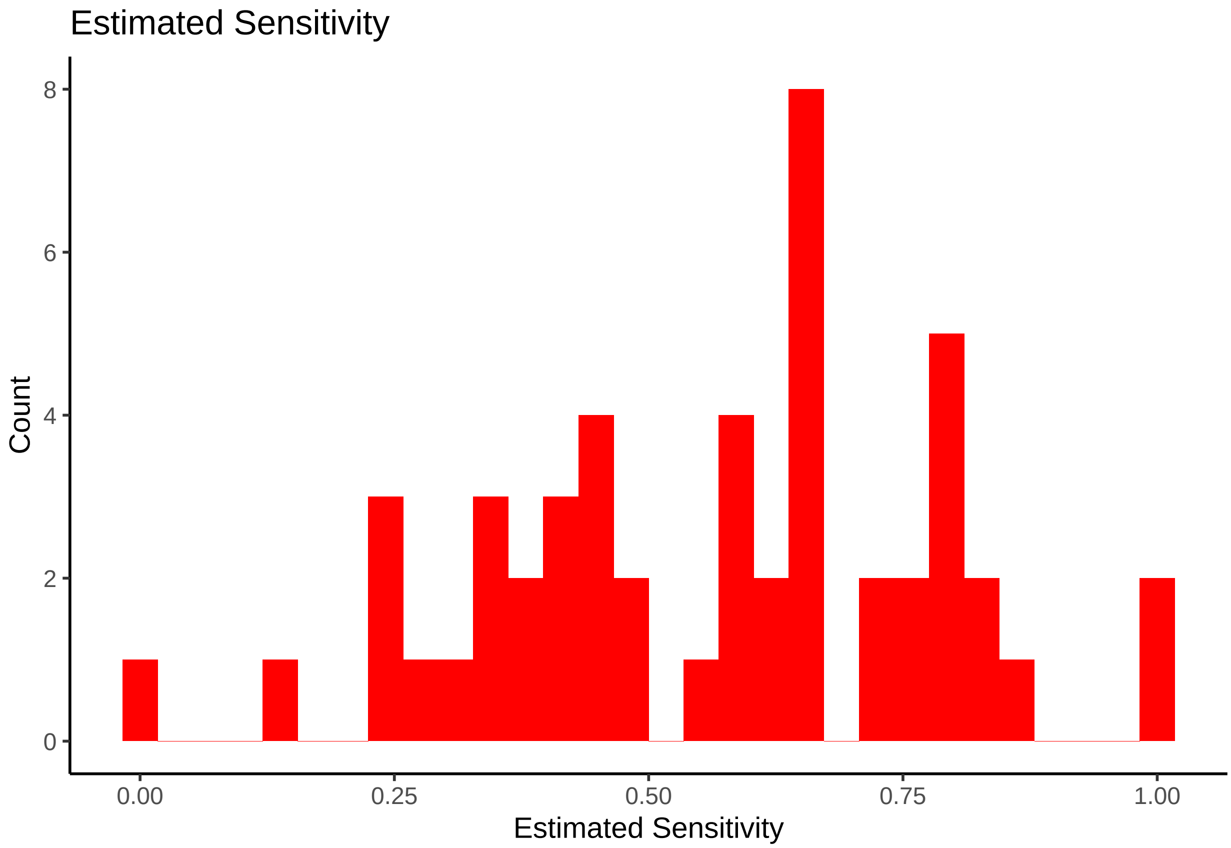
```
## pick k = 6 because it is the highest for all metrics

final_rad_wkflow <- rad_wkflow %>%
  finalize_workflow(list(neighbors = 6))

final_fit <- final_rad_wkflow %>% fit(data = rad_train)

final_rad_aug <- augment(final_fit, new_data = rad_test)

my_metrics(final_rad_aug, truth = BinaryDiagnosis, estimate = .pred_class, event_level = "second")
```

```
## # A tibble: 3 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 sens     binary         0.833
## 2 spec     binary         0.636
## 3 accuracy binary         0.725
```

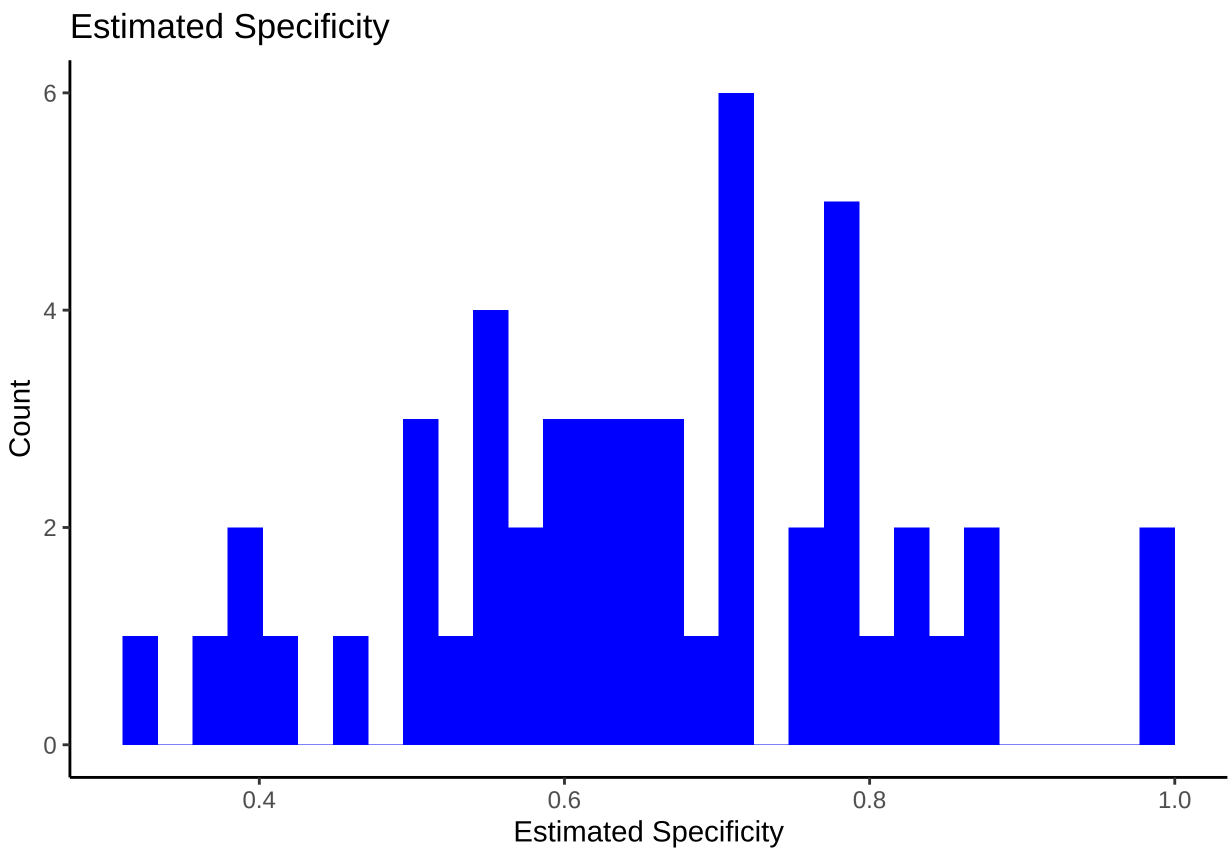
For the final model, I chose `k = 6` because in the tuning results, 6 was in the top 5 values for all 3 metrics. While it was not the highest for any individual metric, it was the most consistently high performing choice for `k`, making it the best choice for the final model.

Exercise 5

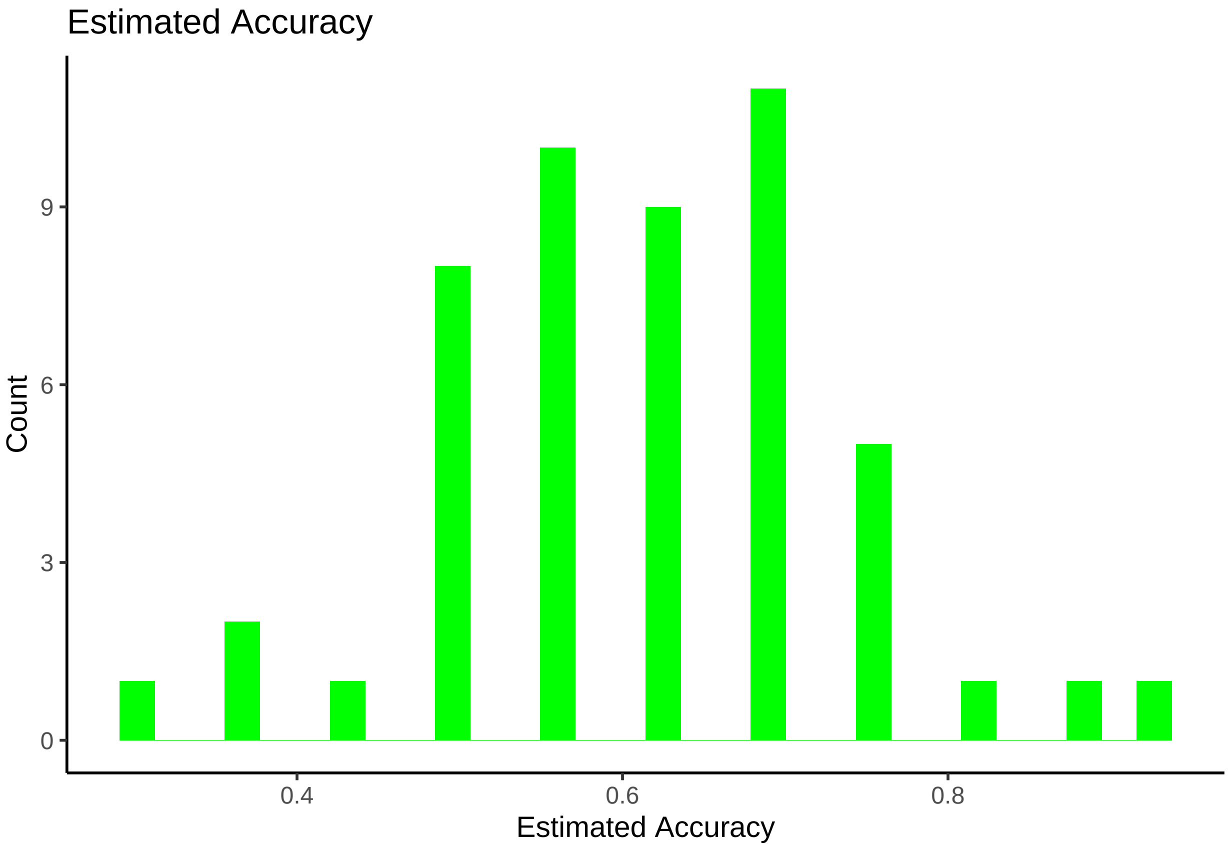
```
ggplot(collect_metrics(rad_res_2, summarize = F, event_level = "second") %>%
         filter(.metric == "sens") %>% filter(neighbors == 6), aes(x = .estimate)) + geom_histogram(fil
```



```
ggplot(collect_metrics(rad_res_2, summarize = F, event_level = "second") %>%
         filter(.metric == "spec") %>% filter(neighbors == 6), aes(x = .estimate)) + geom_histogram(fil
```



```
ggplot(collect_metrics(rad_res_2, summarize = F, event_level = "second") %>%
        filter(.metric == "accuracy") %>% filter(neighbors == 6), aes(x = .estimate)) + geom_histogram
```



The estimated sensitivity is pretty normal, with a large standard deviation. We see that the values are centered around the mean. The estimated specificity is also fairly normal, although the standard deviation is a bit smaller. The values from 0.2 to 1 rather than from 0 to 1. The estimated accuracy has a peak around the mean, with few values outside the range from 0.5 to 0.75. These distributions are very similar to the first set of distributions that did not include tuning.