# lab-jupyter-linear-models-refinments

April 15, 2023

Refine the Baseline Regression Models

Estimated time needed: **120** minutes

## 0.1 Lab Overview:

Now you have built a baseline regression model with some relatively good RMSE and R-squared reported values. However, we could still improve it by using methods like adding polynomial and interaction terms, regularization, and so on.

In this lab, you will be asked to continue using `tidymodels` to improve the performance of baseline model:

- **TASK: Add polynomial terms**
- **TASK: Add interactions terms**
- **TASK: Add regularizations terms**
- **TASK: Experiment to search for improved models**

Let's start!

First install and import necessary libraries

[1]:
```
library("tidymodels")
library("tidyverse")
library("stringr")
```

```
 Attaching packages                     tidymodels 1.0.0
broom        1.0.4      recipes      1.0.5
dials        1.1.0      rsample      1.1.1
dplyr        1.1.0      tibble       3.2.0
ggplot2      3.4.1      tidyr        1.3.0
infer        1.0.4      tune         1.0.1
modeldata    1.1.0      workflows    1.1.3
parsnip      1.0.4      workflowsets 1.0.0
purrr        1.0.1      yardstick    1.1.0
 Conflicts                         tidymodels_conflicts()
purrr::discard() masks scales::discard()
dplyr::filter()  masks stats::filter()
dplyr::lag()     masks stats::lag()
recipes::step()  masks stats::step()
• Use tidymodels_prefer() to resolve common conflicts.
```

```
 Attaching packages                                   tidyverse 1.3.0
readr   1.3.1      forcats 0.5.0
stringr 1.5.0
 Conflicts                                tidyverse_conflicts()
readr::col_factor() masks scales::col_factor()
purrr::discard()    masks scales::discard()
dplyr::filter()     masks stats::filter()
stringr::fixed()    masks recipes::fixed()
dplyr::lag()        masks stats::lag()
readr::spec()       masks yardstick::spec()
```

The processed Seoul bike sharing dataset `seoul_bike_sharing_converted_normalized.csv`, includes the converted indicator variables, and the numerical variables have been normalized. Let's read it as a dataframe first:

[2]:
```
# Dataset URL
dataset_url <- "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.
 ↪cloud/IBMDeveloperSkillsNetwork-RP0321EN-SkillsNetwork/labs/datasets/
 ↪seoul_bike_sharing_converted_normalized.csv"
bike_sharing_df <- read_csv(dataset_url)
spec(bike_sharing_df)
```

```
Parsed with column specification:
cols(
  .default = col_double(),
  DATE = col_character(),
  FUNCTIONING_DAY = col_character()
)
See spec(…) for full column specifications.
```

```
cols(
  DATE = col_character(),
  RENTED_BIKE_COUNT = col_double(),
  TEMPERATURE = col_double(),
  HUMIDITY = col_double(),
  WIND_SPEED = col_double(),
  VISIBILITY = col_double(),
  DEW_POINT_TEMPERATURE = col_double(),
  SOLAR_RADIATION = col_double(),
  RAINFALL = col_double(),
  SNOWFALL = col_double(),
  FUNCTIONING_DAY = col_character(),
  `0` = col_double(),
  `1` = col_double(),
  `10` = col_double(),
  `11` = col_double(),
  `12` = col_double(),
  `13` = col_double(),
  `14` = col_double(),
```

```
  `15` = col_double(),
  `16` = col_double(),
  `17` = col_double(),
  `18` = col_double(),
  `19` = col_double(),
  `2` = col_double(),
  `20` = col_double(),
  `21` = col_double(),
  `22` = col_double(),
  `23` = col_double(),
  `3` = col_double(),
  `4` = col_double(),
  `5` = col_double(),
  `6` = col_double(),
  `7` = col_double(),
  `8` = col_double(),
  `9` = col_double(),
  AUTUMN = col_double(),
  SPRING = col_double(),
  SUMMER = col_double(),
  WINTER = col_double(),
  HOLIDAY = col_double(),
  NO_HOLIDAY = col_double()
)
```

We won't be using the `DATE` column, because 'as is', it basically acts like an data entry index. (However, given more time, we could use the `DATE` colum to create a 'day of week' or 'isWeekend' column, which we might expect has an affect on preferred bike rental times.) We also do not need the `FUNCTIONAL DAY` column because it only has one distinct value remaining (`YES`) after missing value processing.

[3]:
```
bike_sharing_df <- bike_sharing_df %>%
                   select(-DATE, -FUNCTIONING_DAY)
```

Define a linear regression model specification.

[4]:
```
lm_spec <- linear_reg() %>%
  set_engine("lm") %>%
  set_mode("regression")
```

Split the data into training and testing datasets.

[5]:
```
set.seed(1234)
data_split <- initial_split(bike_sharing_df, prop = 4/5)
train_data <- training(data_split)
test_data <- testing(data_split)
```

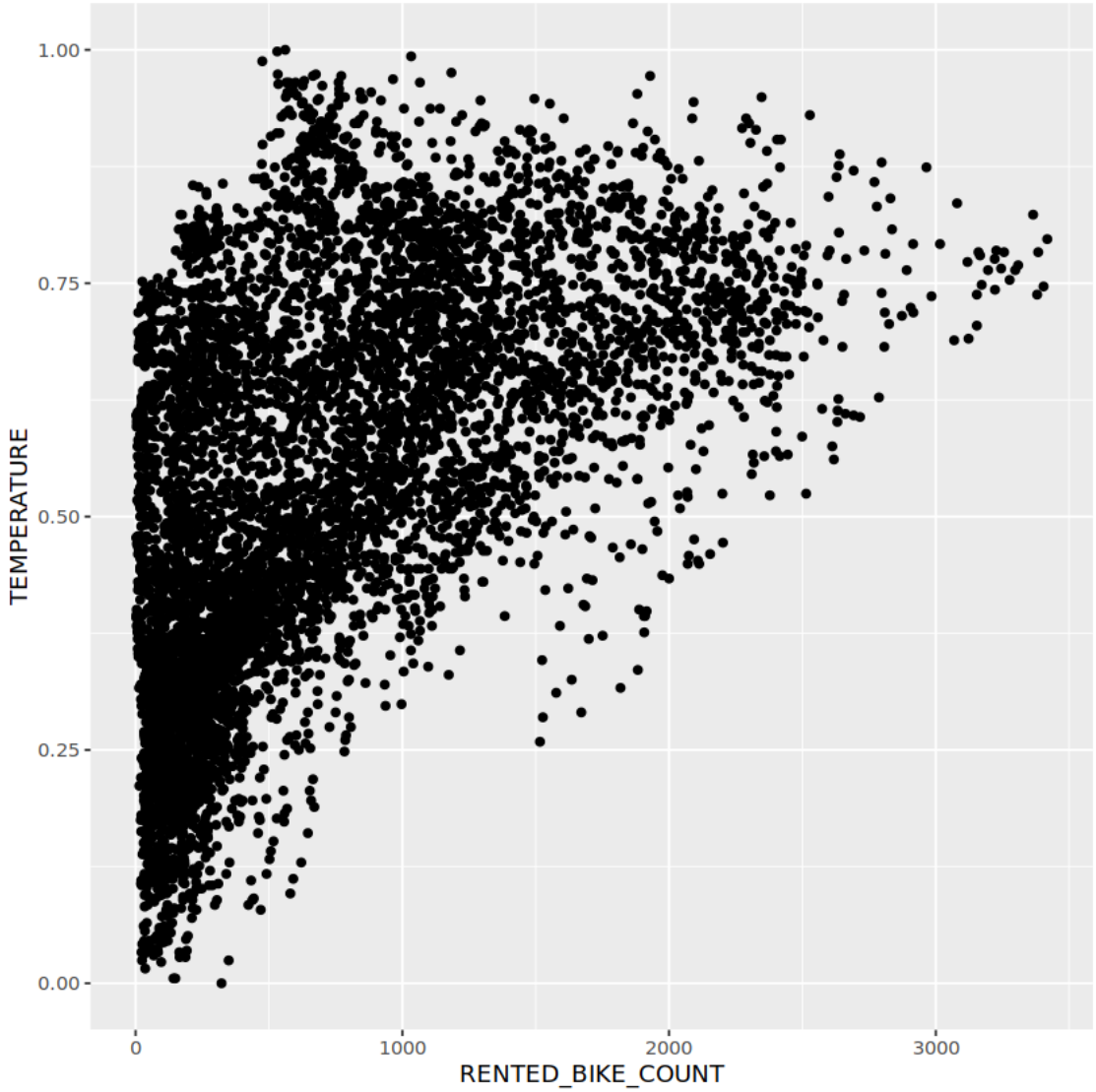
Now we are ready to refine the previous baseline regression model.

# 1 TASK: Add polynomial terms

Linear regression models are the most suitable models to capture the linear correlations among variables. However, in real world data, many relationships may be non-linear.

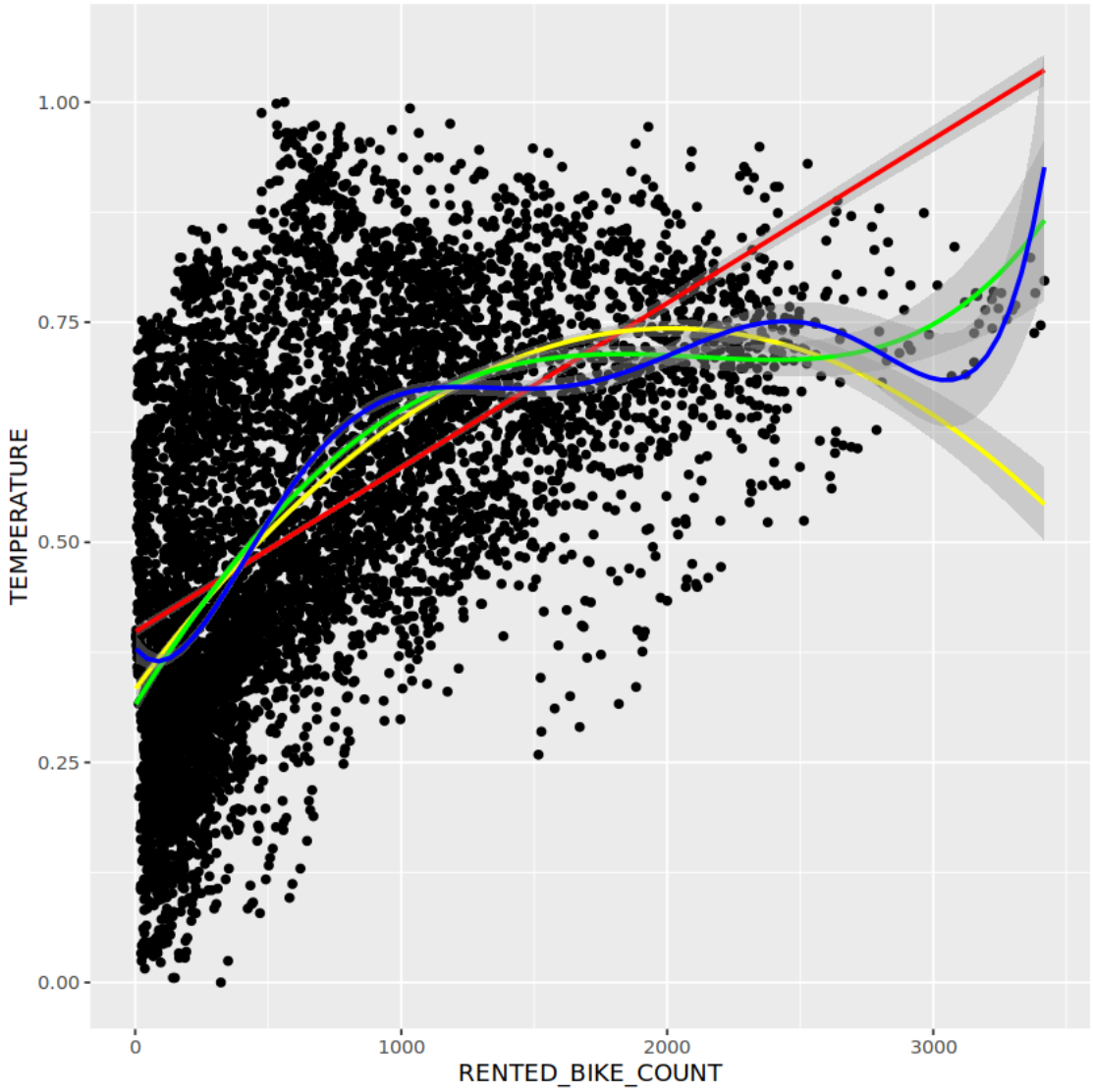
For example, the correlation between `RENTED_BIKE_COUNT` and `TEMPERATURE` does not look like linear:

[6]:
```
ggplot(data = train_data, aes(RENTED_BIKE_COUNT, TEMPERATURE)) +
    geom_point()
```




One solution to handle such nonlinearity is using polynomial regression by adding polynomial terms. As shown before, higher order polynomials are better than the first order polynomial.

[7]:
```
# Plot the higher order polynomial fits
ggplot(data=train_data, aes(RENTED_BIKE_COUNT, TEMPERATURE)) +
    geom_point() +
    geom_smooth(method = "lm", formula = y ~ x, color="red") +
    geom_smooth(method = "lm", formula = y ~ poly(x, 2), color="yellow") +
    geom_smooth(method = "lm", formula = y ~ poly(x, 4), color="green") +
    geom_smooth(method = "lm", formula = y ~ poly(x, 6), color="blue")
```




OK, let's add some higher order polynomials of important variables to the regression models

*TODO:* Fit a linear regression model `lm_poly` with higher order polynomial terms on the important variables (larger coefficients) found in the baseline model

[9]:
```
# Fit a linear model with higher order polynomial on some important variables

# #HINT: Use ploy function to build polynomial terms, lm_poly <-␣
 ↪RENTED_BIKE_COUNT ~ poly(TEMPERATURE, 6) + poly(HUMIDITY, 4) .....
lm_poly <- RENTED_BIKE_COUNT ~ poly(TEMPERATURE, 6) + poly(HUMIDITY, 4)+␣
 ↪poly(WIND_SPEED, 4)+ poly(VISIBILITY, 3)+
                              poly(DEW_POINT_TEMPERATURE,6)+␣
 ↪poly(SOLAR_RADIATION, 5) + poly(RAINFALL, 6)+ poly(SNOWFALL, 4) +
                              `0` + `1` + `10` + `11` + `12` + `13` + `14` +␣
 ↪`15` + `16` + `17` + `18` + `19` + `2` + `20` +
                              `21` + `22` + `23` + `3` + `4` + `5` + `6` + `7`␣
 ↪+ `8`+ `9` + AUTUMN+ SPRING + SUMMER + WINTER + HOLIDAY + NO_HOLIDAY

train_fit <- lm_spec %>%
  fit(lm_poly,  data=train_data)
```

[10]:
```
# Print model summary

# summary(lm_poly$fit)
summary(train_fit$fit)
```

```
Call:
stats::lm(formula = RENTED_BIKE_COUNT ~ poly(TEMPERATURE, 6) +
    poly(HUMIDITY, 4) + poly(WIND_SPEED, 4) + poly(VISIBILITY,
    3) + poly(DEW_POINT_TEMPERATURE, 6) + poly(SOLAR_RADIATION,
    5) + poly(RAINFALL, 6) + poly(SNOWFALL, 4) + `0` + `1` +
    `10` + `11` + `12` + `13` + `14` + `15` + `16` + `17` + `18` +
    `19` + `2` + `20` + `21` + `22` + `23` + `3` + `4` + `5` +
    `6` + `7` + `8` + `9` + AUTUMN + SPRING + SUMMER + WINTER +
    HOLIDAY + NO_HOLIDAY, data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-1740.41  -181.03     5.47   166.41  1273.15

Coefficients: (3 not defined because of singularities)
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                      268.266     25.716  10.432  < 2e-16 ***
poly(TEMPERATURE, 6)1          38602.565   4229.867   9.126  < 2e-16 ***
poly(TEMPERATURE, 6)2           9664.062    709.405  13.623  < 2e-16 ***
poly(TEMPERATURE, 6)3         -10741.230    452.201 -23.753  < 2e-16 ***
poly(TEMPERATURE, 6)4          -7261.710    429.991 -16.888  < 2e-16 ***
poly(TEMPERATURE, 6)5             28.176    373.887   0.075 0.939930
poly(TEMPERATURE, 6)6           1929.759    359.981   5.361 8.57e-08 ***
poly(HUMIDITY, 4)1              -635.410   2020.776  -0.314 0.753198
poly(HUMIDITY, 4)2             -3445.652    511.522  -6.736 1.76e-11 ***
```

```
poly(HUMIDITY, 4)3                -2318.015    371.292  -6.243 4.55e-10 ***
poly(HUMIDITY, 4)4                -1137.423    427.066  -2.663 0.007755 **
poly(WIND_SPEED, 4)1               -434.552    376.046  -1.156 0.247894
poly(WIND_SPEED, 4)2              -1196.774    323.654  -3.698 0.000219 ***
poly(WIND_SPEED, 4)3               -741.600    318.185  -2.331 0.019798 *
poly(WIND_SPEED, 4)4                457.772    315.694   1.450 0.147092
poly(VISIBILITY, 3)1              -1455.631    438.722  -3.318 0.000912 ***
poly(VISIBILITY, 3)2                449.168    347.159   1.294 0.195765
poly(VISIBILITY, 3)3              -1427.409    326.985  -4.365 1.29e-05 ***
poly(DEW_POINT_TEMPERATURE, 6)1 -22734.324   4969.154  -4.575 4.85e-06 ***
poly(DEW_POINT_TEMPERATURE, 6)2 -14046.254    716.295 -19.610  < 2e-16 ***
poly(DEW_POINT_TEMPERATURE, 6)3  -1833.481    457.249  -4.010 6.14e-05 ***
poly(DEW_POINT_TEMPERATURE, 6)4    964.956    417.751   2.310 0.020925 *
poly(DEW_POINT_TEMPERATURE, 6)5    775.914    370.171   2.096 0.036111 *
poly(DEW_POINT_TEMPERATURE, 6)6    830.193    331.322   2.506 0.012245 *
poly(SOLAR_RADIATION, 5)1        12009.756    918.417  13.077  < 2e-16 ***
poly(SOLAR_RADIATION, 5)2       -11033.039    484.377 -22.778  < 2e-16 ***
poly(SOLAR_RADIATION, 5)3         5766.435    384.602  14.993  < 2e-16 ***
poly(SOLAR_RADIATION, 5)4        -3123.697    348.673  -8.959  < 2e-16 ***
poly(SOLAR_RADIATION, 5)5         2525.991    336.015   7.517 6.31e-14 ***
poly(RAINFALL, 6)1               -3905.280    369.401 -10.572  < 2e-16 ***
poly(RAINFALL, 6)2                2660.767    340.569   7.813 6.45e-15 ***
poly(RAINFALL, 6)3               -1524.905    331.201  -4.604 4.22e-06 ***
poly(RAINFALL, 6)4                2746.691    322.789   8.509  < 2e-16 ***
poly(RAINFALL, 6)5               -2155.922    319.000  -6.758 1.51e-11 ***
poly(RAINFALL, 6)6                1742.463    316.481   5.506 3.81e-08 ***
poly(SNOWFALL, 4)1                -500.801    350.361  -1.429 0.152939
poly(SNOWFALL, 4)2                 777.141    329.531   2.358 0.018386 *
poly(SNOWFALL, 4)3                -276.317    334.489  -0.826 0.408783
poly(SNOWFALL, 4)4                -130.617    332.264  -0.393 0.694248
`0`                                321.528     32.344   9.941  < 2e-16 ***
`1`                                205.454     32.406   6.340 2.45e-10 ***
`10`                              -209.025     26.363  -7.929 2.58e-15 ***
`11`                              -158.828     27.339  -5.810 6.55e-09 ***
`12`                               -56.458     28.253  -1.998 0.045724 *
`13`                               -46.091     28.441  -1.621 0.105157
`14`                               -34.976     27.918  -1.253 0.210318
`15`                                 6.625     27.643   0.240 0.810596
`16`                               118.001     27.527   4.287 1.84e-05 ***
`17`                               401.130     27.289  14.700  < 2e-16 ***
`18`                               919.057     28.280  32.498  < 2e-16 ***
`19`                               741.880     30.010  24.721  < 2e-16 ***
`2`                                105.017     32.031   3.279 0.001049 **
`20`                               796.528     32.108  24.808  < 2e-16 ***
`21`                               820.118     32.791  25.010  < 2e-16 ***
`22`                               689.736     32.585  21.167  < 2e-16 ***
`23`                               462.290     32.495  14.227  < 2e-16 ***
`3`                                 31.572     32.148   0.982 0.326097
```

```
`4`                                -47.965     31.901  -1.504 0.132735
`5`                                -20.809     32.081  -0.649 0.516594
`6`                                130.379     31.658   4.118 3.86e-05 ***
`7`                                299.601     29.297  10.226  < 2e-16 ***
`8`                                519.212     26.796  19.377  < 2e-16 ***
`9`                                     NA         NA      NA       NA
AUTUMN                             371.833     19.516  19.053  < 2e-16 ***
SPRING                             208.836     18.876  11.064  < 2e-16 ***
SUMMER                             300.650     24.712  12.166  < 2e-16 ***
WINTER                                  NA         NA      NA       NA
HOLIDAY                            -84.725     18.030  -4.699 2.66e-06 ***
NO_HOLIDAY                              NA         NA      NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 311.5 on 6706 degrees of freedom
Multiple R-squared:  0.7683,        Adjusted R-squared:  0.766
F-statistic: 342.1 on 65 and 6706 DF,  p-value: < 2.2e-16
```

*TODO:* Make predictions on test dataset using the `lm_poly` models

[11]:
```
# Use predict() function to generate test results for `lm_poly`
test_results <- train_fit %>%
predict(new_data = test_data) %>%
mutate(truth = test_data$RENTED_BIKE_COUNT)
head(test_results)
```

```
Warning message in predict.lm(object = object$fit, newdata = new_data, type =
"response"):
"prediction from a rank-deficient fit may be misleading"
```

A tibble: 6 × 2

| .pred <dbl> | truth <dbl> |
|---|---|
| 127.102562 | 254 |
| -3.939999 | 204 |
| -232.061186 | 100 |
| 80.637429 | 460 |
| 315.503696 | 930 |
| 539.709057 | 398 |

Another minor improvement we could do here is to convert all negative prediction results to zero, because we can not have negative rented bike counts

[12]:
```
# e.g., test_results[test_results<0] <- 0
test_results[test_results<0] <- 0
```

Now, calculate R-squared and RMSE for the test results generated by `lm_ploy` model

[13]:
```
# Calculate R-squared and RMSE from the test results
rsq_1<-rsq(test_results, truth = truth, estimate = .pred)

rmse_1<-rmse(test_results, truth = truth, estimate = .pred)


model_1_results<-c( rsq_1, rmse_1)
model_1_results
```

**$.metric** 'rsq'

**$.estimator** 'standard'

**$.estimate** 0.771285774483661

**$.metric** 'rmse'

**$.estimator** 'standard'

**$.estimate** 305.848811787698

If you include all variables, and additionally include some of the more important ones as higher order poly terms, then you should notice improved `R-squared` and `RMSE` values.

## 2 TASK: Add interaction terms

In real-world scenarios, in addition to non-linear relationships between response variables and predictor variables, you may also encounter relationships among variables called `interaction effects`.

For example, the effect of predictor variable `TEMPERATURE` on `RENTED_BIKE_COUNT` may also depend on other variables such as `HUMIDITY`, `RAINFALL`, or both (they `interact`) and the effect of `SEASON` on `RENTED_BIKE_COUNT` may also depend on `HOLIDAY`, `HOUR`, or both.

To capture such interaction effects, we could add some interaction terms such as `RAINFALL*HUMIDITY` to the regression model, similar to what we did with polynominal terms. In this task, you will explore and conduct some experiments to search for interaction terms which will improve the model performance.

*TODO:* Try adding some interaction terms to the previous polynomial models.

[14]:
```
# Add interaction terms to the poly regression built in previous step

# HINT: You could use `*` operator to create interaction terms such as␣
 ↪HUMIDITY*TEMPERATURE and make the formula look like:
# RENTED_BIKE_COUNT ~ RAINFALL*HUMIDITY ...
lm_interactive <- RENTED_BIKE_COUNT ~ `18`*TEMPERATURE*DEW_POINT_TEMPERATURE +␣
 ↪RAINFALL*HUMIDITY*`4` +SOLAR_RADIATION*SNOWFALL+
                                        WIND_SPEED*VISIBILITY +`18`*TEMPERATURE*␣
 ↪+poly(TEMPERATURE, 6) + poly(HUMIDITY, 4)+ poly(WIND_SPEED, 4)+␣
 ↪poly(VISIBILITY, 3)+
```

```
  poly(DEW_POINT_TEMPERATURE,6)+ poly(SOLAR_RADIATION, 5) + poly(RAINFALL, 6)+␣
 ↪poly(SNOWFALL, 4) +
  `0` + `1` + `10` + `11` + `12` + `13` + `14` + `15` + `16` + `17`  + `19` +␣
 ↪`2` + `20` +
  `21` + `22` + `23` + `3`  + `5` + `6` + `7` + `8`+ `9` + AUTUMN+ SPRING +␣
 ↪SUMMER + WINTER + HOLIDAY + NO_HOLIDAY

train_fit_2 <- lm_spec %>%
  fit(lm_interactive,  data=train_data)

train_fit_2
```

```
parsnip model object


Call:
stats::lm(formula = RENTED_BIKE_COUNT ~ `18` * TEMPERATURE *
    DEW_POINT_TEMPERATURE + RAINFALL * HUMIDITY * `4` + SOLAR_RADIATION *
    SNOWFALL + WIND_SPEED * VISIBILITY + `18` * TEMPERATURE *
    +poly(TEMPERATURE, 6) + poly(HUMIDITY, 4) + poly(WIND_SPEED,
    4) + poly(VISIBILITY, 3) + poly(DEW_POINT_TEMPERATURE, 6) +
    poly(SOLAR_RADIATION, 5) + poly(RAINFALL, 6) + poly(SNOWFALL,
    4) + `0` + `1` + `10` + `11` + `12` + `13` + `14` + `15` +
    `16` + `17` + `19` + `2` + `20` + `21` + `22` + `23` + `3` +
    `5` + `6` + `7` + `8` + `9` + AUTUMN + SPRING + SUMMER +
    WINTER + HOLIDAY + NO_HOLIDAY, data = data)

Coefficients:
                          (Intercept)                                   `18`
                              1905.03                                 -54.45
                           TEMPERATURE                  DEW_POINT_TEMPERATURE
                             -1745.35                               -7335.56
                              RAINFALL                               HUMIDITY
                            -11539.95                                 587.40
                                   `4`                        SOLAR_RADIATION
                              -21.80                                  503.07
                             SNOWFALL                             WIND_SPEED
                              -62.54                                 -183.19
                           VISIBILITY                  poly(TEMPERATURE, 6)1
                              -91.16                                      NA
                 poly(TEMPERATURE, 6)2                  poly(TEMPERATURE, 6)3
                            -9084.46                              -10880.02
                 poly(TEMPERATURE, 6)4                  poly(TEMPERATURE, 6)5
                            -7137.86                               -1022.73
                 poly(TEMPERATURE, 6)6                     poly(HUMIDITY, 4)1
                             -432.87                                      NA
                    poly(HUMIDITY, 4)2                     poly(HUMIDITY, 4)3
```

| | |
|---|---|
| -1751.93 | -3850.38 |
| poly(HUMIDITY, 4)4 | poly(WIND_SPEED, 4)1 |
| -897.96 | NA |
| poly(WIND_SPEED, 4)2 | poly(WIND_SPEED, 4)3 |
| -1285.05 | -697.99 |
| poly(WIND_SPEED, 4)4 | poly(VISIBILITY, 3)1 |
| 368.81 | NA |
| poly(VISIBILITY, 3)2 | poly(VISIBILITY, 3)3 |
| 388.98 | -1391.17 |
| poly(DEW_POINT_TEMPERATURE, 6)1 | poly(DEW_POINT_TEMPERATURE, 6)2 |
| NA | -34162.01 |
| poly(DEW_POINT_TEMPERATURE, 6)3 | poly(DEW_POINT_TEMPERATURE, 6)4 |
| -1529.42 | 929.40 |
| poly(DEW_POINT_TEMPERATURE, 6)5 | poly(DEW_POINT_TEMPERATURE, 6)6 |
| 619.69 | 1094.87 |
| poly(SOLAR_RADIATION, 5)1 | poly(SOLAR_RADIATION, 5)2 |
| NA | -8889.49 |
| poly(SOLAR_RADIATION, 5)3 | poly(SOLAR_RADIATION, 5)4 |
| 4583.15 | -2649.33 |
| poly(SOLAR_RADIATION, 5)5 | poly(RAINFALL, 6)1 |
| 2446.66 | NA |
| poly(RAINFALL, 6)2 | poly(RAINFALL, 6)3 |
| 2514.35 | -1366.80 |
| poly(RAINFALL, 6)4 | poly(RAINFALL, 6)5 |
| 2581.49 | -2002.60 |
| poly(RAINFALL, 6)6 | poly(SNOWFALL, 4)1 |
| 1727.17 | NA |
| poly(SNOWFALL, 4)2 | poly(SNOWFALL, 4)3 |
| 619.52 | -212.48 |
| poly(SNOWFALL, 4)4 | `0` |
| -202.87 | 248.42 |
| `1` | `10` |
| 130.65 | -202.49 |
| `11` | `12` |
| -151.21 | -47.55 |
| `13` | `14` |
| -39.29 | -25.55 |
| `15` | `16` |
| 21.26 | 134.71 |
| `17` | `19` |
| 411.22 | 701.09 |
| `2` | `20` |
| 29.55 | 731.26 |
| `21` | `22` |
| 749.77 | 618.49 |
| `23` | `3` |
| 389.94 | -42.22 |
| `5` | `6` |

| | |
|---|---|
| -95.23 | 57.81 |
| `7` | `8` |
| 245.01 | 495.50 |
| `9` | AUTUMN |
| NA | 362.02 |
| SPRING | SUMMER |
| 206.31 | 301.28 |
| WINTER | HOLIDAY |
| NA | -78.59 |
| NO_HOLIDAY | `18`:TEMPERATURE |
| NA | 2578.48 |
| `18`:DEW_POINT_TEMPERATURE | TEMPERATURE:DEW_POINT_TEMPERATURE |
| -210.98 | 9272.40 |
| RAINFALL:HUMIDITY | RAINFALL:`4` |
| 10139.20 | -35728.97 |
| HUMIDITY:`4` | SOLAR_RADIATION:SNOWFALL |
| -157.40 | -876.79 |
| WIND_SPEED:VISIBILITY | `18`:poly(TEMPERATURE, 6)1 |
| 186.09 | NA |
| `18`:poly(TEMPERATURE, 6)2 | `18`:poly(TEMPERATURE, 6)3 |
| 2043.88 | -1605.91 |
| `18`:poly(TEMPERATURE, 6)4 | `18`:poly(TEMPERATURE, 6)5 |
| -2447.71 | 3610.29 |
| `18`:poly(TEMPERATURE, 6)6 | TEMPERATURE:poly(TEMPERATURE, 6)1 |
| -3917.69 | NA |
| TEMPERATURE:poly(TEMPERATURE, 6)2 | TEMPERATURE:poly(TEMPERATURE, 6)3 |
| NA | NA |
| TEMPERATURE:poly(TEMPERATURE, 6)4 | TEMPERATURE:poly(TEMPERATURE, 6)5 |
| NA | NA |
| TEMPERATURE:poly(TEMPERATURE, 6)6 | `18`:TEMPERATURE:DEW_POINT_TEMPERATURE |
| 4565.48 | -1019.59 |
| RAINFALL:HUMIDITY:`4` | `18`:TEMPERATURE:poly(TEMPERATURE, 6)1 |
| 40581.51 | NA |
| `18`:TEMPERATURE:poly(TEMPERATURE, 6)2 | `18`:TEMPERATURE:poly(TEMPERATURE, 6)3 |
| NA | NA |
| `18`:TEMPERATURE:poly(TEMPERATURE, 6)4 | `18`:TEMPERATURE:poly(TEMPERATURE, 6)5 |
| NA | NA |
| `18`:TEMPERATURE:poly(TEMPERATURE, 6)6 | |
| 1560.88 | |

[15]:
```
# Print model summary
summary(train_fit_2$fit)
```

```
Call:
stats::lm(formula = RENTED_BIKE_COUNT ~ `18` * TEMPERATURE *
    DEW_POINT_TEMPERATURE + RAINFALL * HUMIDITY * `4` + SOLAR_RADIATION *
```

```
    SNOWFALL + WIND_SPEED * VISIBILITY + `18` * TEMPERATURE *
    +poly(TEMPERATURE, 6) + poly(HUMIDITY, 4) + poly(WIND_SPEED,
    4) + poly(VISIBILITY, 3) + poly(DEW_POINT_TEMPERATURE, 6) +
    poly(SOLAR_RADIATION, 5) + poly(RAINFALL, 6) + poly(SNOWFALL,
    4) + `0` + `1` + `10` + `11` + `12` + `13` + `14` + `15` +
    `16` + `17` + `19` + `2` + `20` + `21` + `22` + `23` + `3` +
    `5` + `6` + `7` + `8` + `9` + AUTUMN + SPRING + SUMMER +
    WINTER + HOLIDAY + NO_HOLIDAY, data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-2069.64  -175.47     5.46   165.68  1260.71

Coefficients: (22 not defined because of singularities)
                                     Estimate Std. Error t value Pr(>|t|)
(Intercept)                           1905.03     781.06   2.439 0.014753
`18`                                   -54.45     288.23  -0.189 0.850175
TEMPERATURE                          -1745.35    1473.33  -1.185 0.236205
DEW_POINT_TEMPERATURE                -7335.56    2505.56  -2.928 0.003426
RAINFALL                            -11539.95    6391.21  -1.806 0.071026
HUMIDITY                              587.40     304.09   1.932 0.053438
`4`                                   -21.80      79.83  -0.273 0.784778
SOLAR_RADIATION                       503.07      45.49  11.059  < 2e-16
SNOWFALL                              -62.54      96.50  -0.648 0.516956
WIND_SPEED                           -183.19      79.64  -2.300 0.021466
VISIBILITY                            -91.16      26.51  -3.438 0.000589
poly(TEMPERATURE, 6)1                      NA         NA      NA       NA
poly(TEMPERATURE, 6)2                -9084.46    6760.65  -1.344 0.179082
poly(TEMPERATURE, 6)3               -10880.02     488.43 -22.276  < 2e-16
poly(TEMPERATURE, 6)4                -7137.86     433.18 -16.478  < 2e-16
poly(TEMPERATURE, 6)5                -1022.73     496.35  -2.061 0.039388
poly(TEMPERATURE, 6)6                 -432.87     783.10  -0.553 0.580438
poly(HUMIDITY, 4)1                         NA         NA      NA       NA
poly(HUMIDITY, 4)2                   -1751.93     975.10  -1.797 0.072436
poly(HUMIDITY, 4)3                   -3850.38     643.07  -5.987 2.24e-09
poly(HUMIDITY, 4)4                    -897.96     470.62  -1.908 0.056429
poly(WIND_SPEED, 4)1                       NA         NA      NA       NA
poly(WIND_SPEED, 4)2                 -1285.05     324.56  -3.959 7.59e-05
poly(WIND_SPEED, 4)3                  -697.99     314.58  -2.219 0.026532
poly(WIND_SPEED, 4)4                   368.81     311.34   1.185 0.236233
poly(VISIBILITY, 3)1                       NA         NA      NA       NA
poly(VISIBILITY, 3)2                   388.98     344.62   1.129 0.259055
poly(VISIBILITY, 3)3                 -1391.17     322.56  -4.313 1.63e-05
poly(DEW_POINT_TEMPERATURE, 6)1            NA         NA      NA       NA
poly(DEW_POINT_TEMPERATURE, 6)2     -34162.01    8341.90  -4.095 4.27e-05
poly(DEW_POINT_TEMPERATURE, 6)3      -1529.42     493.01  -3.102 0.001929
poly(DEW_POINT_TEMPERATURE, 6)4        929.40     413.88   2.246 0.024764
poly(DEW_POINT_TEMPERATURE, 6)5        619.69     369.59   1.677 0.093654
```

| | | | | |
|---|---|---|---|---|
| poly(DEW_POINT_TEMPERATURE, 6)6 | 1094.87 | 336.83 | 3.250 | 0.001158 |
| poly(SOLAR_RADIATION, 5)1 | NA | NA | NA | NA |
| poly(SOLAR_RADIATION, 5)2 | -8889.49 | 504.53 | -17.620 | < 2e-16 |
| poly(SOLAR_RADIATION, 5)3 | 4583.15 | 389.84 | 11.756 | < 2e-16 |
| poly(SOLAR_RADIATION, 5)4 | -2649.33 | 346.24 | -7.652 | 2.26e-14 |
| poly(SOLAR_RADIATION, 5)5 | 2446.66 | 332.23 | 7.364 | 1.99e-13 |
| poly(RAINFALL, 6)1 | NA | NA | NA | NA |
| poly(RAINFALL, 6)2 | 2514.35 | 387.09 | 6.495 | 8.87e-11 |
| poly(RAINFALL, 6)3 | -1366.80 | 351.87 | -3.884 | 0.000104 |
| poly(RAINFALL, 6)4 | 2581.49 | 322.31 | 8.009 | 1.35e-15 |
| poly(RAINFALL, 6)5 | -2002.60 | 324.13 | -6.178 | 6.86e-10 |
| poly(RAINFALL, 6)6 | 1727.17 | 314.34 | 5.495 | 4.06e-08 |
| poly(SNOWFALL, 4)1 | NA | NA | NA | NA |
| poly(SNOWFALL, 4)2 | 619.52 | 328.76 | 1.884 | 0.059556 |
| poly(SNOWFALL, 4)3 | -212.48 | 329.57 | -0.645 | 0.519131 |
| poly(SNOWFALL, 4)4 | -202.87 | 331.85 | -0.611 | 0.540989 |
| `0` | 248.42 | 32.28 | 7.696 | 1.61e-14 |
| `1` | 130.65 | 32.35 | 4.038 | 5.44e-05 |
| `10` | -202.49 | 25.97 | -7.797 | 7.32e-15 |
| `11` | -151.21 | 26.95 | -5.610 | 2.11e-08 |
| `12` | -47.55 | 27.92 | -1.703 | 0.088585 |
| `13` | -39.29 | 28.10 | -1.398 | 0.162053 |
| `14` | -25.55 | 27.58 | -0.926 | 0.354372 |
| `15` | 21.26 | 27.32 | 0.778 | 0.436488 |
| `16` | 134.71 | 27.22 | 4.948 | 7.66e-07 |
| `17` | 411.22 | 26.93 | 15.272 | < 2e-16 |
| `19` | 701.09 | 29.71 | 23.598 | < 2e-16 |
| `2` | 29.55 | 32.00 | 0.924 | 0.355744 |
| `20` | 731.26 | 31.96 | 22.880 | < 2e-16 |
| `21` | 749.77 | 32.69 | 22.939 | < 2e-16 |
| `22` | 618.49 | 32.50 | 19.032 | < 2e-16 |
| `23` | 389.94 | 32.41 | 12.030 | < 2e-16 |
| `3` | -42.22 | 32.10 | -1.315 | 0.188461 |
| `5` | -95.23 | 32.05 | -2.971 | 0.002974 |
| `6` | 57.81 | 31.61 | 1.829 | 0.067461 |
| `7` | 245.01 | 29.11 | 8.417 | < 2e-16 |
| `8` | 495.50 | 26.44 | 18.741 | < 2e-16 |
| `9` | NA | NA | NA | NA |
| AUTUMN | 362.02 | 19.37 | 18.694 | < 2e-16 |
| SPRING | 206.31 | 18.68 | 11.043 | < 2e-16 |
| SUMMER | 301.28 | 24.59 | 12.250 | < 2e-16 |
| WINTER | NA | NA | NA | NA |
| HOLIDAY | -78.59 | 17.81 | -4.413 | 1.04e-05 |
| NO_HOLIDAY | NA | NA | NA | NA |
| `18`:TEMPERATURE | 2578.48 | 622.33 | 4.143 | 3.47e-05 |
| `18`:DEW_POINT_TEMPERATURE | -210.98 | 725.96 | -0.291 | 0.771352 |
| TEMPERATURE:DEW_POINT_TEMPERATURE | 9272.40 | 3631.12 | 2.554 | 0.010684 |
| RAINFALL:HUMIDITY | 10139.20 | 6518.31 | 1.555 | 0.119876 |

```
RAINFALL:`4`                             -35728.97   37984.48  -0.941 0.346933
HUMIDITY:`4`                               -157.40     105.50  -1.492 0.135784
SOLAR_RADIATION:SNOWFALL                   -876.79     824.86  -1.063 0.287836
WIND_SPEED:VISIBILITY                       186.09      98.37   1.892 0.058582
`18`:poly(TEMPERATURE, 6)1                      NA         NA      NA       NA
`18`:poly(TEMPERATURE, 6)2                2043.88    5299.52   0.386 0.699751
`18`:poly(TEMPERATURE, 6)3               -1605.91    3075.27  -0.522 0.601546
`18`:poly(TEMPERATURE, 6)4               -2447.71    3302.25  -0.741 0.458582
`18`:poly(TEMPERATURE, 6)5                3610.29    2503.74   1.442 0.149361
`18`:poly(TEMPERATURE, 6)6               -3917.69    7334.37  -0.534 0.593252
TEMPERATURE:poly(TEMPERATURE, 6)1               NA         NA      NA       NA
TEMPERATURE:poly(TEMPERATURE, 6)2               NA         NA      NA       NA
TEMPERATURE:poly(TEMPERATURE, 6)3               NA         NA      NA       NA
TEMPERATURE:poly(TEMPERATURE, 6)4               NA         NA      NA       NA
TEMPERATURE:poly(TEMPERATURE, 6)5               NA         NA      NA       NA
TEMPERATURE:poly(TEMPERATURE, 6)6         4565.48    1324.59   3.447 0.000571
`18`:TEMPERATURE:DEW_POINT_TEMPERATURE   -1019.59    1210.00  -0.843 0.399464
RAINFALL:HUMIDITY:`4`                    40581.51   39601.33   1.025 0.305518
`18`:TEMPERATURE:poly(TEMPERATURE, 6)1          NA         NA      NA       NA
`18`:TEMPERATURE:poly(TEMPERATURE, 6)2          NA         NA      NA       NA
`18`:TEMPERATURE:poly(TEMPERATURE, 6)3          NA         NA      NA       NA
`18`:TEMPERATURE:poly(TEMPERATURE, 6)4          NA         NA      NA       NA
`18`:TEMPERATURE:poly(TEMPERATURE, 6)5          NA         NA      NA       NA
`18`:TEMPERATURE:poly(TEMPERATURE, 6)6    1560.88    9727.28   0.160 0.872520

(Intercept)                              *
`18`
TEMPERATURE
DEW_POINT_TEMPERATURE                    **
RAINFALL                                 .
HUMIDITY                                 .
`4`
SOLAR_RADIATION                          ***
SNOWFALL
WIND_SPEED                               *
VISIBILITY                               ***
poly(TEMPERATURE, 6)1
poly(TEMPERATURE, 6)2
poly(TEMPERATURE, 6)3                    ***
poly(TEMPERATURE, 6)4                    ***
poly(TEMPERATURE, 6)5                    *
poly(TEMPERATURE, 6)6
poly(HUMIDITY, 4)1
poly(HUMIDITY, 4)2                       .
poly(HUMIDITY, 4)3                       ***
poly(HUMIDITY, 4)4                       .
poly(WIND_SPEED, 4)1
poly(WIND_SPEED, 4)2                     ***
```

```
poly(WIND_SPEED, 4)3                *
poly(WIND_SPEED, 4)4
poly(VISIBILITY, 3)1
poly(VISIBILITY, 3)2
poly(VISIBILITY, 3)3                ***
poly(DEW_POINT_TEMPERATURE, 6)1
poly(DEW_POINT_TEMPERATURE, 6)2     ***
poly(DEW_POINT_TEMPERATURE, 6)3     **
poly(DEW_POINT_TEMPERATURE, 6)4     *
poly(DEW_POINT_TEMPERATURE, 6)5     .
poly(DEW_POINT_TEMPERATURE, 6)6     **
poly(SOLAR_RADIATION, 5)1
poly(SOLAR_RADIATION, 5)2           ***
poly(SOLAR_RADIATION, 5)3           ***
poly(SOLAR_RADIATION, 5)4           ***
poly(SOLAR_RADIATION, 5)5           ***
poly(RAINFALL, 6)1
poly(RAINFALL, 6)2                  ***
poly(RAINFALL, 6)3                  ***
poly(RAINFALL, 6)4                  ***
poly(RAINFALL, 6)5                  ***
poly(RAINFALL, 6)6                  ***
poly(SNOWFALL, 4)1
poly(SNOWFALL, 4)2                  .
poly(SNOWFALL, 4)3
poly(SNOWFALL, 4)4
`0`                                 ***
`1`                                 ***
`10`                                ***
`11`                                ***
`12`                                .
`13`
`14`
`15`
`16`                                ***
`17`                                ***
`19`                                ***
`2`
`20`                                ***
`21`                                ***
`22`                                ***
`23`                                ***
`3`
`5`                                 **
`6`                                 .
`7`                                 ***
`8`                                 ***
`9`
```

```
AUTUMN                                  ***
SPRING                                  ***
SUMMER                                  ***
WINTER
HOLIDAY                                 ***
NO_HOLIDAY
`18`:TEMPERATURE                        ***
`18`:DEW_POINT_TEMPERATURE
TEMPERATURE:DEW_POINT_TEMPERATURE       *
RAINFALL:HUMIDITY
RAINFALL:`4`
HUMIDITY:`4`
SOLAR_RADIATION:SNOWFALL
WIND_SPEED:VISIBILITY                   .
`18`:poly(TEMPERATURE, 6)1
`18`:poly(TEMPERATURE, 6)2
`18`:poly(TEMPERATURE, 6)3
`18`:poly(TEMPERATURE, 6)4
`18`:poly(TEMPERATURE, 6)5
`18`:poly(TEMPERATURE, 6)6
TEMPERATURE:poly(TEMPERATURE, 6)1
TEMPERATURE:poly(TEMPERATURE, 6)2
TEMPERATURE:poly(TEMPERATURE, 6)3
TEMPERATURE:poly(TEMPERATURE, 6)4
TEMPERATURE:poly(TEMPERATURE, 6)5
TEMPERATURE:poly(TEMPERATURE, 6)6       ***
`18`:TEMPERATURE:DEW_POINT_TEMPERATURE
RAINFALL:HUMIDITY:`4`
`18`:TEMPERATURE:poly(TEMPERATURE, 6)1
`18`:TEMPERATURE:poly(TEMPERATURE, 6)2
`18`:TEMPERATURE:poly(TEMPERATURE, 6)3
`18`:TEMPERATURE:poly(TEMPERATURE, 6)4
`18`:TEMPERATURE:poly(TEMPERATURE, 6)5
`18`:TEMPERATURE:poly(TEMPERATURE, 6)6
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 306.6 on 6689 degrees of freedom
Multiple R-squared:  0.7761,        Adjusted R-squared:  0.7733
F-statistic: 282.7 on 82 and 6689 DF,  p-value: < 2.2e-16
```

[16]:
```r
# Calculate R-squared and RMSE for the new model to see if performance has␣
  ↪improved
test_results_2 <- train_fit_2 %>%
predict(new_data = test_data) %>%
mutate(truth = test_data$RENTED_BIKE_COUNT)
```

```
head(test_results_2)
test_results_2[test_results_2<0] <- 0
```

```
Warning message in predict.lm(object = object$fit, newdata = new_data, type =
"response"):
"prediction from a rank-deficient fit may be misleading"
```

A tibble: 6 × 2

| .pred <dbl> | truth <dbl> |
|---|---|
| 115.35980 | 254 |
| -25.03473 | 204 |
| -246.11749 | 100 |
| 85.92372 | 460 |
| 352.11412 | 930 |
| 538.93776 | 398 |

[17]:
```
rsq_2<-rsq(test_results_2, truth = truth, estimate = .pred)

rmse_2<- rmse(test_results_2, truth = truth, estimate = .pred)

model_2_results<-c(rsq_2, rmse_2)
model_2_results
```

**$.metric** 'rsq'

**$.estimator** 'standard'

**$.estimate** 0.772629482653108

**$.metric** 'rmse'

**$.estimator** 'standard'

**$.estimate** 304.46639380277

# 3 TASK: Add regularization

In previous tasks, you were asked to add polynominal and interaction terms to the model, aiming to capture nonlinearity and interaction effects between the predictor variables. Hopefully, your updated models have better R-squared and RMSE values.

However, adding these terms makes your model more complicated, more difficult to explain, and more likely to suffer from overfitting. To overcome these issues, one solution is to add regularization terms to your models.

When building the baseline model, we used the basic `lm` engine. In this task, you will use a more advanced and generalized `glmnet` engine. It provides a generalized linear model with Lasso, Ridge, and Elastic Net regularizations.

In general, using `glmnet` can enhance your models in the following ways: - Address overfitting issues by shrinking the coefficients - Address predictor variable colinearity by selecting only one

variable from each group of colinear variables (by shrinking their coefficients to zero) - Make your models more interpretable due to simplification (fewer variables in the outcome models)

Now, let's switch our regression engine to `glmnet`

*TODO:* Define a linear regression model specification `glmnet_spec` using `glmnet` engine

[18]:
```
# HINT: Use linear_reg() function with two parameters: penalty and mixture
# - penalty controls the intensity of model regularization
# - mixture controls the tradeoff between L1 and L2 regularizations

# You could manually try different parameter combinations or use grid search to␣
↪find optimal combinations
```

Fit a glmnet model called `lm_glmnet` using the `fit()` function. For the formula part, keep the polynominal and interaction terms you used in the previous task.

[19]:
```
install.packages('glmnet')
library('glmnet')
```

```
Warning message:
"package 'glmnet' is not available (for R version 3.5.1)"Loading required
package: Matrix

Attaching package: 'Matrix'

The following objects are masked from 'package:tidyr':

    expand, pack, unpack

Loading required package: foreach

Attaching package: 'foreach'

The following objects are masked from 'package:purrr':

    accumulate, when

Loaded glmnet 2.0-18
```

[20]:
```
# Fit a glmnet model using the fit() function
glmnet_spec <- linear_reg(penalty = 0.2, mixture = 0.5) %>%
  set_engine("glmnet") %>%
  set_mode("regression")
```

[21]:
```
# Report rsq and rmse of the `lm_glmnet` model
glmnet_fit<-glmnet_spec %>% fit(RENTED_BIKE_COUNT ~␣
↪`18`*TEMPERATURE*DEW_POINT_TEMPERATURE + RAINFALL*HUMIDITY*`4`␣
↪+SOLAR_RADIATION*SNOWFALL +
```

```
                                    WIND_SPEED*VISIBILITY +`18`*TEMPERATURE*␣
↪+poly(TEMPERATURE, 6) + poly(HUMIDITY, 4)+ poly(WIND_SPEED, 4)+␣
↪poly(VISIBILITY, 3)+
                                    poly(DEW_POINT_TEMPERATURE,6)+␣
↪poly(SOLAR_RADIATION, 5) + poly(RAINFALL, 6)+ poly(SNOWFALL, 4) +
                                    `0` + `1` + `10` + `11` + `12` + `13` + `14`␣
↪+ `15` + `16` + `17`  + `19` + `2` + `20` +
                                    `21` + `22` + `23` + `3`  + `5` + `6` + `7` +␣
↪`8`+ `9` + AUTUMN+ SPRING + SUMMER + WINTER + HOLIDAY + NO_HOLIDAY ,
                                  data = train_data )
```

[22]:
```
test_results_glment <- glmnet_fit %>%
predict(new_data = test_data) %>%
mutate(truth = test_data$RENTED_BIKE_COUNT)

head(test_results_glment)
```

A tibble: 6 × 2

| .pred <dbl> | truth <dbl> |
|---|---|
| 125.97474 | 254 |
| -13.93975 | 204 |
| -235.76937 | 100 |
| 97.92126 | 460 |
| 364.40560 | 930 |
| 527.00551 | 398 |

[23]:
```
test_results_glment[test_results_glment<0] <- 0
rsq_3<-rsq(test_results_glment, truth = truth, estimate = .pred)

rmse_3<-rmse(test_results_glment, truth = truth, estimate = .pred)

model_3_results<-c(rsq_3, rmse_3)
model_3_results
```

**$.metric** 'rsq'

**$.estimator** 'standard'

**$.estimate** 0.776136730923099

**$.metric** 'rmse'

**$.estimator** 'standard'

**$.estimate** 302.333828956636

## 4 TASK: Experiment to search for improved models

Now you understand some of the methods that you can use to try to improve your models.

*TODO:* Experiment by building and testing at least five different models. For each of your experiments, include polynomial terms, interaction terms, and one of the three regularizations we introduced.

[24]:
```
# Build at least five different models using polynomial terms, interaction␣
↪terms, and regularizations.

# Save their rmse and rsq values
```

[25]:
```
model_prediction<- function(model_fit, test_data )  {
  test_results<-model_fit %>%
    predict(new_data = test_data) %>%
    mutate(truth = test_data$RENTED_BIKE_COUNT)
  test_results[test_results<0] <- 0
  return(test_results)}
```

[26]:
```
model_evaluation <- function(test_results){
    rsq_model<-rsq(test_results, truth = truth, estimate = .pred)
    rmse_model<-rmse(test_results, truth = truth, estimate = .pred)
    results<-c(rsq_model, rmse_model)
    return(results)
}
```

[ ]:
```
lm_spec <- linear_reg() %>%
  set_engine("lm") %>%
  set_mode("regression")
```

[27]:
```
model_4_fit <- lm_spec %>% fit(RENTED_BIKE_COUNT ~␣
↪`18`*TEMPERATURE*DEW_POINT_TEMPERATURE + RAINFALL*HUMIDITY*`4`␣
↪+SOLAR_RADIATION*SNOWFALL+
                               WIND_SPEED*VISIBILITY␣
↪+`18`*TEMPERATURE+RAINFALL*HUMIDITY*`18` +poly(TEMPERATURE, 6) +␣
↪poly(HUMIDITY, 6)+ poly(WIND_SPEED, 2)+ poly(VISIBILITY, 3)+
                               poly(DEW_POINT_TEMPERATURE,6)+␣
↪poly(SOLAR_RADIATION, 5) + poly(RAINFALL, 6)+ poly(SNOWFALL, 2) + `18`+`4` +
                               `0` + `1` + `10` + `11` + `12` + `13` + `14` +␣
↪`15` + `16` + `17`  + `19` + `2` + `20` +
                               `21` + `22` + `23` + `3`  + `5` + `6` + `7` + `8`+␣
↪`9` + AUTUMN+ SPRING + SUMMER + WINTER + HOLIDAY + NO_HOLIDAY+
                               AUTUMN*SPRING*SUMMER*WINTER*HOLIDAY*`18`*`4`*`19`,␣
↪data = train_data )

results_model_4<-model_prediction(model_4_fit, test_data)
model_4_results<-model_evaluation(results_model_4)
model_4_results
```

```
Warning message in predict.lm(object = object$fit, newdata = new_data, type =
"response"):
```

"prediction from a rank-deficient fit may be misleading"

**$.metric** 'rsq'

**$.estimator** 'standard'

**$.estimate** 0.782932552014311

**$.metric** 'rmse'

**$.estimator** 'standard'

**$.estimate** 296.977827428816

[28]:
```
glmnet_spec <- linear_reg(penalty = 0.2, mixture = 0.5) %>%
  set_engine("glmnet") %>%
  set_mode("regression")
```

[29]:
```
glmnet_fit<-glmnet_spec %>% fit(RENTED_BIKE_COUNT ~␣
 ↪`18`*TEMPERATURE*DEW_POINT_TEMPERATURE + RAINFALL*HUMIDITY*`4` +
                                SOLAR_RADIATION*SNOWFALL+ WIND_SPEED*VISIBILITY␣
 ↪+`18`*TEMPERATURE+RAINFALL*HUMIDITY*`18` +
                                poly(TEMPERATURE, 6) + poly(HUMIDITY, 6)+␣
 ↪poly(WIND_SPEED, 2)+ poly(VISIBILITY, 3)+
                                poly(DEW_POINT_TEMPERATURE,6)+␣
 ↪poly(SOLAR_RADIATION, 5) + poly(RAINFALL, 6)+ poly(SNOWFALL, 2) +
                                `18`+`4`+ `0` + `1` + `10` + `11` + `12` + `13`␣
 ↪+ `14` + `15` + `16` + `17`  + `19` +
                                `2` + `20` + `21` + `22` + `23` + `3`  + `5` +␣
 ↪`6` + `7` + `8`+ `9` +
                                AUTUMN+ SPRING + SUMMER + WINTER + HOLIDAY +␣
 ↪NO_HOLIDAY+
                               ␣
 ↪AUTUMN*SPRING*SUMMER*WINTER*HOLIDAY*`18`*`4`*`19`, data = train_data )



results_model_5<-model_prediction(glmnet_fit, test_data)
model_5_results<-model_evaluation(results_model_5)
model_5_results
```

**$.metric** 'rsq'

**$.estimator** 'standard'

**$.estimate** 0.785709328328774

**$.metric** 'rmse'

**$.estimator** 'standard'

**$.estimate** 295.334507497087

[30]:
```
# Report the best performed model in terms of rmse and rsq
```

[31]:
```
rsq_rsme_data<-data.frame(model_1_results)
rsq_rsme_data<-rbind(rsq_rsme_data, model_2_results, model_3_results,␣
↪model_4_results, model_5_results)
rsq_rsme_data['model']<-c("model_1", "model_2", "model_3", "model_4", "model_5")
colnames(rsq_rsme_data)[6]<-"RSME"
colnames(rsq_rsme_data)[3]<-"Rsquared"
rsq_rsme_data
```

A data.frame: 5 × 7

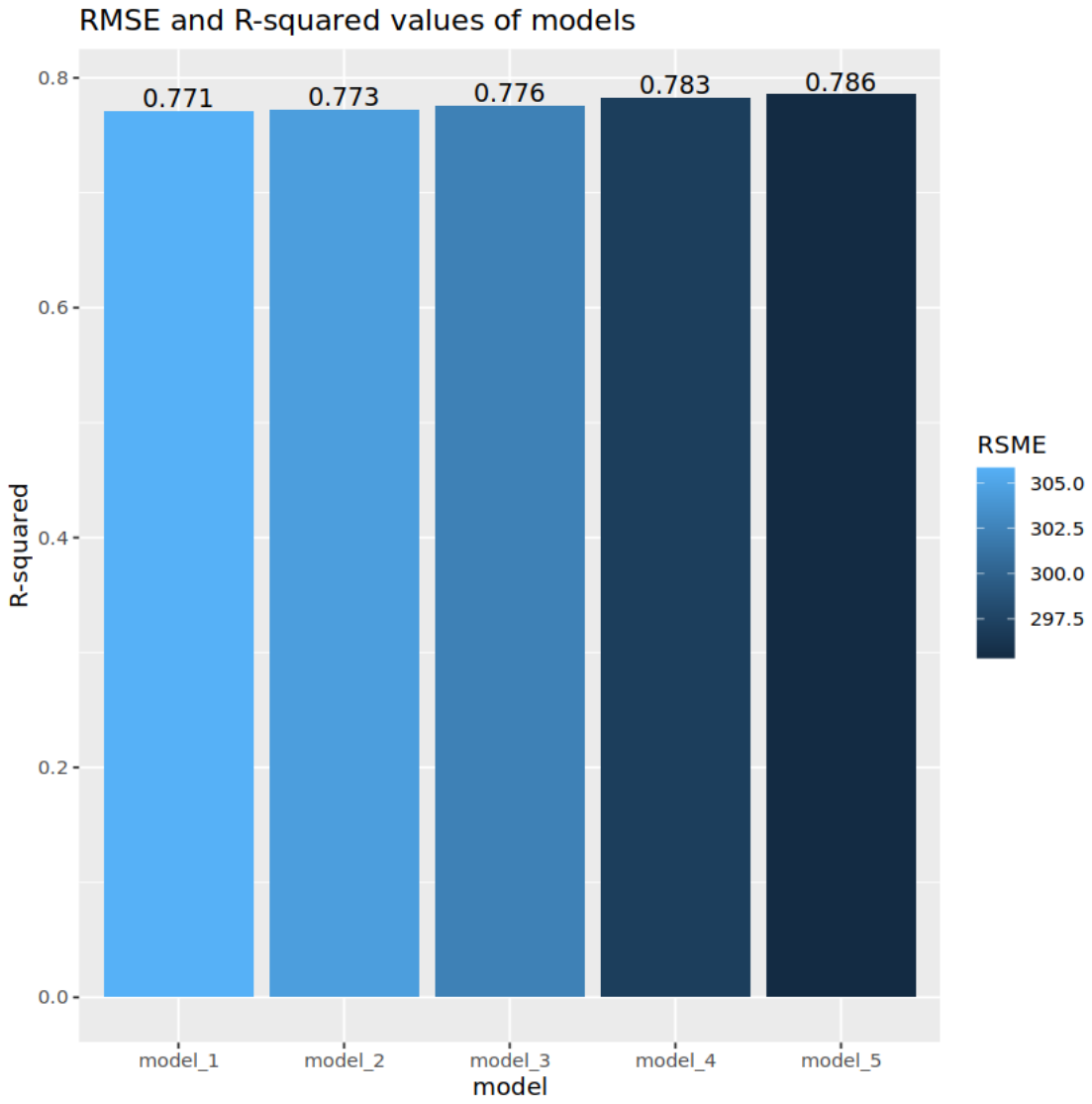
| | .metric <fct> | .estimator <fct> | Rsquared <dbl> | .metric.1 <fct> | .estimator.1 <fct> | RSME <dbl> | model <chr> |
|---|---|---|---|---|---|---|---|
| 1 | rsq | standard | 0.7712858 | rmse | standard | 305.8488 | model_1 |
| 2 | rsq | standard | 0.7726295 | rmse | standard | 304.4664 | model_2 |
| 3 | rsq | standard | 0.7761367 | rmse | standard | 302.3338 | model_3 |
| 4 | rsq | standard | 0.7829326 | rmse | standard | 296.9778 | model_4 |
| 5 | rsq | standard | 0.7857093 | rmse | standard | 295.3345 | model_5 |

Here are the performance requirements for your best model: - The RMSE should be less than 330 (rougly 10% of the max value in test dataset) - R-squared should be greater than 0.72

*TODO:* Visualize the saved RMSE and R-squared values using a grouped barchart

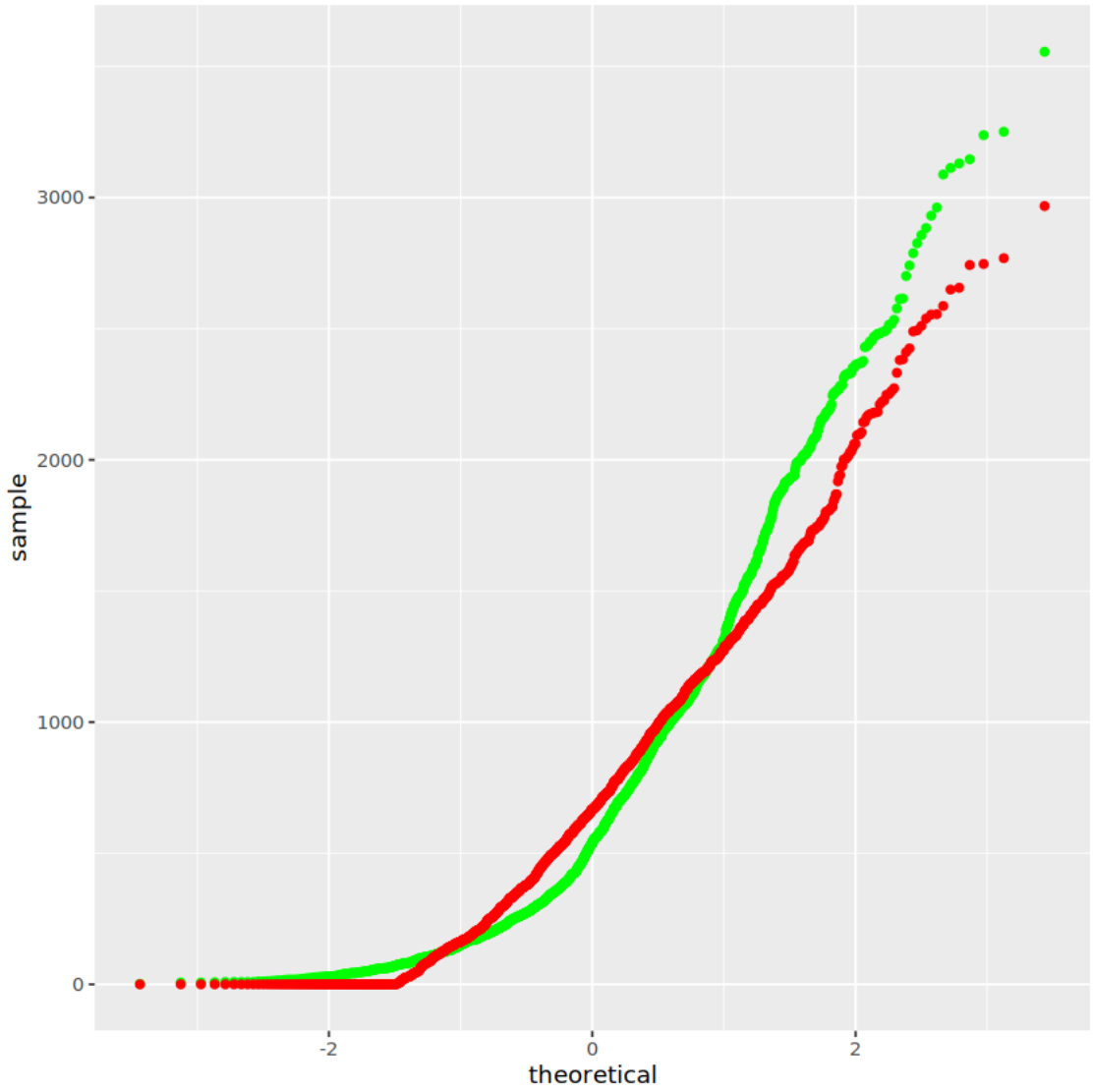
[32]:
```
# HINT: Use ggplot() + geom_bar()
library(ggplot2)

ggplot(rsq_rsme_data, aes(fill=RSME, x=model, y=Rsquared))+
  geom_bar(position="stack", stat="identity")+
  geom_text(aes(label = format(round(Rsquared, 3), nsmall = 2)), vjust = -0.2,␣
↪size = 4)+
  ylab("R-squared")+
  ggtitle("RMSE and R-squared values of models ")
```




*TODO:* Create a Q-Q plot by plotting the distribution difference between the predictions generated by your best model and the true values on the test dataset.

[33]:
```
# HINT: Use ggplot() +
    # stat_qq(aes(sample=truth), color='green') +
    # stat_qq(aes(sample=prediction), color='red')
    ggplot(results_model_5)+
  stat_qq(aes(sample=truth), color='green') +
  stat_qq(aes(sample=.pred), color='red')
```




One example of such Q-Q plot may look like this:

## 5 More model improvment methods beyond this course

In addition to the methods mentioned in this lab and previous data analysis courses, you could also explore to try the following methods yourself to see if they could improve model performance: - Remove potential redundant variables. If two variables have extremly high correlated, it is possible that they are redundant and could be removed from the model to improve the performance. - Remove some outliers. Linear regression models are very sensitive to outliers, you could try to remove some outliers to see if it would improve performance - Apply logarithm transformation. In case variable distributions are not normal distribution such as log-normal distribution, you could apply logaritm transformation on the variable to make them more look like normal distribution. In addition, logarithm transformation helps capture the non-linear relationships.

If you have time, you could research and try more methods by searching related research papers/articles, discussion forums, etc. If you know how to use other machine learning models with `Tidymodels` such as Neural Networks, Tree models, or Boosting models, you can also try and compare them with the linear regression models.

# 6 Next Steps:

Great! You have improved your baseline model using polynomial terms, interaction terms, and regularizations, and have found your best model.

Now it's time to build an interactive dashboard to provide more interactive user-interactions.

## 6.1 Authors

Yan Luo

### 6.1.1 Other Contributors

Jeff Grossman

## 6.2 Change Log

| Date (YYYY-MM-DD) | Version | Changed By | Change Description |
|---|---|---|---|
| 2021-04-08 | 1.0 | Yan | Initial version created |

##