Predicting children in hotel bookings

Suggested answers

APPLICATION EXERCISE ANSWERS

MODIFIED
September 12, 2024

Your Turn 1

Run the chunk below and look at the output. Then, copy/paste the code and edit to create:

- a decision tree model for classification
- that uses the `C5.0` engine.

Save it as `tree_mod` and look at the object. What is different about the output?

Hint: you'll need https://www.tidymodels.org/find/parsnip/

```
lr_mod <- logistic_reg() |>
  set_engine(engine = "glm") |>
  set_mode("classification")
lr_mod
```

```
Logistic Regression Model Specification (classification)

Computational engine: glm
```

```
tree_mod <- decision_tree() |>
  set_engine(engine = "C5.0") |>
  set_mode("classification")
tree_mod
```

```
Decision Tree Model Specification (classification)

Computational engine: C5.0
```

Your Turn 2

Fill in the blanks.

Use `initial_split()`, `training()`, and `testing()` to:

1. Split **hotels** into training and test sets. Save the rsplit!

2. Extract the training data and fit your classification tree model.

3. Check the proportions of the `test` variable in each set.

Keep `set.seed(100)` at the start of your code.

Hint: Be sure to remove every _ before running the code!

```
set.seed(100) # Important!

hotels_split <- initial_split(data = hotels, prop = 3 / 4)
hotels_train <- training(hotels_split)
hotels_test <- testing(hotels_split)

# check distribution
count(x = hotels_train, children) |>
  mutate(prop = n / sum(n))
```

```
# A tibble: 2 × 3
  children     n  prop
  <fct>    <int> <dbl>
1 children  1503 0.501
2 none      1497 0.499
```

```
count(x = hotels_test, children) |>
  mutate(prop = n / sum(n))
```

```
# A tibble: 2 × 3
  children     n  prop
  <fct>    <int> <dbl>
1 children   497 0.497
2 none       503 0.503
```

Your Turn 3

Run the code below. What does it return?

```
set.seed(100)
hotels_folds <- vfold_cv(data = hotels_train, v = 10)
hotels_folds
```

```
#  10-fold cross-validation
# A tibble: 10 × 2
   splits             id
   <list>             <chr>
 1 <split [2700/300]> Fold01
```

```
 2 <split [2700/300]> Fold02
 3 <split [2700/300]> Fold03
 4 <split [2700/300]> Fold04
 5 <split [2700/300]> Fold05
 6 <split [2700/300]> Fold06
 7 <split [2700/300]> Fold07
 8 <split [2700/300]> Fold08
 9 <split [2700/300]> Fold09
10 <split [2700/300]> Fold10
```

Your Turn 4

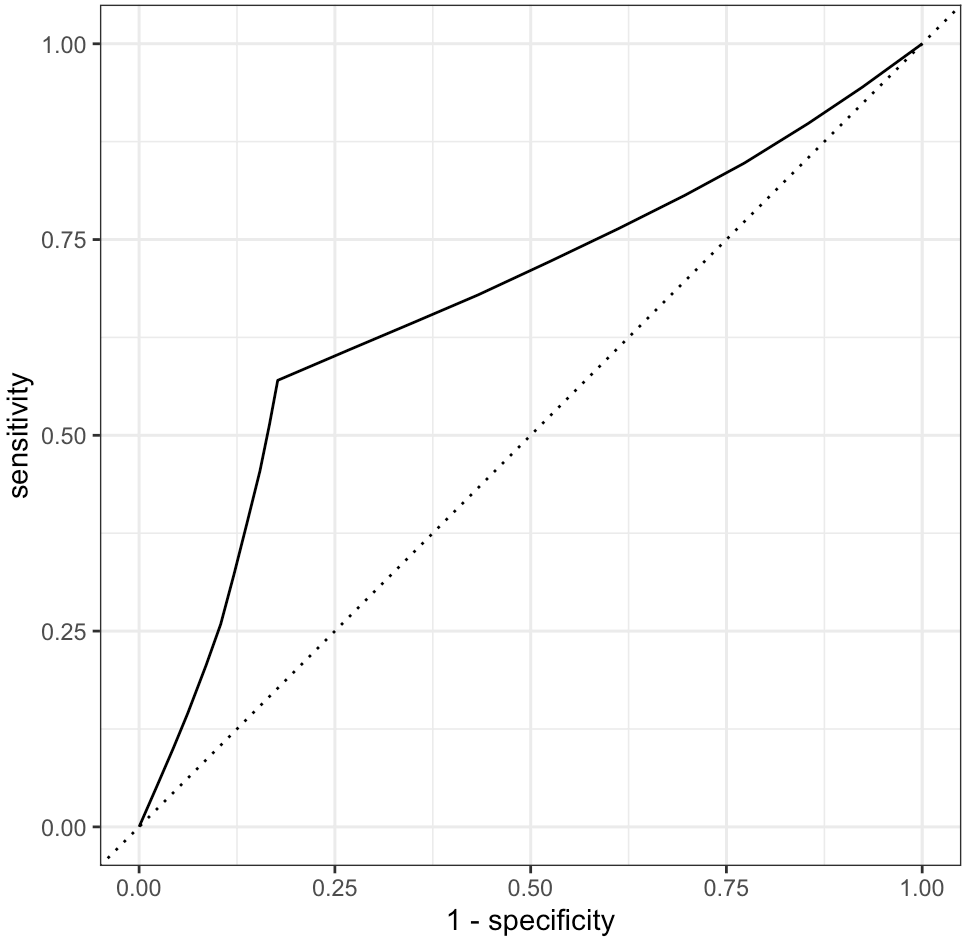
Add a `autoplot()` to visualize the ROC AUC. How well does the model perform?

```
tree_preds <- tree_mod |>
  fit_resamples(
    children ~ average_daily_rate + stays_in_weekend_nights,
    resamples = hotels_folds,
    control = control_resamples(save_pred = TRUE)
  )

tree_preds |>
  collect_predictions() |>
  roc_auc(truth = children, .pred_children)
```

```
# A tibble: 1 × 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 roc_auc binary         0.670
```

```
tree_preds |>
  collect_predictions() |>
  roc_curve(truth = children, .pred_children) |>
  autoplot()
```



It's moderately successful. Better than 0.5, but still has a lot of room for improvement.

Acknowledgments

- Materials derived from Tidymodels, Virtually by Allison Hill and licensed under a Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA) License.
- Dataset and some modeling steps derived from A predictive modeling case study and licensed under a Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA) License.

Session information