Prediction and Model Performance

ECO 6416

2022-10-18

Contents

Here are all the packages needed to get started.

```
library(readxl) # reading in excel file
library(dplyr) # for pipe operator
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
sessionInfo()
```

```
## R version 4.2.1 (2022-06-23 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19044)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] dplyr_1.0.9  readxl_1.4.1
##
```

```
## loaded via a namespace (and not attached):
##  [1] rstudioapi_0.14  knitr_1.39       magrittr_2.0.3   tidyselect_1.1.2
##  [5] R6_2.5.1         rlang_1.0.3      fastmap_1.1.0    fansi_1.0.3
##  [9] stringr_1.4.0    tools_4.2.1      xfun_0.31        utf8_1.2.2
## [13] DBI_1.1.3        cli_3.3.0        htmltools_0.5.2  ellipsis_0.3.2
## [17] assertthat_0.2.1 yaml_2.3.5       digest_0.6.29    tibble_3.1.7
## [21] lifecycle_1.0.1  crayon_1.5.1     purrr_0.3.4      vctrs_0.4.1
## [25] glue_1.6.2       evaluate_0.15    rmarkdown_2.14   stringi_1.7.8
## [29] compiler_4.2.1   pillar_1.7.0     cellranger_1.1.0 generics_0.1.3
## [33] pkgconfig_2.0.3
```

1 Modeling College Football Attendance

Let's bring in the College Football Attendance dataset:

```
attend <- read_excel("../Data/College Football Attendance.xlsx")[,-1] #dropped names of teams
```

- AttendAv= Division I-A Football attendance (in thousands)
- Top25CNN= No of times the team is ranked in top 25 by CNN ratings
- Win%10Yr= Average winning percentage in the last 10 years
- ProgAge= Age of the football program
- Enrolmt=Total enrollment of students in the university (in thousands)

1.1 Split the Data

To see how our model does before deploying it in the wild, we can use randomly omit some data, run the regression on the remaining, then calculate performance metrics on the data we omitted.

```
set.seed(123456)

index <- sample(seq_len(nrow(attend)), size = 5)

train <- attend[-index,]
test <- attend[index,]
```

Check the summary statistics:

```
summary(train)
```

```
##     AttendAv         Top25CNN      WinPercentTenYr    ProgAge
##  Min.   :  4.70   Min.   : 0.000   Min.   :38.00   Min.   : 25.00
##  1st Qu.: 23.57   1st Qu.: 0.000   1st Qu.:51.00   1st Qu.: 80.00
##  Median : 35.35   Median : 1.000   Median :55.00   Median : 97.00
##  Mean   : 39.80   Mean   : 2.533   Mean   :55.76   Mean   : 90.26
##  3rd Qu.: 51.67   3rd Qu.: 3.000   3rd Qu.:60.00   3rd Qu.:101.00
##  Max.   :105.70   Max.   :11.000   Max.   :76.00   Max.   :123.00
##     Enrollmt
##  Min.   : 2.00
##  1st Qu.:15.00
##  Median :21.00
##  Mean   :21.88
##  3rd Qu.:27.75
##  Max.   :49.00
```

```
summary(test)
```

```
##     AttendAv        Top25CNN   WinPercentTenYr    ProgAge         Enrollmt
```

```
##  Min.   :33.20   Min.   :0.0   Min.   :57      Min.   : 70.0   Min.   :16.0
##  1st Qu.:38.90   1st Qu.:1.0   1st Qu.:59      1st Qu.:100.0   1st Qu.:24.0
##  Median :40.80   Median :2.0   Median :64      Median :102.0   Median :30.0
##  Mean   :44.46   Mean   :3.4   Mean   :64      Mean   : 95.8   Mean   :31.8
##  3rd Qu.:51.60   3rd Qu.:6.0   3rd Qu.:70      3rd Qu.:103.0   3rd Qu.:37.0
##  Max.   :57.80   Max.   :8.0   Max.   :70      Max.   :104.0   Max.   :52.0
```

1.2 Run the Model

```
model <- lm(AttendAv ~. , data = train)

summary(model)
```

```
##
## Call:
## lm(formula = AttendAv ~ ., data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -30.227 -16.469  -6.602  12.209  64.162
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     34.26724   24.73760   1.385    0.170
## Top25CNN        -0.93134    1.00671  -0.925    0.358
## WinPercentTenYr -0.11435    0.42459  -0.269    0.788
## ProgAge          0.17262    0.14603   1.182    0.240
## Enrollmt        -0.05994    0.24488  -0.245    0.807
##
## Residual standard error: 22.59 on 85 degrees of freedom
## Multiple R-squared:  0.04124,    Adjusted R-squared:  -0.003875
## F-statistic: 0.9141 on 4 and 85 DF,  p-value: 0.4596
```

1.3 Prediction of UCF Attendance

Suppose we wanted to predict UCF attendance. Based off some basic googling, suppose UCF has the following:

```
UCF <- data.frame(Top25CNN = 0,
                  WinPercentTenYr = 58,
                  ProgAge = 43,
                  Enrollmt = 70)

predict(model, newdata = UCF,interval = "prediction")
```

```
##        fit      lwr      upr
## 1 30.86175 -23.9189 85.64239
```

```
predict(model, newdata = UCF,interval = "confidence")
```

```
##        fit       lwr      upr
## 1 30.86175 -0.496176 62.21967
```

The point estimate is the same, but you can see the lower and upper bounds are wider. One thing to note here is that R did not identify this as extrapolation error. Other software may tell you about this.

1.4 Calculating Performance Measures

Let's see how the model predicts on our test dataset.

```
test$Prediction <- predict(model, newdata = test)
```

1.4.1 Calculating Error

Recall the formula for calculating error:

$$Error = Forecasted - Actual$$

```
test$error <- test$Prediction - test$AttendAv
```

1.4.2 Bias

The bias is simply the average of those errors

```
mean(test$error)
```

```
## [1] -6.046469
```

On average, there is a negative bias (model under predicts attendance).

1.4.3 Mean Absolute Error

```
test$error %>%
  abs() %>%
  mean()
```

```
## [1] 11.60584
```

1.4.4 Root Mean Squared Error

```
test$error^2 %>%
  mean() %>%
  sqrt()
```

```
## [1] 13.29218
```

1.4.5 MAPE

```
(test$error/test$AttendAv) %>%
  abs() %>%
  mean()
```

```
## [1] 0.2503241
```

A MAPE less than 5% is considered as an indication that the forecast is acceptably accurate. A MAPE greater than 10% but less than 25% indicates low, but acceptable accuracy and MAPE greater than 25% very low accuracy, so low that the forecast is not acceptable in terms of its accuracy[1]

If you had competing models, you would look at these metrics from both and decide on a model going forward.

[1] On the Relationship among Values of the Same Summary Measure of Error when it is used across Multiple Characteristics at the Same Point in Time: An Examination of MALPE and MAPE; Dr. David A. Swanson