Class Activity 22

Your name here

February 22 2024

Group Activity 1

Load the `mlbench` package to get `PimaIndiansDiabetes2` dataset.

```
# Load the data - diabetes
data(PimaIndiansDiabetes2)
db <- PimaIndiansDiabetes2
db <- db %>% drop_na() %>% mutate(diabetes = fct_rev(factor(diabetes)))
db_raw <- db %>% select(glucose, insulin, diabetes)
```

a. Split the data `75-25` into training and test set using the following code.

```
set.seed(123)

db_split <- initial_split(db, prop = 0.75)

# Create training data
db_train <- db_split %>% training()

# Create testing data
db_test <- db_split %>%  testing()
```

b. Follow the steps to train a 7-NN classifier using the `tidymodels` toolkit

```
# define recipe and preprocess the data
db_recipe <- recipe(diabetes ~ ., data = db_raw) %>%
  step_scale(all_predictors()) %>%
  step_center(all_predictors()) %>%
  prep()
```

```
# specify the model
db_knn_spec7 <- nearest_neighbor(mode = "classification",
                              engine = "kknn",
                              weight_func = "rectangular",
                              neighbors = 7)
```

```
# define the workflow
db_workflow <- workflow() %>%
  add_recipe(db_recipe) %>%
  add_model(db_knn_spec7)
```

```
# fit the model
db_fit <- fit(db_workflow, data = db_train)
```

c. Classify the penguins in the `test` data frame.

```
test_features <- db_test %>% select(glucose, insulin)
db_pred <- predict(db_fit, test_features, type = "raw")

db_results <- db_test %>%
  select(glucose, insulin, diabetes) %>%
  bind_cols(predicted = db_pred)

head(db_results, 6)
   glucose insulin diabetes predicted
4       89      94      neg       neg
7       78      88      pos       neg
15     166     175      pos       pos
19     103      83      neg       neg
32     158     245      pos       pos
36     103     192      neg       neg
```

Group Activity 2

Calculate the accuracy, sensitivity, specificity, and positive predictive value by hand using the following confusion matrix.

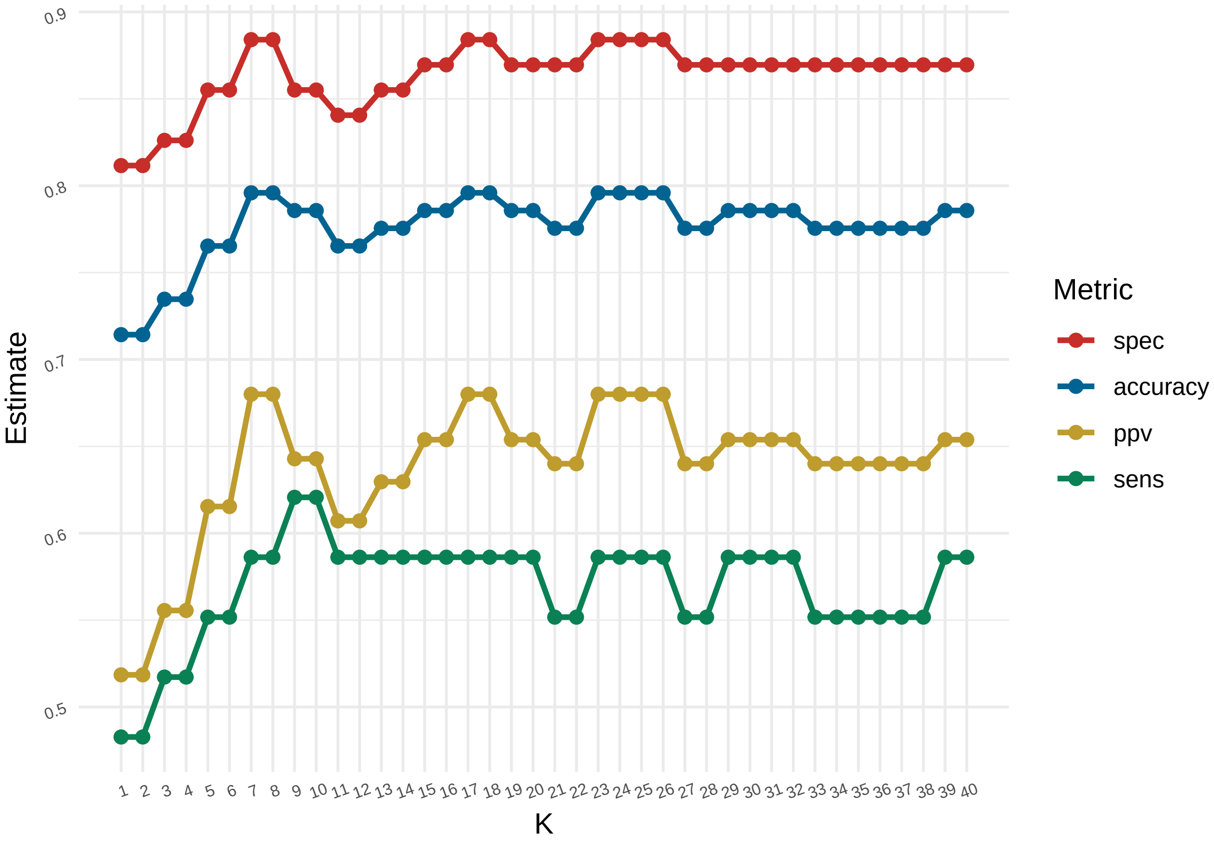
```
conf_mat(db_results, truth = diabetes, estimate = predicted)
          Truth
Prediction pos neg
       pos  17   8
       neg  12  61
```

```
accuracy(db_results, truth = diabetes,
         estimate = predicted)
# A tibble: 1 x 3
  .metric  .estimator .estimate
  <chr>    <chr>          <dbl>
1 accuracy binary         0.796

sens(db_results, truth = diabetes,
         estimate = predicted)
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 sens    binary         0.586

spec(db_results, truth = diabetes,
         estimate = predicted)
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
```

```
1 spec    binary         0.884

ppv(db_results, truth = diabetes,
         estimate = predicted)
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 ppv     binary          0.68
```

Extra: Code to recreate the plot in the `slides` for the `diabetes` dataset.

```
metrics_for_k <- function(k, db_train, db_test){
db_knn_spec <- nearest_neighbor(mode = "classification",
                             engine = "kknn",
                             weight_func = "rectangular",
                             neighbors = k)

db_knn_wkflow <- workflow() %>%
  add_recipe(db_recipe) %>%
  add_model(db_knn_spec)

db_knn_fit <- fit(db_knn_wkflow, data = db_train)
test_features <- db_test %>% select(glucose, insulin)
nn1_pred <- predict(db_knn_fit, test_features, type = "raw")

db_results <- db_test %>%
  select(diabetes) %>%
  bind_cols(predicted = nn1_pred)
custom_metrics <- metric_set(accuracy, sens, spec, ppv)

metrics <- custom_metrics(db_results,
               truth = diabetes,
               estimate = predicted)
metrics <- metrics %>% select(-.estimator) %>% mutate(k = rep(k,4))

return(list = metrics)
}


k <- seq(1,40, by=1)
optim.results <- purrr::map_df(k, ~metrics_for_k(.x, db_train, db_test))

optim.results %>%
  ggplot(aes(x = k, y = .estimate, color = forcats::fct_reorder2(.metric, k, .estimate ))) +
  geom_line(size = 1) +
  geom_point(size = 2) +
  theme_minimal() +
  ggthemes::scale_color_wsj() +
  scale_x_continuous(breaks = k) +
  theme(panel.grid.minor.x = element_blank(),
        axis.text=element_text(size=6, angle = 20))+
  labs(color='Metric', y = "Estimate", x = "K")
```



Estimate
0.9
0.8
0.7
0.6
0.5
K
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
Metric
spec
accuracy
ppv
sens