Model Fitting Part 2

Joe Martin

10/18/2021

Introduction

The goal of this section is to determine whether it is possible to predict nausea based on other symptoms a patient is displaying. This is first tested with all present variables as predictors, then tested against the main predictor of interest, Runny Nose.

```
# Write code that takes the data and splits it randomly into a train and test that, following for insta

# I messed this part up and created a new Body Temp variable. This is unnecessary, but I'm leaving it.

set.seed(2)

df$high_temp <- ifelse(df$BodyTemp > 99, "high_temp","normal_range")
df$high_temp <- factor(df$high_temp)

# Use 70/30 split on data

data_split <- initial_split(df, prop = 7/10)

train_data <- training(data_split)
test_data <- testing(data_split)
```

Testing All Predictors

The following code generates recipes for the training and test sets of data, then builds a workflow to fit to a logistic model to all predictor variables.

```
# Next, following the example in the Create Recipes section of the Get Started tidymodels tutorial, cre

nausea_rec <- recipe(Nausea ~ ., data = train_data)
nausea_test <- recipe(Nausea ~ ., data = test_data)

#summary(ht_rec)
```

Set up logistic model and create workflow

```
lr_model <- logistic_reg() %>%
  set_engine("glm")

nausea_workflow <-
```

```
  workflow() %>%
  add_model(lr_model) %>%
  add_recipe(nausea_rec)
```

Show relationships between predictor variables and outcome

```
nausea_fit <- nausea_workflow %>%
  fit(data = train_data)

nausea_fit %>%
  extract_fit_parsnip() %>%
  tidy()
```

```
## # A tibble: 39 x 5
##    term                 estimate std.error statistic p.value
##    <chr>                   <dbl>     <dbl>     <dbl>   <dbl>
##  1 (Intercept)          -20.1       15.1      -1.33    0.184
##  2 SwollenLymphNodesYes  -0.270      0.240    -1.12    0.262
##  3 ChestCongestionYes     0.205      0.267     0.767   0.443
##  4 ChillsSweatsYes        0.0574     0.353     0.163   0.871
##  5 NasalCongestionYes     0.342      0.306     1.12    0.264
##  6 CoughYNYes            -0.152      0.623    -0.244   0.807
##  7 SneezeYes              0.0453     0.264     0.171   0.864
##  8 FatigueYes             0.599      0.476     1.26    0.208
##  9 SubjectiveFeverYes     0.328      0.292     1.12    0.261
## 10 HeadacheYes            0.439      0.365     1.20    0.229
## # ... with 29 more rows
```

The following code predicts whether a patient has nausea based on the model built above.

```
# use predict to predict if patient has nausea
predict(nausea_fit, test_data)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

The following code shows the model's predictions for the test data set

```
nausea_aug <- augment(nausea_fit, test_data)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

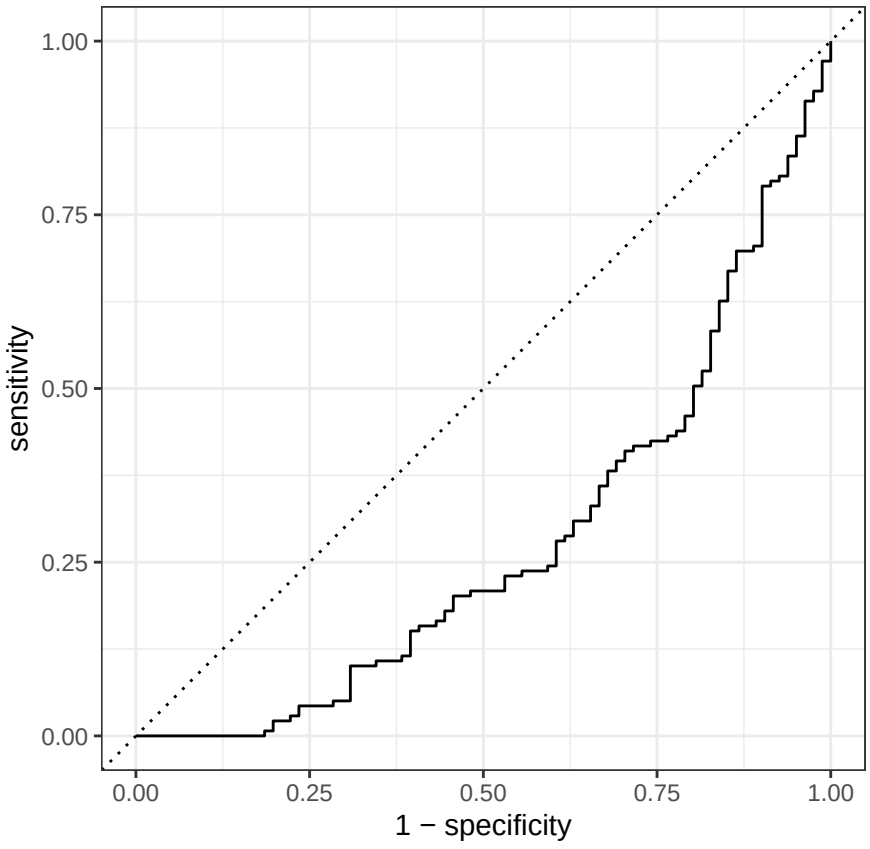
```
nausea_aug %>% select(Nausea, .pred_class, .pred_Yes, .pred_No)
```

```
## # A tibble: 220 x 4
##    Nausea .pred_class .pred_Yes .pred_No
```

```
##    <fct>  <fct>           <dbl>    <dbl>
##  1 No     No              0.229   0.771
##  2 Yes    Yes             0.966   0.0342
##  3 Yes    Yes             0.949   0.0514
##  4 No     No              0.224   0.776
##  5 Yes    No              0.243   0.757
##  6 Yes    No              0.142   0.858
##  7 Yes    No              0.215   0.785
##  8 No     No              0.222   0.778
##  9 No     No              0.210   0.790
## 10 Yes    Yes             0.715   0.285
## # ... with 210 more rows
```

The results of the ROC curve the the ROC-AUC value of .29 show the model not ideal for predicting nausea.

```
#Follow the example in the Use a trained workflow to predict section of the tutorial to look at the pre

nausea_aug %>%
  roc_curve(truth = Nausea, .pred_Yes) %>%
  autoplot()
```




```
nausea_aug %>%
  roc_auc(truth = Nausea, .pred_Yes)
```

```
## # A tibble: 1 x 3
```

```
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.278
```

Testing the Main Predictor

Runny Nose

The following model will test whether having a runny nose is a good predictor of nausea. Based on the results testing all variables as predictors of nausea, it is unlikely that having a runny nose will predict nausea (p-value of .699).

```
#Let's re-do the fitting but now with a model that only fits the main predictor to the categorical outc

# Create new recipes
RunnyNose_rec <- recipe(Nausea ~ RunnyNose, data = train_data)
RunnyNose_test <- recipe(Nausea ~ RunnyNose, data = test_data)
```

```
RunnyNose_workflow <-
  workflow() %>%
  add_model(lr_model) %>%
  add_recipe(RunnyNose_rec)
```

```
RunnyNose_fit <- RunnyNose_workflow %>%
  fit(data = train_data)
```

```
RunnyNose_fit %>%
  extract_fit_parsnip() %>%
  tidy()
```

```
## # A tibble: 2 x 5
##   term         estimate std.error statistic  p.value
##   <chr>           <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)  -0.661       0.178   -3.72   0.000198
## 2 RunnyNoseYes  0.00462     0.209    0.0221 0.982
```

```
# use predict to predict if patient has a high temperature
predict(RunnyNose_fit, test_data)
```

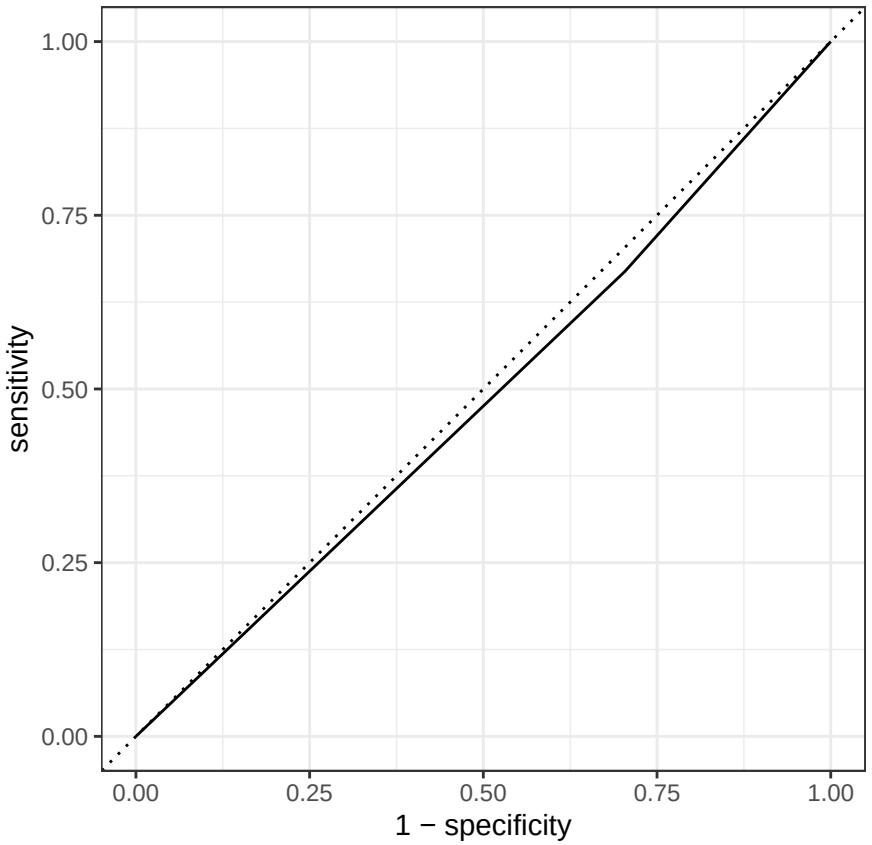
```
RunnyNose_aug <- augment(RunnyNose_fit, test_data)

RunnyNose_aug %>% select(Nausea, .pred_class, .pred_Yes, .pred_No)
```

```
## # A tibble: 220 x 4
##    Nausea .pred_class .pred_Yes .pred_No
##    <fct>  <fct>           <dbl>    <dbl>
##  1 No     No              0.340    0.660
##  2 Yes    No              0.340    0.660
##  3 Yes    No              0.340    0.660
##  4 No     No              0.341    0.659
```

```
##  5 Yes    No              0.341    0.659
##  6 Yes    No              0.341    0.659
##  7 Yes    No              0.341    0.659
##  8 No     No              0.341    0.659
##  9 No     No              0.341    0.659
## 10 Yes    No              0.341    0.659
## # ... with 210 more rows
```

Once again, the ROC curve and ROC-AUC value (.48) indicate that RunnyNose is not a good predictor for Nausea.

```
RunnyNose_aug %>%
  roc_curve(truth = Nausea, .pred_Yes) %>%
  autoplot()
```




```
RunnyNose_aug %>%
  roc_auc(truth = Nausea, .pred_Yes)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.483
```