Week 10

Chang Dong

2023-04-07

```
library(ISLR)
library(tidymodels)
```

```
## ── Attaching packages ─────────────────────────────────────── tidymodels 1.0.0 ──
```

```
## ✔ broom        1.0.3     ✔ recipes      1.0.5
## ✔ dials        1.1.0     ✔ rsample      1.1.1
## ✔ dplyr        1.1.0     ✔ tibble       3.1.8
## ✔ ggplot2      3.4.1     ✔ tidyr        1.3.0
## ✔ infer        1.0.4     ✔ tune         1.0.1
## ✔ modeldata    1.1.0     ✔ workflows    1.1.3
## ✔ parsnip      1.0.4     ✔ workflowsets 1.0.0
## ✔ purrr        1.0.1     ✔ yardstick    1.1.0
```

```
## ── Conflicts ────────────────────────────────────── tidymodels_conflicts() ──
## ✖ purrr::discard() masks scales::discard()
## ✖ dplyr::filter()  masks stats::filter()
## ✖ dplyr::lag()     masks stats::lag()
## ✖ recipes::step()  masks stats::step()
## • Use tidymodels_prefer() to resolve common conflicts.
```

```
library(rpart)
```

```
##
## Attaching package: 'rpart'
```

```
## The following object is masked from 'package:dials':
##
##     prune
```

```
library(rpart.plot)
as_tibble(Carseats)
```

```
## # A tibble: 400 × 11
##    Sales CompPr…¹ Income Adver…² Popul…³ Price Shelv…⁴   Age Educa…⁵ Urban US
##    <dbl>    <dbl>  <dbl>   <dbl>   <dbl> <dbl> <fct>   <dbl>   <dbl> <fct> <fct>
##  1  9.5       138     73      11     276   120 Bad        42      17 Yes   Yes
##  2 11.2       111     48      16     260    83 Good       65      10 Yes   Yes
##  3 10.1       113     35      10     269    80 Medium     59      12 Yes   Yes
##  4  7.4       117    100       4     466    97 Medium     55      14 Yes   Yes
##  5  4.15      141     64       3     340   128 Bad        38      13 Yes   No
##  6 10.8       124    113      13     501    72 Bad        78      16 No    Yes
##  7  6.63      115    105       0      45   108 Medium     71      15 Yes   No
##  8 11.8       136     81      15     425   120 Good       67      10 Yes   Yes
##  9  6.54      132    110       0     108   124 Medium     76      10 No    No
## 10  4.69      132    113       0     131   124 Medium     76      17 No    Yes
## # … with 390 more rows, and abbreviated variable names ¹CompPrice,
## #   ²Advertising, ³Population, ⁴ShelveLoc, ⁵Education
```

```
Carseats <- Carseats %>% mutate("Sales_high" = ifelse(Sales > 8, "Yes", "No"))
```

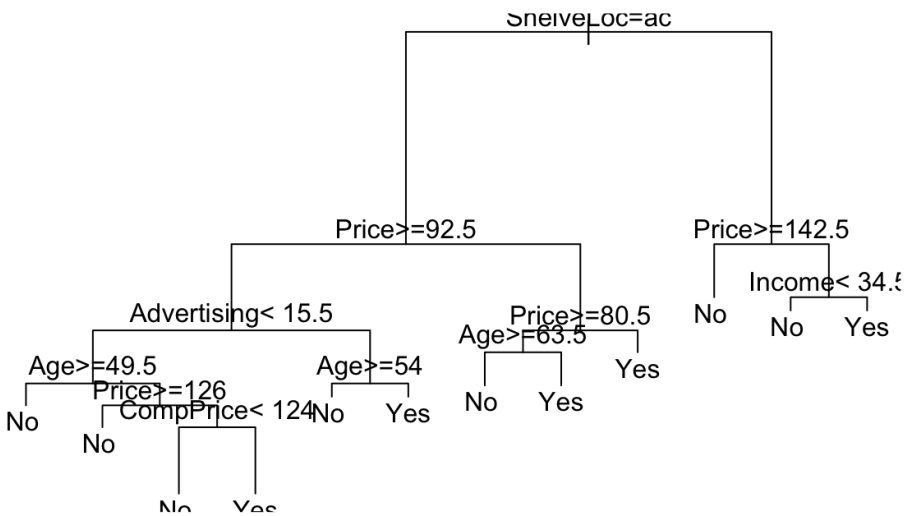
```
Carseats$Sales_high <- factor(Carseats$Sales_high)
```

```
Carseats <- Carseats %>% select(-Sales)
```

```
set.seed(2022)
car_split <- initial_split( Carseats )
car_train <- training( car_split )
car_test <- testing( car_split )
car_cv <- vfold_cv( car_train, v = 5 )
```

```
car_tree_spec <- decision_tree( mode = "classification" ) %>% set_engine( "rpart" )
```

```
car_tree <- car_tree_spec %>% fit( Sales_high ~ . , data = car_train)
plot( car_tree$fit )
text( car_tree$fit )
```

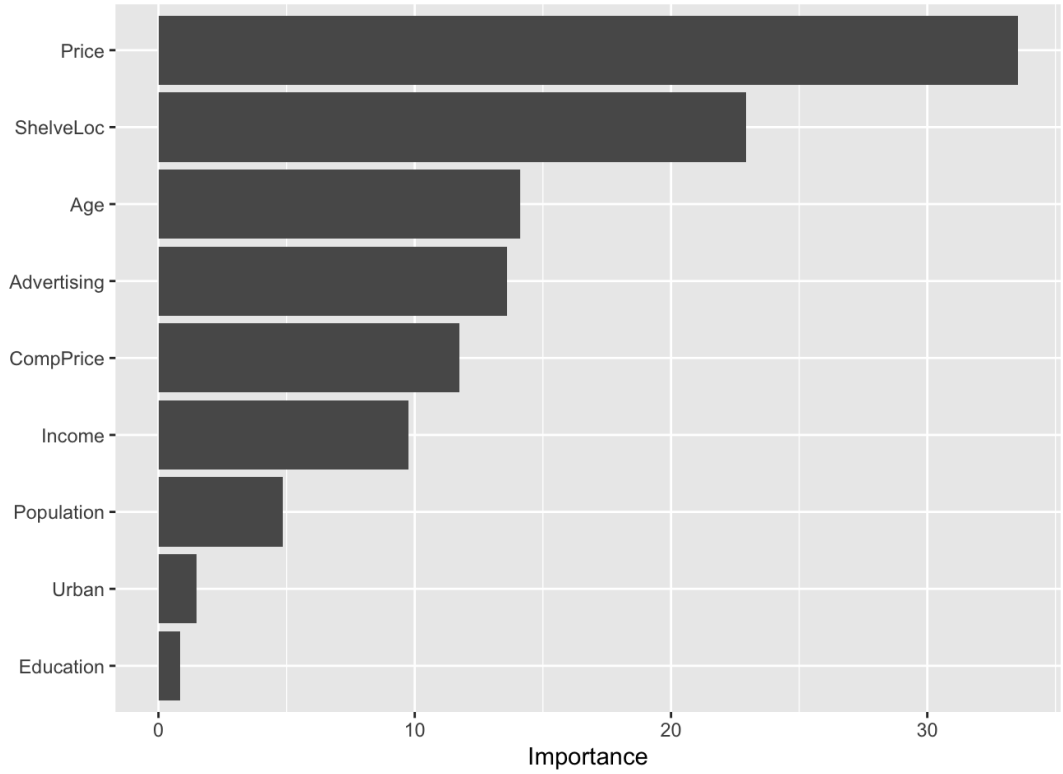



```
library(vip)
```

```
##
## Attaching package: 'vip'
```

```
## The following object is masked from 'package:utils':
##
##     vi
```

```
car_tree %>% vip()
```




```
car_tree_tune <- decision_tree(mode = "classification", cost_complexity = tune()) %>% set_engine("rpart")
```

```
cost_grid <- grid_regular( cost_complexity(), levels = 20 )
cost_grid
```

```
## # A tibble: 20 × 1
##    cost_complexity
##              <dbl>
##  1        1   e-10
##  2        2.98e-10
##  3        8.86e-10
##  4        2.64e- 9
##  5        7.85e- 9
##  6        2.34e- 8
##  7        6.95e- 8
##  8        2.07e- 7
##  9        6.16e- 7
## 10        1.83e- 6
## 11        5.46e- 6
## 12        1.62e- 5
## 13        4.83e- 5
## 14        1.44e- 4
## 15        4.28e- 4
## 16        1.27e- 3
## 17        3.79e- 3
## 18        1.13e- 2
## 19        3.36e- 2
## 20        1   e- 1
```

```
doParallel::registerDoParallel()
tree_tune <- tune_grid(object = car_tree_tune,

preprocessor = recipe(Sales_high ~ ., data = car_train), resamples = car_cv, grid = cost_grid)
```

```
collect_metrics(tree_tune)
```

```
## # A tibble: 40 × 7
##    cost_complexity .metric  .estimator  mean     n std_err .config
##              <dbl> <chr>    <chr>      <dbl> <int>   <dbl> <chr>
##  1        1   e-10 accuracy binary     0.76      5  0.0256 Preprocessor1_Model01
##  2        1   e-10 roc_auc  binary     0.778     5  0.0244 Preprocessor1_Model01
##  3        2.98e-10 accuracy binary     0.76      5  0.0256 Preprocessor1_Model02
##  4        2.98e-10 roc_auc  binary     0.778     5  0.0244 Preprocessor1_Model02
##  5        8.86e-10 accuracy binary     0.76      5  0.0256 Preprocessor1_Model03
##  6        8.86e-10 roc_auc  binary     0.778     5  0.0244 Preprocessor1_Model03
##  7        2.64e- 9 accuracy binary     0.76      5  0.0256 Preprocessor1_Model04
##  8        2.64e- 9 roc_auc  binary     0.778     5  0.0244 Preprocessor1_Model04
##  9        7.85e- 9 accuracy binary     0.76      5  0.0256 Preprocessor1_Model05
## 10        7.85e- 9 roc_auc  binary     0.778     5  0.0244 Preprocessor1_Model05
## # … with 30 more rows
```

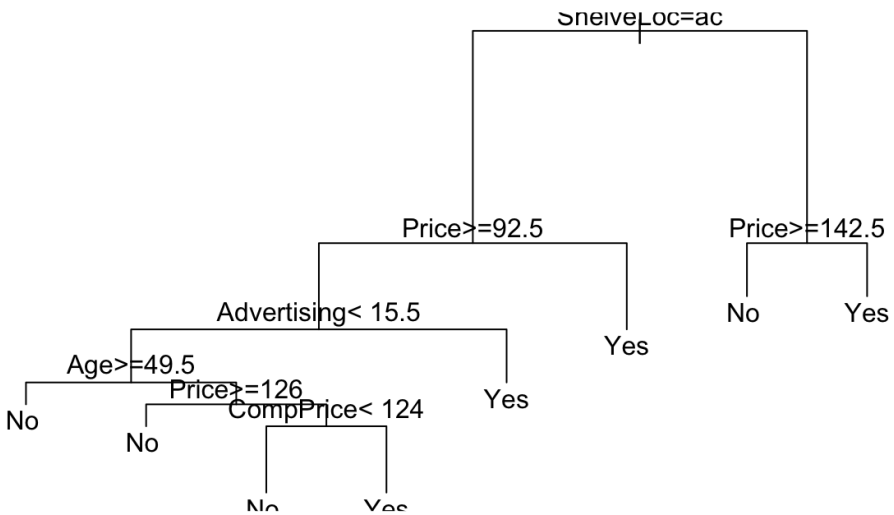
```
collect_metrics(tree_tune) %>% filter(mean==max(mean))
```

```
## # A tibble: 1 × 7
##   cost_complexity .metric .estimator  mean     n std_err .config
##             <dbl> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1          0.0336 roc_auc binary     0.781     5  0.0195 Preprocessor1_Model19
```

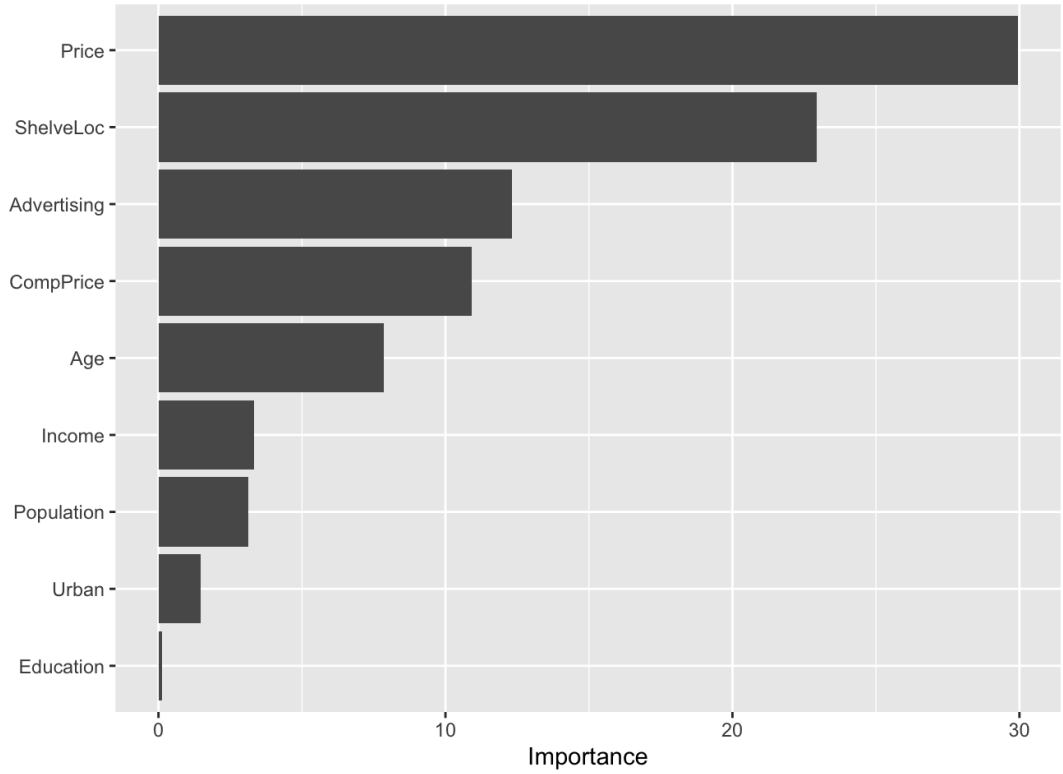
```
select_best( tree_tune, "roc_auc")
```

```
## # A tibble: 1 × 2
##   cost_complexity .config
##             <dbl> <chr>
## 1          0.0336 Preprocessor1_Model19
```

```
best_auc <- select_best( tree_tune, "roc_auc")
final_spec <- finalize_model( car_tree_tune, best_auc )
final_tree <- final_spec %>% fit( Sales_high ~ . , data = car_train )

plot( final_tree$fit )
text( final_tree$fit )
```




```
final_tree %>% vip()
```

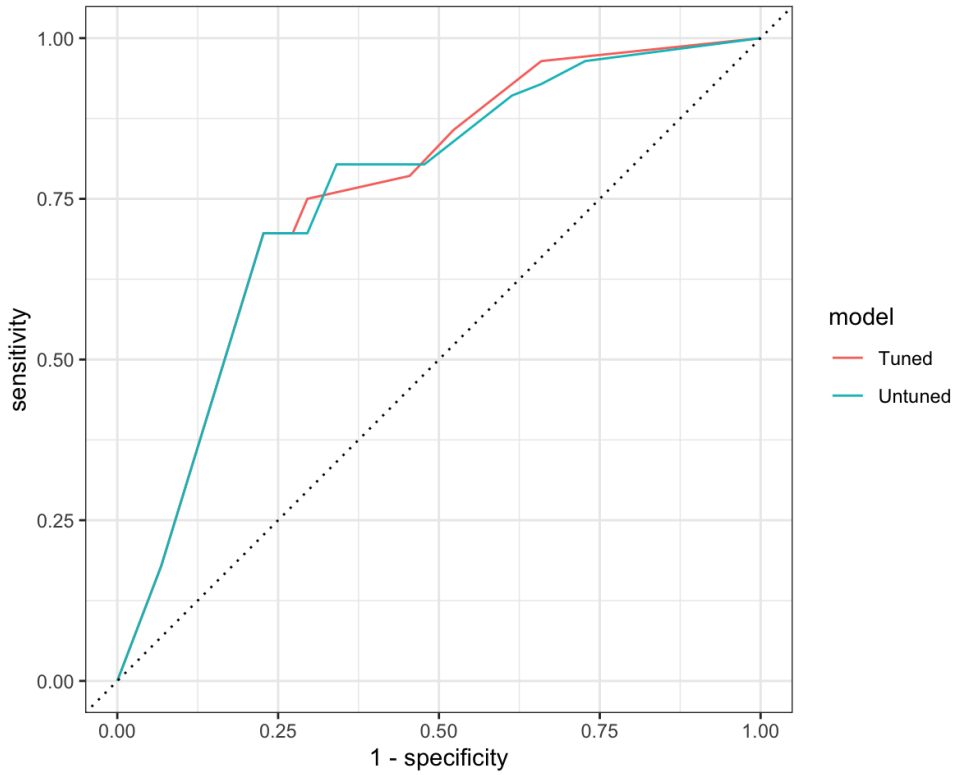



```
car_train_preds <- car_tree %>%

predict( new_data = car_test,type = "class") %>% bind_cols( car_test )

car_train_preds %>% metrics( truth = Sales_high, estimate = .pred_class)
```

```
## # A tibble: 2 × 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.71
## 2 kap      binary         0.401
```

```
car_preds <- predict( final_tree, # Get probability predict
                      new_data = car_test,

type = "prob" ) %>%

bind_cols( car_test %>% # Add on the truth.
  select( Sales_high ) %>%
    mutate( model = "Tuned" ) ) %>% # Add a vari
bind_rows( # Bind on the predictions from the orginal tre
  predict( car_tree, # Get probability predictions
           new_data = car_test,
            type = "prob" ) %>%

  bind_cols( car_test %>% # Add truth and model tacker
               select( Sales_high ) %>%

    mutate( model = "Untuned" ) )

)
car_preds %>% group_by( model ) %>% # Seperate by model type.
  roc_curve( truth = Sales_high, estimate = .pred_No ) %>% autoplot
```

```
## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## ℹ Please use `reframe()` instead.
## ℹ When switching from `summarise()` to `reframe()`, remember that `reframe()`
##   always returns an ungrouped data frame and adjust accordingly.
## ℹ The deprecated feature was likely used in the yardstick package.
##   Please report the issue at < ]8;;https://github.com/tidymodels/yardstick/issues https://github.com/ti
dymodels/yardstick/issues ]8;; >.
```




```
Gini_left=0.30*0.70 + 0.70*0.30

Gini_left
```

```
## [1] 0.42
```

```
Gini_right=0.74*0.26 + 0.26*0.74

Gini_right
```

```
## [1] 0.3848
```

```
Weighted_gini=0.78*Gini_left+0.22*Gini_right
Weighted_gini
```

```
## [1] 0.412256
```