Lab 04: The Office

Feature engineering

Stats is 'Fun': Thomas Barker, Dav King, Harry Liu

2022-10-11

Setup

Load packages and data:

```r
library(tidyverse)
library(tidymodels)
library(schrute) #install.packages("schrute")
library(lubridate)
library(knitr)
```

[Select this page for the "Workflow & formatting" and "Team agreement" sections in Gradescope.]

Exercises

```r
theoffice <- theoffice |>
  mutate(air_date = ymd(as.character(air_date)))
```

Exercise 1

```r
theoffice <- theoffice |>
  mutate(
    text = str_to_lower(text),
    halloween_mention = if_else(str_detect(text, "halloween"), 1, 0),
    valentine_mention = if_else(str_detect(text, "valentine"), 1, 0),
    christmas_mention = if_else(str_detect(text, "christmas"), 1, 0)
  )
```

Exercise 2

```r
office_episodes <- theoffice |>
  group_by(season, episode, episode_name, imdb_rating, total_votes, air_date) |>
  summarize(
    n_lines = n(),
    lines_jim = sum(character == "Jim") / n_lines,
    lines_pam = sum(character == "Pam") / n_lines,
    lines_michael = sum(character == "Michael") / n_lines,
    lines_dwight = sum(character == "Dwight") / n_lines,
    halloween = if_else(sum(halloween_mention) >= 1, "yes", "no"),
    valentine = if_else(sum(valentine_mention) >= 1, "yes", "no"),
    christmas = if_else(sum(christmas_mention) >= 1, "yes", "no"),
    .groups = "drop"
  ) |>
  select(-n_lines)
office_episodes
```

```
# A tibble: 186 x 13
   season episode episode_name      imdb_rating total_votes air_date   lines_jim
    <int>   <int> <chr>                   <dbl>       <int> <date>         <dbl>
 1      1       1 Pilot                     7.6        3706 2005-03-24    0.157
 2      1       2 Diversity Day             8.3        3566 2005-03-29    0.123
 3      1       3 Health Care               7.9        2983 2005-04-05    0.172
 4      1       4 The Alliance              8.1        2886 2005-04-12    0.202
 5      1       5 Basketball                8.4        3179 2005-04-19    0.0913
 6      1       6 Hot Girl                  7.8        2852 2005-04-26    0.159
 7      2       1 The Dundies               8.7        3213 2005-09-20    0.125
 8      2       2 Sexual Harassment         8.2        2736 2005-09-27    0.0565
 9      2       3 Office Olympics           8.4        2742 2005-10-04    0.196
10      2       4 The Fire                  8.4        2713 2005-10-11    0.160
# ... with 176 more rows, and 6 more variables: lines_pam <dbl>,
#   lines_michael <dbl>, lines_dwight <dbl>, halloween <chr>, valentine <chr>,
#   christmas <chr>
```

Exercise 3

```
office_episodes <- office_episodes |>
  mutate(michael = if_else(season > 7, "no", "yes"))
```

…

Exercise 4

```r
dim(office_episodes)
```

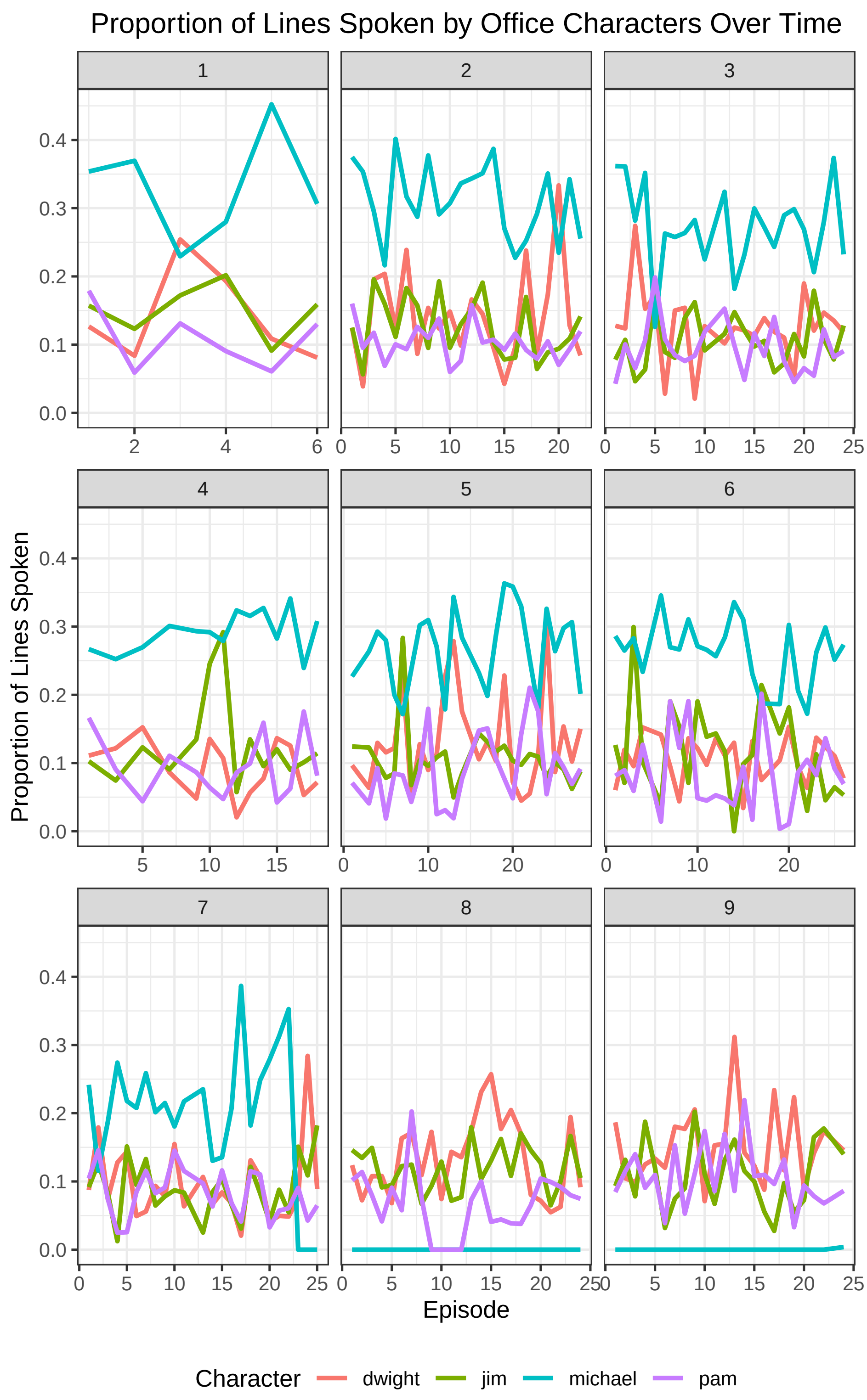
```
[1] 186  14
```

```r
names(office_episodes)
```

```
 [1] "season"        "episode"       "episode_name"  "imdb_rating"  
 [5] "total_votes"   "air_date"      "lines_jim"     "lines_pam"    
 [9] "lines_michael" "lines_dwight"  "halloween"     "valentine"    
[13] "christmas"     "michael"      
```

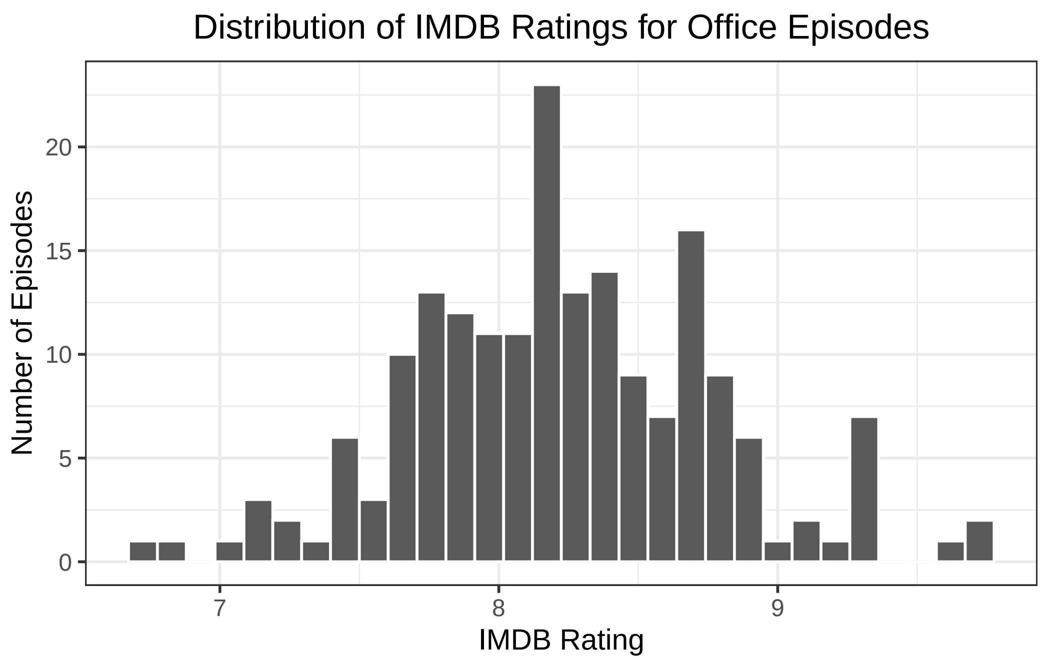
EDA

```r
office_episodes %>%
  pivot_longer(cols = starts_with("lines_"), names_to = "character",
               names_prefix = "lines_", values_to = "proportion") %>%
  ggplot(aes(x = episode, y = proportion, color = character)) +
  geom_line(size = 1) +
  facet_wrap(~season, scales = "free_x") +
  theme_bw() +
  labs(x = "Episode", y = "Proportion of Lines Spoken", color = "Character",
       title = "Proportion of Lines Spoken by Office Characters Over Time") +
  theme(plot.title = element_text(hjust = 0.5), legend.position = "bottom")
```


Proportion of Lines Spoken by Office Characters Over Time
1
2
3
4
5
6
7
8
9
Proportion of Lines Spoken
Episode
Character
dwight
jim
michael
pam

```r
ggplot(office_episodes, aes(x = imdb_rating)) +
  geom_histogram(color = "white") +
  theme_bw() +
  labs(x = "IMDB Rating", y = "Number of Episodes",
       title = "Distribution of IMDB Ratings for Office Episodes") +
  theme(plot.title = element_text(hjust = 0.5))
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```r
fivenum(office_episodes$imdb_rating)
```

```
[1] 6.7 7.9 8.2 8.6 9.7
```

```r
min(office_episodes$imdb_rating)
```

```
[1] 6.7
```

```r
max(office_episodes$imdb_rating)
```

```
[1] 9.7
```

```r
mean(office_episodes$imdb_rating)
```

```
[1] 8.250538
```

```r
sd(office_episodes$imdb_rating)
```

```
[1] 0.5351683
```

```r
office_episodes %>%
  mutate(halloween = if_else(halloween == "no", 0, 1),
         valentine = if_else(valentine == "no", 0, 1),
         christmas = if_else(christmas == "no", 0, 1)) %>%
  summarize(halloween_prop = mean(halloween),
            valentine_prop = mean(valentine),
            christmas_prop = mean(christmas))
```

```
# A tibble: 1 x 3
  halloween_prop valentine_prop christmas_prop
           <dbl>          <dbl>          <dbl>
1         0.0538         0.0376          0.183
```

…

Exercise 5

```
set.seed(123)
office_split <- initial_split(office_episodes)
office_train <- training(office_split)
office_test <- testing(office_split)
```

…

Exercise 6

```r
office_spec <- linear_reg() |>
  set_engine("lm")

office_spec
```

```
Linear Regression Model Specification (regression)

Computational engine: lm
```

Exercise 7

```r
office_rec <- recipe(imdb_rating ~ ., data = office_train) |>
  update_role(episode_name, new_role = "ID") |>
  step_rm(air_date, season) |>
  step_dummy(all_nominal_predictors()) |>
  step_zv(all_predictors())
office_rec
```

```
Recipe

Inputs:

      role #variables
        ID          1
   outcome          1
 predictor         12

Operations:

Delete terms air_date, season
Dummy variables from all_nominal_predictors()
Zero variance filter on all_predictors()
```

Exercise 8

```
office_wflow <- workflow() |>
  add_model(office_spec) |>
  add_recipe(office_rec)
```

Exercise 9

```r
office_fit <- office_wflow |>
  fit(data = office_train)

tidy(office_fit) %>%
  kable(digits = 3)
```

term	estimate	std.error	statistic	p.value
(Intercept)	7.010	0.158	44.368	0.000
episode	0.006	0.004	1.518	0.131
total_votes	0.000	0.000	10.329	0.000
lines_jim	0.273	0.621	0.439	0.662
lines_pam	0.130	0.668	0.194	0.846
lines_michael	-0.279	0.538	-0.519	0.605
lines_dwight	0.392	0.496	0.790	0.431
halloween_yes	-0.177	0.126	-1.403	0.163
valentine_yes	-0.076	0.154	-0.497	0.620
christmas_yes	0.199	0.077	2.584	0.011
michael_yes	0.418	0.156	2.683	0.008

If an episode mentions the word "halloween", we would expect it to have an IMDB rating that is 0.177 points lower, on average, than episodes that do not mention the word "halloween", holding all other predictors constant.

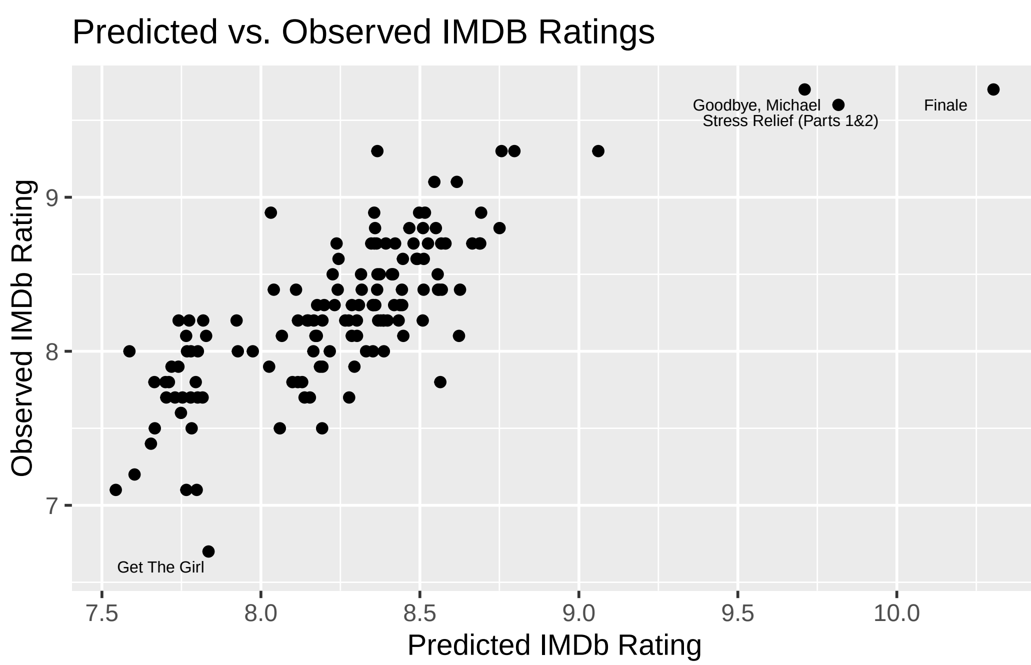
If an episode mentions the word "christmas", we would expect it to have an IMDB rating that is 0.199 points higher, on average, than episodes that do not mention the word "christmas", holding all other predictors constant.

Exercise 10

```
office_train_pred <- predict(office_fit, office_train) |>
  bind_cols(office_train)

unusual_office <- office_train_pred |>
  filter(imdb_rating < 7 | imdb_rating > 9.5)

office_train_pred |>
  ggplot(aes(x = .pred, y = imdb_rating, label = episode_name)) +
  geom_point() +
  geom_text(data = unusual_office, size = 2, nudge_x = -0.15, nudge_y = -0.1) +
  labs(x = "Predicted IMDb Rating",
       y = "Observed IMDb Rating",
       title = "Predicted vs. Observed IMDB Ratings")
```



Exercise 11

```
rsq(office_train_pred, truth = imdb_rating, estimate = .pred)
```

```
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rsq     standard       0.614
```

```
rmse(office_train_pred, truth = imdb_rating, estimate = .pred)
```

```
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard       0.318
```

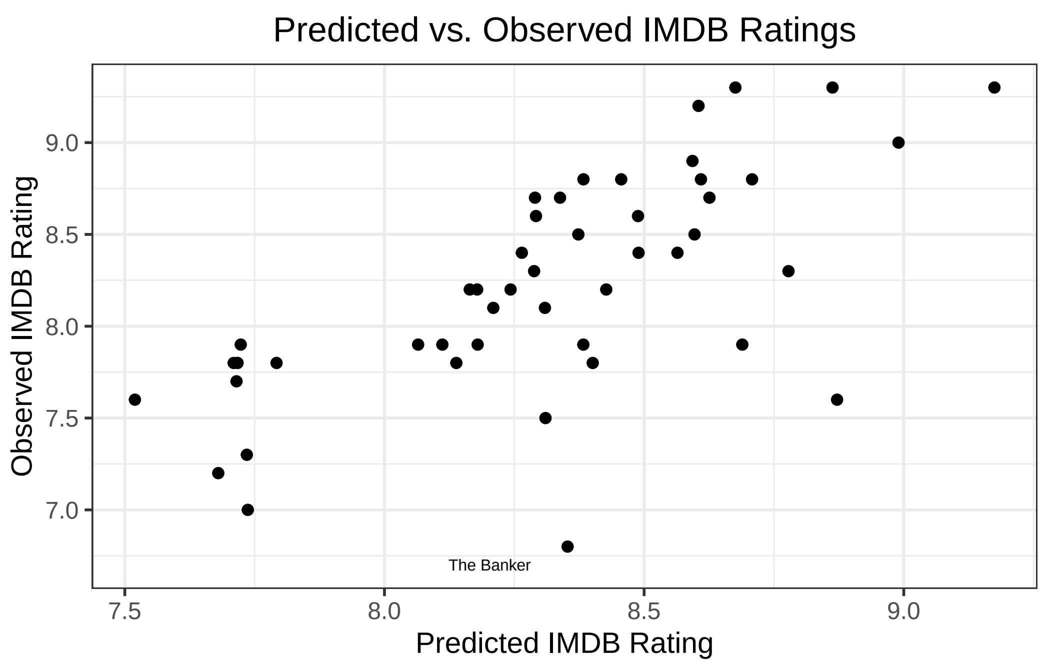
- The R^2 of 0.614 tells us that, based on our training data, 61.4% of the variability in IMDB rating can be explained by the predictors in our model.
- The RMSE of 0.3176 tells us that, based on our training data, on average, the error in predicted IMDB rating is 0.318 points.

Exercise 12

```r
office_test_pred <- predict(office_fit, office_test) %>%
  bind_cols(office_test)

unusual_office_test <- office_test_pred %>%
  filter(imdb_rating < 7 | imdb_rating > 9.5)

office_test_pred |>
  ggplot(aes(x = .pred, y = imdb_rating, label = episode_name)) +
  geom_point() +
  geom_text(data = unusual_office_test,
            size = 2, nudge_x = -0.15, nudge_y = -0.1) +
  labs(x = "Predicted IMDB Rating",
       y = "Observed IMDB Rating",
       title = "Predicted vs. Observed IMDB Ratings") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



...

Exercise 13

Based on the visualization created in exercise 12, we would expect the R^2 for this model to be lower for predictions on the testing data compared to the training data. This is because the model was created on the training data; thus, it is designed to explain as much of the variance in the training data as possible. However, because the testing data differs somewhat from the training data, the model will not be able to explain as much of the variance in the testing data. Similarly, we would expect the RMSE for this model to be higher for predictions on the testing data compared to the training data. The model was designed to minimize error on the training data; however, because the testing data differ, we would expect to see more error in this model's predictions for the testing data.

...

Exercise 14

```r
rsq(office_test_pred, truth = imdb_rating, estimate = .pred)
```

```
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rsq     standard       0.461
```

```r
rmse(office_test_pred, truth = imdb_rating, estimate = .pred)
```

```
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard       0.447
```

This answer presumes that there was an error in the lab's writing, and we should indeed be calculating R^2 and RMSE for predictions on the testing, not training, data. We calcluated these values on the testing data in exercise 11, and this flows more logically from exercise 13.

These data confirm that our intuition was correct. The R^2 value for predictions from this model on the testing data was 0.461, suggesting that 46.1% of the variance in IMDB ratings in the testing data can be explained by the predictors in this model - notably less than the 61.4% of variance explained by this model on the training data. Similarly, the RMSE value for predictions from this model on the testing data was 0.447, suggesting that on average, the error in predicted IMDB rating in our testing data is 0.447 points - notably more than our average error of 0.318 points in predicted IMDB rating for our training data.

…