Exercise 13

Marc Dotson

Return to soup\_data and the models from the previous exercise.

1. Split the data with 0.90 of the data in the training data *in order* using initial\_time\_split().
2. Fit the models again on the training data.
3. Compute the RMSE using the testing data.
4. Identify the best-fitting model based on , Adjusted , and RMSE. Is it the same? Why or why not?
5. Render the Quarto document into Word and upload to Canvas.

**Five points total, one point each for:**

* **Splitting the data with 0.90 of the data in the training data *in order* using initial\_time\_split().**
* **Fitting all four models again using the training data.**
* **Computing the RMSE using the testing data for each model.**
* **Identifying the best-fitting model and discussing if they are the same across metrics and considering why or why not.**
* **One point for submitting a rendered Word document.**

## Split the Data

Let’s load the packages we’ll need, import the data, and split the data as specified.

# Load packages.  
library(tidyverse)

── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
✔ dplyr 1.1.4 ✔ readr 2.1.4  
✔ forcats 1.0.0 ✔ stringr 1.5.1  
✔ ggplot2 3.4.4 ✔ tibble 3.2.1  
✔ lubridate 1.9.3 ✔ tidyr 1.3.0  
✔ purrr 1.0.2   
── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
✖ dplyr::filter() masks stats::filter()  
✖ dplyr::lag() masks stats::lag()  
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(tidymodels)

── Attaching packages ────────────────────────────────────── tidymodels 1.1.1 ──  
✔ broom 1.0.5 ✔ rsample 1.2.0  
✔ dials 1.2.0 ✔ tune 1.1.2  
✔ infer 1.0.5 ✔ workflows 1.1.3  
✔ modeldata 1.2.0 ✔ workflowsets 1.0.1  
✔ parsnip 1.1.1 ✔ yardstick 1.2.0  
✔ recipes 1.0.9   
── Conflicts ───────────────────────────────────────── tidymodels\_conflicts() ──  
✖ scales::discard() masks purrr::discard()  
✖ dplyr::filter() masks stats::filter()  
✖ recipes::fixed() masks stringr::fixed()  
✖ dplyr::lag() masks stats::lag()  
✖ yardstick::spec() masks readr::spec()  
✖ recipes::step() masks stats::step()  
• Use tidymodels\_prefer() to resolve common conflicts.

# Import and filter data.  
soup\_data <- read\_csv(here::here("Data", "soup\_data.csv")) |>   
 filter(Retailer\_Trade\_Areas == "WEST CENSUS TA", Brand\_High == "CAMPBELL'S")

Rows: 3192 Columns: 66  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (5): Four\_Weeks\_Periods, Retailer\_Trade\_Areas, Category, Sub\_Category, ...  
dbl (61): Sales, Base\_Sales, Incr\_Sales, Units, Base\_Units, Incr\_Units, Perc...  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

# Split the data.  
soup\_split <- initial\_time\_split(soup\_data, prop = 0.90)

## Fit the Models

Now let’s refit the models we ran previously.

# Full model.  
fit\_01 <- linear\_reg() |>   
 set\_engine(engine = "lm") |>   
 fit(  
 Sales ~ Any\_Disp\_Spend + Any\_Feat\_Spend + Any\_Price\_Decr\_Spend,   
 data = training(soup\_split)  
 )  
  
# Model without display spend.  
fit\_02 <- linear\_reg() |>   
 set\_engine(engine = "lm") |>   
 fit(  
 Sales ~ Any\_Feat\_Spend + Any\_Price\_Decr\_Spend,   
 data = training(soup\_split)  
 )  
  
# Model without feature spend.  
fit\_03 <- linear\_reg() |>   
 set\_engine(engine = "lm") |>   
 fit(  
 Sales ~ Any\_Disp\_Spend + Any\_Price\_Decr\_Spend,   
 data = training(soup\_split)  
 )  
  
# Model without price decrease spend.  
fit\_04 <- linear\_reg() |>   
 set\_engine(engine = "lm") |>   
 fit(  
 Sales ~ Any\_Disp\_Spend + Any\_Feat\_Spend,   
 data = training(soup\_split)  
 )

## Overall Model Fit

Now let’s compute and compare RMSE, as well as R-squared and the Adjusted R-squared.

# Compute RMSE.  
rmse\_01 <- fit\_01 |>   
 predict(new\_data = testing(soup\_split)) |>  
 bind\_cols(testing(soup\_split)) |>  
 rmse(truth = Sales, estimate = .pred)  
  
rmse\_02 <- fit\_02 |>   
 predict(new\_data = testing(soup\_split)) |>  
 bind\_cols(testing(soup\_split)) |>  
 rmse(truth = Sales, estimate = .pred)  
  
rmse\_03 <- fit\_03 |>   
 predict(new\_data = testing(soup\_split)) |>  
 bind\_cols(testing(soup\_split)) |>  
 rmse(truth = Sales, estimate = .pred)  
  
rmse\_04 <- fit\_04 |>   
 predict(new\_data = testing(soup\_split)) |>  
 bind\_cols(testing(soup\_split)) |>  
 rmse(truth = Sales, estimate = .pred)  
  
# Compare RMSEs.  
tibble(  
 model = c(  
 "Full model",   
 "Model without display spend",   
 "Model without feature spend",   
 "Model without price decrease spend"  
 )  
) |>   
 bind\_cols(  
 bind\_rows(  
 rmse\_01,  
 rmse\_02,  
 rmse\_03,  
 rmse\_04  
 )  
 ) |>   
 arrange(.estimate)

# A tibble: 4 × 4  
 model .metric .estimator .estimate  
 <chr> <chr> <chr> <dbl>  
1 Full model rmse standard 82203.  
2 Model without display spend rmse standard 83522.  
3 Model without price decrease spend rmse standard 85459.  
4 Model without feature spend rmse standard 86023.

Based on RMSE, the best-fitting model is the model with all three explanatory variables – the “full model.”

# Model comparison.  
tibble(  
 model = c(  
 "Full model",   
 "Model without display spend",   
 "Model without feature spend",   
 "Model without price decrease spend"  
 )  
) |>   
 bind\_cols(  
 bind\_rows(  
 glance(fit\_01),   
 glance(fit\_02),   
 glance(fit\_03),   
 glance(fit\_04)  
 )  
 ) |>   
 arrange(desc(r.squared))

# A tibble: 4 × 13  
 model r.squared adj.r.squared sigma statistic p.value df logLik AIC  
 <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 Full mod… 0.907 0.900 56448. 136. 1.22e-21 3 -566. 1143.  
2 Model wi… 0.890 0.885 60464. 174. 2.36e-21 2 -570. 1148.  
3 Model wi… 0.878 0.872 63752. 155. 2.30e-20 2 -573. 1153.  
4 Model wi… 0.863 0.857 67419. 136. 2.54e-19 2 -575. 1158.  
# ℹ 4 more variables: BIC <dbl>, deviance <dbl>, df.residual <int>, nobs <int>

According to R-squared and Adjusted R-squared, the “full model” that includes all three of the explanatory variables fits best.

The fact that RMSE identifies the same best-fitting model as R-squared and Adjusted R-squared suggests that the “full model” isn’t overfitting the data. That might not be surprising since the “full model” only has three explanatory variables.