# final-project-machine-learning

Merly Klaas

2022-12-08

```
#Import data
dat <- import(here("data","dat-category.csv"))

str(dat)
```

```
## 'data.frame':    583 obs. of  80 variables:
##  $ facilityid          : int  1001 1101 2002 2101 3001 3101 4002 4101 5001 5002 ...
##  $ id_overall          : num  1.55 1.68 1.67 1.25 2.54 ...
##  $ sd                  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ smp                 : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ hs                  : num  1 0.33 1 0.5 1 0.67 0.5 0.75 1 1 ...
##  $ s1                  : num  0 0.67 0 0.5 0 0.33 0.5 0.25 0 0 ...
##  $ group               : chr  "treatment" "control" "treatment" "control" ...
##  $ type                : chr  "KB" "TK" "KB" "TK" ...
##  $ teach_num           : int  1 3 2 2 2 3 2 4 1 2 ...
##  $ sup_pnpm            : chr  "No" "No" "No" "No" ...
##  $ sup_gov             : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ sup_uni             : chr  "No" "No" "No" "No" ...
##  $ sup_private         : chr  "No" "No" "No" "No" ...
##  $ sup_ngo             : chr  "No" "No" "No" "No" ...
##  $ sup_himpaudi        : chr  "No" "No" "No" "No" ...
##  $ sup_socind          : chr  "No" "Yes" "No" "Yes" ...
##  $ freq_wk             : num  3 6 3 6 3 6 0.3 6 2 2 ...
##  $ duration            : int  120 180 120 120 120 150 120 120 120 120 ...
##  $ tot_student         : int  23 40 29 23 33 15 43 43 30 37 ...
##  $ ratio               : int  23 13 15 12 17 5 22 11 30 19 ...
##  $ room_age_separate   : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ parents_inv         : chr  "Yes" "Yes" "No" "Yes" ...
##  $ refresher_training  : chr  "Yes" "Yes" "Yes" "No" ...
##  $ visit_gov           : int  4 2 0 0 1 0 0 0 5 1 ...
##  $ room_avail          : int  1 1 2 1 1 1 1 1 1 1 ...
##  $ book                : int  1 1 1 1 1 3 1 1 1 1 ...
##  $ book_no             : int  25 60 11 23 20 NA 20 5 35 10 ...
##  $ clean_water         : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ toilet              : int  0 0 0 0 1 2 0 0 0 0 ...
##  $ bin                 : int  1 1 0 1 1 1 1 1 0 1 ...
##  $ sink                : int  0 2 0 0 1 0 0 0 0 0 ...
##  $ weight_scale        : int  1 0 1 0 1 0 1 2 0 0 ...
##  $ heigth              : int  0 0 1 0 1 0 1 2 1 1 ...
##  $ head_measure        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ firstaid            : int  1 2 1 0 0 0 1 0 0 0 ...
##  $ facilities          : int  17 23 16 8 21 14 11 12 14 9 ...
```

```
##  $ population          : int  970 970 1115 1115 1134 1134 1209 1209 1218 1218 ...
##  $ household           : int  287 287 397 397 384 384 279 279 299 299 ...
##  $ poor_hh             : int  90 90 112 112 254 254 100 100 119 119 ...
##  $ poor_pct            : num  31.4 31.4 28.2 28.2 66.1 ...
##  $ poor_level          : chr  "Moderate-High" "Moderate-High" "Moderate-High" "Moderate-High" ...
##  $ bpd                 : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ bpd_meet            : int  0 0 0 0 1 1 6 6 1 1 ...
##  $ cp_koperasi         : int  4 4 1 1 1 1 4 4 1 1 ...
##  $ cp_gotong_royong    : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ cp_karangta3a       : int  3 3 4 4 4 4 4 4 3 3 ...
##  $ cp_health           : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ cp_women            : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ cp_labor            : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ mean_cp             : num  3.33 3.33 3 3 3 ...
##  $ safe_play           : int  3 3 3 3 3 3 3 3 2 2 ...
##  $ safe_clean          : int  2 2 2 2 3 3 3 3 2 2 ...
##  $ safe_availplayground: int  1 1 3 3 3 3 3 3 3 3 ...
##  $ safe_noharm         : int  3 3 3 3 3 3 4 4 3 3 ...
##  $ safe_noconflict     : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ safe_ownership      : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ mean_safe           : num  2.5 2.5 2.83 2.83 3 ...
##  $ pp_raskin           : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ pp_blt              : int  2 2 2 2 1 1 1 1 2 2 ...
##  $ pp_pkh              : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ pp_jamkesmas        : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ pp_kur              : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ pp_wash             : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ pp_pnpminf          : int  2 2 2 2 2 2 4 4 2 2 ...
##  $ pp_pnpmeco          : int  4 4 4 4 1 1 3 3 1 1 ...
##  $ pp_pnpmsoc          : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ pp_pnpmgen          : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ mean_pp             : num  1.9 1.9 1.9 1.9 1.5 1.5 1.9 1.9 1.6 1.6 ...
##  $ market              : num  0.5 0.5 6 6 1 1 15 15 80 80 ...
##  $ capital_district    : num  0.7 0.7 6 6 9 ...
##  $ capital_regency     : num  100 100 98 98 113 113 95 95 80 80 ...
##  $ capital_prov        : num  320 320 288 288 309 309 360 360 275 275 ...
##  $ bank                : num  100 100 98 98 113 113 95 95 80 80 ...
##  $ terminal            : num  100 100 98 98 113 113 15 15 80 80 ...
##  $ health_service      : num  0.3 0.3 6 6 9 ...
##  $ hospital            : num  100 100 98 98 113 113 95 95 70 70 ...
##  $ primary_sc          : num  1 1 0.5 0.5 0.5 0.5 1 1 1 1 ...
##  $ junior_sc           : num  0.7 0.7 1.5 1.5 1 ...
##  $ high_school         : num  2 2 1.5 1.5 1 1 3.5 3.5 2 2 ...
##  $ university          : num  100 100 98 98 113 113 95 95 80 80 ...
```

```
ff_glimpse(dat)
```

```
## $Continuous
##                                label var_type   n missing_n
## facilityid                  facilityid    <int> 583         0
## id_overall                  id_overall    <dbl> 583         0
## sd                                  sd    <dbl> 583         0
## smp                                smp    <dbl> 583         0
## hs                                  hs    <dbl> 583         0
```

```
## s1                                   s1    <dbl> 583         0
## teach_num                       teach_num    <int> 583         0
## freq_wk                           freq_wk    <dbl> 582         1
## duration                         duration    <int> 582         1
## tot_student                   tot_student    <int> 583         0
## ratio                               ratio    <int> 583         0
## room_age_separate       room_age_separate    <int> 582         1
## visit_gov                       visit_gov    <int> 583         0
## room_avail                     room_avail    <int> 583         0
## book                                 book    <int> 583         0
## book_no                           book_no    <int> 541        42
## clean_water                   clean_water    <int> 583         0
## toilet                             toilet    <int> 583         0
## bin                                   bin    <int> 583         0
## sink                                 sink    <int> 583         0
## weight_scale                 weight_scale    <int> 583         0
## heigth                             heigth    <int> 583         0
## head_measure                 head_measure    <int> 583         0
## firstaid                         firstaid    <int> 583         0
## facilities                     facilities    <int> 583         0
## population                     population    <int> 583         0
## household                       household    <int> 579         4
## poor_hh                           poor_hh    <int> 566        17
## poor_pct                         poor_pct    <dbl> 564        19
## bpd_meet                         bpd_meet    <int> 571        12
## cp_koperasi                   cp_koperasi    <int> 583         0
## cp_gotong_royong         cp_gotong_royong    <int> 583         0
## cp_karangta3a               cp_karangta3a    <int> 583         0
## cp_health                       cp_health    <int> 583         0
## cp_women                         cp_women    <int> 583         0
## cp_labor                         cp_labor    <int> 583         0
## mean_cp                           mean_cp    <dbl> 583         0
## safe_play                       safe_play    <int> 583         0
## safe_clean                     safe_clean    <int> 583         0
## safe_availplayground safe_availplayground    <int> 583         0
## safe_noharm                   safe_noharm    <int> 583         0
## safe_noconflict           safe_noconflict    <int> 583         0
## safe_ownership             safe_ownership    <int> 583         0
## mean_safe                       mean_safe    <dbl> 583         0
## pp_raskin                       pp_raskin    <int> 583         0
## pp_blt                             pp_blt    <int> 583         0
## pp_pkh                             pp_pkh    <int> 583         0
## pp_jamkesmas                 pp_jamkesmas    <int> 583         0
## pp_kur                             pp_kur    <int> 583         0
## pp_wash                           pp_wash    <int> 583         0
## pp_pnpminf                     pp_pnpminf    <int> 583         0
## pp_pnpmeco                     pp_pnpmeco    <int> 583         0
## pp_pnpmsoc                     pp_pnpmsoc    <int> 583         0
## pp_pnpmgen                     pp_pnpmgen    <int> 583         0
## mean_pp                           mean_pp    <dbl> 583         0
## market                             market    <dbl> 583         0
## capital_district         capital_district    <dbl> 583         0
## capital_regency           capital_regency    <dbl> 583         0
## capital_prov                 capital_prov    <dbl> 583         0
```

```
## bank                              bank    <dbl> 583         0
## terminal                       terminal    <dbl> 583         0
## health_service           health_service    <dbl> 583         0
## hospital                       hospital    <dbl> 583         0
## primary_sc                   primary_sc    <dbl> 583         0
## junior_sc                     junior_sc    <dbl> 583         0
## high_school                 high_school    <dbl> 583         0
## university                   university    <dbl> 583         0
##                      missing_percent     mean      sd    min quartile_25
## facilityid                       0.0 155464.0 87835.4 1001.0     81551.5
## id_overall                       0.0      2.9     1.0    1.0         2.2
## sd                               0.0      0.0     0.0    0.0         0.0
## smp                              0.0      0.0     0.1    0.0         0.0
## hs                               0.0      0.6     0.4    0.0         0.2
## s1                               0.0      0.4     0.4    0.0         0.0
## teach_num                        0.0      3.2     1.8    1.0         2.0
## freq_wk                          0.2      5.3     1.2    0.3         4.2
## duration                         0.2    136.0    80.7   15.0       120.0
## tot_student                      0.0     35.8    21.0    0.0        21.5
## ratio                            0.0     13.2     7.5    0.0         8.0
## room_age_separate                0.2      0.6     0.5    0.0         0.0
## visit_gov                        0.0      2.2     3.0    0.0         0.0
## room_avail                       0.0      1.7     0.9    1.0         1.0
## book                             0.0      1.1     0.5    0.0         1.0
## book_no                          7.2     72.7   123.3    2.0        15.0
## clean_water                      0.0      0.9     0.6    0.0         1.0
## toilet                           0.0      0.9     0.6    0.0         1.0
## bin                              0.0      1.0     0.5    0.0         1.0
## sink                             0.0      0.7     0.7    0.0         0.0
## weight_scale                     0.0      0.8     0.5    0.0         0.0
## heigth                           0.0      0.7     0.5    0.0         0.0
## head_measure                     0.0      0.5     0.5    0.0         0.0
## firstaid                         0.0      1.0     0.6    0.0         1.0
## facilities                       0.0     20.6     6.4    1.0        16.5
## population                       0.0   4346.8  3090.9  316.0      1908.0
## household                        0.7   1312.8  1067.9   92.0       520.5
## poor_hh                          2.9    465.5   460.9   23.0       168.0
## poor_pct                         3.3     36.8    16.0    5.6        24.7
## bpd_meet                         2.1      2.8     2.0    0.0         1.0
## cp_koperasi                      0.0      3.8     0.7    1.0         4.0
## cp_gotong_royong                 0.0      3.7     0.5    1.0         4.0
## cp_karangta3a                    0.0      3.5     0.7    1.0         3.0
## cp_health                        0.0      2.9     1.4    1.0         1.0
## cp_women                         0.0      3.9     0.4    1.0         4.0
## cp_labor                         0.0      4.0     0.2    1.0         4.0
## mean_cp                          0.0      3.6     0.3    2.7         3.5
## safe_play                        0.0      3.0     0.4    1.0         3.0
## safe_clean                       0.0      2.9     0.4    1.0         3.0
## safe_availplayground             0.0      1.9     1.0    1.0         1.0
## safe_noharm                      0.0      3.1     0.5    1.0         3.0
## safe_noconflict                  0.0      3.1     0.5    1.0         3.0
## safe_ownership                   0.0      3.0     0.4    1.0         3.0
## mean_safe                        0.0      2.8     0.3    1.7         2.7
## pp_raskin                        0.0      1.0     0.2    1.0         1.0
```

```
## pp_blt                        0.0      2.0     0.4    1.0         2.0
## pp_pkh                        0.0      2.7     1.5    1.0         1.0
## pp_jamkesmas                  0.0      3.9     0.3    1.0         4.0
## pp_kur                        0.0      2.7     1.4    1.0         1.0
## pp_wash                       0.0      2.3     1.3    1.0         1.0
## pp_pnpminf                    0.0      3.6     0.8    1.0         4.0
## pp_pnpmeco                    0.0      3.6     1.0    1.0         4.0
## pp_pnpmsoc                    0.0      1.9     1.3    1.0         1.0
## pp_pnpmgen                    0.0      1.8     1.3    1.0         1.0
## mean_pp                       0.0      2.6     0.5    1.4         2.2
## market                        0.0      6.4    19.9    0.0         0.7
## capital_district              0.0      6.9     9.7    0.0         3.0
## capital_regency               0.0     33.3    47.5    1.0        12.0
## capital_prov                  0.0    379.5  1571.3    2.0        40.0
## bank                          0.0     10.1    18.4    0.0         2.0
## terminal                      0.0     18.2    41.6    0.0         3.0
## health_service                0.0      1.9     4.5    0.0         0.2
## hospital                      0.0     26.4    41.4    0.5         8.0
## primary_sc                    0.0      4.3    58.6    0.0         0.1
## junior_sc                     0.0      1.8     2.5    0.0         0.3
## high_school                   0.0      4.4     5.0    0.0         1.0
## university                    0.0     30.3    41.4    0.1         9.0
##                        median quartile_75      max
## facilityid           155102.0    229102.5 310101.0
## id_overall                2.8         3.6      5.7
## sd                        0.0         0.0      0.7
## smp                       0.0         0.0      1.0
## hs                        0.5         1.0      1.0
## s1                        0.4         0.8      1.0
## teach_num                 3.0         4.0     18.0
## freq_wk                   6.0         6.0     14.0
## duration                120.0       150.0   1920.0
## tot_student              31.0        44.0    190.0
## ratio                    12.0        16.0     59.0
## room_age_separate         1.0         1.0      1.0
## visit_gov                 1.0         3.0     24.0
## room_avail                1.0         2.0      7.0
## book                      1.0         1.0      3.0
## book_no                  30.0        77.0    999.0
## clean_water               1.0         1.0      2.0
## toilet                    1.0         1.0      2.0
## bin                       1.0         1.0      2.0
## sink                      1.0         1.0      2.0
## weight_scale              1.0         1.0      2.0
## heigth                    1.0         1.0      2.0
## head_measure              0.0         1.0      2.0
## firstaid                  1.0         1.0      2.0
## facilities               22.0        25.0     34.0
## population             3427.0      6041.0  16605.0
## household               925.0      1886.0   7776.0
## poor_hh                 308.0       612.0   3790.0
## poor_pct                 34.5        45.4    100.0
## bpd_meet                  3.0         3.5     11.0
## cp_koperasi               4.0         4.0      4.0
```

```
## cp_gotong_royong          4.0         4.0      4.0
## cp_karangta3a             4.0         4.0      4.0
## cp_health                 4.0         4.0      4.0
## cp_women                  4.0         4.0      4.0
## cp_labor                  4.0         4.0      4.0
## mean_cp                   3.7         4.0      4.0
## safe_play                 3.0         3.0      4.0
## safe_clean                3.0         3.0      4.0
## safe_availplayground      1.0         3.0      3.0
## safe_noharm               3.0         3.0      8.0
## safe_noconflict           3.0         3.0      8.0
## safe_ownership            3.0         3.0      4.0
## mean_safe                 2.8         3.0      4.3
## pp_raskin                 1.0         1.0      2.0
## pp_blt                    2.0         2.0      4.0
## pp_pkh                    4.0         4.0      4.0
## pp_jamkesmas              4.0         4.0      4.0
## pp_kur                    3.0         4.0      4.0
## pp_wash                   2.0         4.0      4.0
## pp_pnpminf                4.0         4.0      4.0
## pp_pnpmeco                4.0         4.0      4.0
## pp_pnpmsoc                1.0         3.0      4.0
## pp_pnpmgen                1.0         3.0      4.0
## mean_pp                   2.6         2.9      3.6
## market                    2.5         5.0    250.0
## capital_district          5.0         7.0    118.0
## capital_regency          20.0        35.0    415.0
## capital_prov             90.0       180.0  10000.0
## bank                      5.0        10.0    160.0
## terminal                  7.0        15.0    390.0
## health_service            0.8         2.0     50.0
## hospital                 15.0        27.0    390.0
## primary_sc                0.3         0.5    999.0
## junior_sc                 1.0         2.5     30.0
## high_school               3.0         6.0     32.0
## university               18.0        35.0    350.0
##
## $Categorical
##                                label var_type   n missing_n missing_percent
## group                          group    <chr> 583         0             0.0
## type                            type    <chr> 583         0             0.0
## sup_pnpm                    sup_pnpm    <chr> 583         0             0.0
## sup_gov                      sup_gov    <chr> 583         0             0.0
## sup_uni                      sup_uni    <chr> 583         0             0.0
## sup_private                sup_private    <chr> 583         0             0.0
## sup_ngo                        sup_ngo    <chr> 583         0             0.0
## sup_himpaudi              sup_himpaudi    <chr> 583         0             0.0
## sup_socind                  sup_socind    <chr> 583         0             0.0
## parents_inv                parents_inv    <chr> 583         0             0.0
## refresher_training refresher_training    <chr> 583         0             0.0
## poor_level                  poor_level    <chr> 583         0             0.0
## bpd                                bpd    <chr> 583         0             0.0
##                    levels_n levels levels_count levels_percent
## group                     2      -            -              -
```

```
## type                     4      -           -           -
## sup_pnpm                 3      -           -           -
## sup_gov                  3      -           -           -
## sup_uni                  3      -           -           -
## sup_private              3      -           -           -
## sup_ngo                  3      -           -           -
## sup_himpaudi             3      -           -           -
## sup_socind               3      -           -           -
## parents_inv              2      -           -           -
## refresher_training       2      -           -           -
## poor_level               5      -           -           -
## bpd                      2      -           -           -
```

```
dat %>%
select_if((is.character))%>%
Hmisc::describe()
```

```
## .
##
##  13  Variables      583  Observations
## ---------------------------------------------------------------------------
## group
##        n  missing distinct
##      583        0        2
##
## Value        control treatment
## Frequency        346       237
## Proportion     0.593     0.407
## ---------------------------------------------------------------------------
## type
##        n  missing distinct
##      583        0        4
##
## Value         KB    RA    TK   TPQ
## Frequency    304    52   222     5
## Proportion 0.521 0.089 0.381 0.009
## ---------------------------------------------------------------------------
## sup_pnpm
##        n  missing distinct
##      534       49        2
##
## Value       No Yes
## Frequency  427 107
## Proportion 0.8 0.2
## ---------------------------------------------------------------------------
## sup_gov
##        n  missing distinct
##      534       49        2
##
## Value         No   Yes
## Frequency    103   431
## Proportion 0.193 0.807
## ---------------------------------------------------------------------------
## sup_uni
```

```
##        n  missing distinct
##      534       49        2
##
## Value         No   Yes
## Frequency    501    33
## Proportion 0.938 0.062
## ---------------------------------------------------------------------------
## sup_private
##        n  missing distinct
##      534       49        2
##
## Value         No   Yes
## Frequency    506    28
## Proportion 0.948 0.052
## ---------------------------------------------------------------------------
## sup_ngo
##        n  missing distinct
##      534       49        2
##
## Value         No   Yes
## Frequency    505    29
## Proportion 0.946 0.054
## ---------------------------------------------------------------------------
## sup_himpaudi
##        n  missing distinct
##      534       49        2
##
## Value         No   Yes
## Frequency    510    24
## Proportion 0.955 0.045
## ---------------------------------------------------------------------------
## sup_socind
##        n  missing distinct
##      534       49        2
##
## Value         No   Yes
## Frequency    261   273
## Proportion 0.489 0.511
## ---------------------------------------------------------------------------
## parents_inv
##        n  missing distinct
##      583        0        2
##
## Value         No   Yes
## Frequency     57   526
## Proportion 0.098 0.902
## ---------------------------------------------------------------------------
## refresher_training
##        n  missing distinct
##      583        0        2
##
## Value         No   Yes
## Frequency    112   471
## Proportion 0.192 0.808
```

```
## ---------------------------------------------------------------------------
## poor_level
##        n  missing distinct
##      564       19        4
##
## Value               High           Low Moderate-High  Moderate-Low
## Frequency            201            26           299            38
## Proportion         0.356         0.046         0.530         0.067
## ---------------------------------------------------------------------------
## bpd
##        n  missing distinct
##      583        0        2
##
## Value         3  Yes
## Frequency     6  577
## Proportion 0.01 0.99
## ---------------------------------------------------------------------------
```

```
blueprint <- recipe(x  = dat,
                    vars  = colnames(dat),
                    roles = c('ID','outcome',rep('predictor',78))) %>%
                  step_indicate_na(all_predictors()) %>%
                step_zv(all_numeric())%>%
                 step_impute_mean(all_numeric_predictors()) %>%
                 step_impute_mode(all_nominal()) %>%
                 step_poly("poor_pct",degree=3) %>%
                 step_normalize(all_numeric_predictors())%>%
                 step_dummy(all_nominal(), one_hot=TRUE)
blueprint
```

```
## Recipe
##
## Inputs:
##
##       role #variables
##         ID          1
##    outcome          1
##  predictor         78
##
## Operations:
##
## Creating missing data variable indicators for all_predictors()
## Zero variance filter on all_numeric()
## Mean imputation for all_numeric_predictors()
## Mode imputation for all_nominal()
## Orthogonal polynomials on "poor_pct"
## Centering and scaling for all_numeric_predictors()
## Dummy variables from all_nominal()
```

```
view(blueprint %>% prep() %>% summary)
```

**Split the data for: training and test subsets. . Let the training data have the 80% of cases and the test data have the 20% of the cases. Set the seed to 1031000 for any random sampling process before splitting data.**

```
set.seed(1031000)
loc      <- sample(1:nrow(dat), round(nrow(dat) * 0.8))
ind_tr  <- dat[loc, ]
ind_te  <- dat[-loc, ]

dim(ind_tr)
```

```
## [1] 466  80
```

```
dim(ind_te)
```

```
## [1] 117  80
```

```
# Randomly shuffle the training dataset

  set.seed(1031000) # for reproducibility

  ind_tr = ind_tr[sample(nrow(ind_tr)),]

# Create 5 folds with equal size

  folds = cut(seq(1,nrow(ind_tr)),breaks=10,labels=FALSE)

# Create the list for each fold

  my.indices <- vector('list',10)

  for(i in 1:10){
    my.indices[[i]] <- which(folds!=i)
  }

cv <- trainControl(method = "cv",
                   index  = my.indices)
```

### Model 1 Ridge Regression**

```
gridrd <- data.frame(alpha = 0, lambda = seq(0.01,3,.01))
gridrd
```

```
##     alpha lambda
## 1       0   0.01
## 2       0   0.02
## 3       0   0.03
## 4       0   0.04
## 5       0   0.05
## 6       0   0.06
## 7       0   0.07
## 8       0   0.08
## 9       0   0.09
```

```
## 10      0   0.10
## 11      0   0.11
## 12      0   0.12
## 13      0   0.13
## 14      0   0.14
## 15      0   0.15
## 16      0   0.16
## 17      0   0.17
## 18      0   0.18
## 19      0   0.19
## 20      0   0.20
## 21      0   0.21
## 22      0   0.22
## 23      0   0.23
## 24      0   0.24
## 25      0   0.25
## 26      0   0.26
## 27      0   0.27
## 28      0   0.28
## 29      0   0.29
## 30      0   0.30
## 31      0   0.31
## 32      0   0.32
## 33      0   0.33
## 34      0   0.34
## 35      0   0.35
## 36      0   0.36
## 37      0   0.37
## 38      0   0.38
## 39      0   0.39
## 40      0   0.40
## 41      0   0.41
## 42      0   0.42
## 43      0   0.43
## 44      0   0.44
## 45      0   0.45
## 46      0   0.46
## 47      0   0.47
## 48      0   0.48
## 49      0   0.49
## 50      0   0.50
## 51      0   0.51
## 52      0   0.52
## 53      0   0.53
## 54      0   0.54
## 55      0   0.55
## 56      0   0.56
## 57      0   0.57
## 58      0   0.58
## 59      0   0.59
## 60      0   0.60
## 61      0   0.61
## 62      0   0.62
## 63      0   0.63
```

```
## 64      0   0.64
## 65      0   0.65
## 66      0   0.66
## 67      0   0.67
## 68      0   0.68
## 69      0   0.69
## 70      0   0.70
## 71      0   0.71
## 72      0   0.72
## 73      0   0.73
## 74      0   0.74
## 75      0   0.75
## 76      0   0.76
## 77      0   0.77
## 78      0   0.78
## 79      0   0.79
## 80      0   0.80
## 81      0   0.81
## 82      0   0.82
## 83      0   0.83
## 84      0   0.84
## 85      0   0.85
## 86      0   0.86
## 87      0   0.87
## 88      0   0.88
## 89      0   0.89
## 90      0   0.90
## 91      0   0.91
## 92      0   0.92
## 93      0   0.93
## 94      0   0.94
## 95      0   0.95
## 96      0   0.96
## 97      0   0.97
## 98      0   0.98
## 99      0   0.99
## 100     0   1.00
## 101     0   1.01
## 102     0   1.02
## 103     0   1.03
## 104     0   1.04
## 105     0   1.05
## 106     0   1.06
## 107     0   1.07
## 108     0   1.08
## 109     0   1.09
## 110     0   1.10
## 111     0   1.11
## 112     0   1.12
## 113     0   1.13
## 114     0   1.14
## 115     0   1.15
## 116     0   1.16
## 117     0   1.17
```

```
## 118     0   1.18
## 119     0   1.19
## 120     0   1.20
## 121     0   1.21
## 122     0   1.22
## 123     0   1.23
## 124     0   1.24
## 125     0   1.25
## 126     0   1.26
## 127     0   1.27
## 128     0   1.28
## 129     0   1.29
## 130     0   1.30
## 131     0   1.31
## 132     0   1.32
## 133     0   1.33
## 134     0   1.34
## 135     0   1.35
## 136     0   1.36
## 137     0   1.37
## 138     0   1.38
## 139     0   1.39
## 140     0   1.40
## 141     0   1.41
## 142     0   1.42
## 143     0   1.43
## 144     0   1.44
## 145     0   1.45
## 146     0   1.46
## 147     0   1.47
## 148     0   1.48
## 149     0   1.49
## 150     0   1.50
## 151     0   1.51
## 152     0   1.52
## 153     0   1.53
## 154     0   1.54
## 155     0   1.55
## 156     0   1.56
## 157     0   1.57
## 158     0   1.58
## 159     0   1.59
## 160     0   1.60
## 161     0   1.61
## 162     0   1.62
## 163     0   1.63
## 164     0   1.64
## 165     0   1.65
## 166     0   1.66
## 167     0   1.67
## 168     0   1.68
## 169     0   1.69
## 170     0   1.70
## 171     0   1.71
```

```
## 172     0   1.72
## 173     0   1.73
## 174     0   1.74
## 175     0   1.75
## 176     0   1.76
## 177     0   1.77
## 178     0   1.78
## 179     0   1.79
## 180     0   1.80
## 181     0   1.81
## 182     0   1.82
## 183     0   1.83
## 184     0   1.84
## 185     0   1.85
## 186     0   1.86
## 187     0   1.87
## 188     0   1.88
## 189     0   1.89
## 190     0   1.90
## 191     0   1.91
## 192     0   1.92
## 193     0   1.93
## 194     0   1.94
## 195     0   1.95
## 196     0   1.96
## 197     0   1.97
## 198     0   1.98
## 199     0   1.99
## 200     0   2.00
## 201     0   2.01
## 202     0   2.02
## 203     0   2.03
## 204     0   2.04
## 205     0   2.05
## 206     0   2.06
## 207     0   2.07
## 208     0   2.08
## 209     0   2.09
## 210     0   2.10
## 211     0   2.11
## 212     0   2.12
## 213     0   2.13
## 214     0   2.14
## 215     0   2.15
## 216     0   2.16
## 217     0   2.17
## 218     0   2.18
## 219     0   2.19
## 220     0   2.20
## 221     0   2.21
## 222     0   2.22
## 223     0   2.23
## 224     0   2.24
## 225     0   2.25
```

```
## 226     0   2.26
## 227     0   2.27
## 228     0   2.28
## 229     0   2.29
## 230     0   2.30
## 231     0   2.31
## 232     0   2.32
## 233     0   2.33
## 234     0   2.34
## 235     0   2.35
## 236     0   2.36
## 237     0   2.37
## 238     0   2.38
## 239     0   2.39
## 240     0   2.40
## 241     0   2.41
## 242     0   2.42
## 243     0   2.43
## 244     0   2.44
## 245     0   2.45
## 246     0   2.46
## 247     0   2.47
## 248     0   2.48
## 249     0   2.49
## 250     0   2.50
## 251     0   2.51
## 252     0   2.52
## 253     0   2.53
## 254     0   2.54
## 255     0   2.55
## 256     0   2.56
## 257     0   2.57
## 258     0   2.58
## 259     0   2.59
## 260     0   2.60
## 261     0   2.61
## 262     0   2.62
## 263     0   2.63
## 264     0   2.64
## 265     0   2.65
## 266     0   2.66
## 267     0   2.67
## 268     0   2.68
## 269     0   2.69
## 270     0   2.70
## 271     0   2.71
## 272     0   2.72
## 273     0   2.73
## 274     0   2.74
## 275     0   2.75
## 276     0   2.76
## 277     0   2.77
## 278     0   2.78
## 279     0   2.79
```

```
## 280     0   2.80
## 281     0   2.81
## 282     0   2.82
## 283     0   2.83
## 284     0   2.84
## 285     0   2.85
## 286     0   2.86
## 287     0   2.87
## 288     0   2.88
## 289     0   2.89
## 290     0   2.90
## 291     0   2.91
## 292     0   2.92
## 293     0   2.93
## 294     0   2.94
## 295     0   2.95
## 296     0   2.96
## 297     0   2.97
## 298     0   2.98
## 299     0   2.99
## 300     0   3.00
```

```
ridge_mod <- caret::train(blueprint,
                          data      = ind_tr,
                          method    = "glmnet",
                          tuneGrid  = gridrd,
                          trControl = cv)

ridge_mod
```

```
## glmnet
##
## 466 samples
##  79 predictor
##
## Recipe steps: indicate_na, zv, impute_mean, impute_mode, poly, normalize, dummy
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 419, 419, 420, 419, 420, 419, ...
## Resampling results across tuning parameters:
##
##   lambda  RMSE       Rsquared   MAE
##   0.01    0.8954980  0.3184985  0.7140117
##   0.02    0.8954980  0.3184985  0.7140117
##   0.03    0.8954980  0.3184985  0.7140117
##   0.04    0.8954980  0.3184985  0.7140117
##   0.05    0.8942442  0.3193028  0.7128977
##   0.06    0.8901482  0.3217710  0.7096716
##   0.07    0.8865964  0.3239794  0.7067947
##   0.08    0.8834648  0.3259961  0.7041966
##   0.09    0.8806941  0.3278432  0.7018362
##   0.10    0.8782569  0.3295182  0.6997187
##   0.11    0.8760759  0.3310662  0.6977422
##   0.12    0.8741258  0.3324945  0.6958802
##   0.13    0.8723843  0.3338071  0.6941502
```

```
##   0.14    0.8708273  0.3350097  0.6926061
##   0.15    0.8694315  0.3361122  0.6911916
##   0.16    0.8681671  0.3371394  0.6898891
##   0.17    0.8670290  0.3380845  0.6887582
##   0.18    0.8659929  0.3389630  0.6877140
##   0.19    0.8650719  0.3397539  0.6867412
##   0.20    0.8642259  0.3404989  0.6858105
##   0.21    0.8634672  0.3411764  0.6849267
##   0.22    0.8627784  0.3418013  0.6841446
##   0.23    0.8621650  0.3423602  0.6836088
##   0.24    0.8616039  0.3428806  0.6831073
##   0.25    0.8611073  0.3433403  0.6826736
##   0.26    0.8606467  0.3437756  0.6822475
##   0.27    0.8602406  0.3441585  0.6818356
##   0.28    0.8598688  0.3445128  0.6814335
##   0.29    0.8595356  0.3448331  0.6810412
##   0.30    0.8592440  0.3451096  0.6807004
##   0.31    0.8589737  0.3453694  0.6804063
##   0.32    0.8587381  0.3455952  0.6801293
##   0.33    0.8585317  0.3457890  0.6798783
##   0.34    0.8583409  0.3459705  0.6796429
##   0.35    0.8581772  0.3461281  0.6794144
##   0.36    0.8580382  0.3462565  0.6791951
##   0.37    0.8579142  0.3463707  0.6789774
##   0.38    0.8578083  0.3464700  0.6787668
##   0.39    0.8577194  0.3465500  0.6786029
##   0.40    0.8576452  0.3466141  0.6784741
##   0.41    0.8575818  0.3466691  0.6783575
##   0.42    0.8575334  0.3467117  0.6782542
##   0.43    0.8574995  0.3467344  0.6781539
##   0.44    0.8574769  0.3467442  0.6780528
##   0.45    0.8574632  0.3467463  0.6779548
##   0.46    0.8574581  0.3467436  0.6778586
##   0.47    0.8574613  0.3467294  0.6777685
##   0.48    0.8574746  0.3467028  0.6776858
##   0.49    0.8574934  0.3466716  0.6776031
##   0.50    0.8575226  0.3466341  0.6775275
##   0.51    0.8575577  0.3465904  0.6774636
##   0.52    0.8576020  0.3465324  0.6774160
##   0.53    0.8576528  0.3464672  0.6773665
##   0.54    0.8577084  0.3463977  0.6773311
##   0.55    0.8577700  0.3463277  0.6773042
##   0.56    0.8578319  0.3462574  0.6772856
##   0.57    0.8579013  0.3461747  0.6772921
##   0.58    0.8579764  0.3460855  0.6773196
##   0.59    0.8580558  0.3459922  0.6773472
##   0.60    0.8581403  0.3458985  0.6773768
##   0.61    0.8582256  0.3458078  0.6774044
##   0.62    0.8583152  0.3457085  0.6774334
##   0.63    0.8584092  0.3456035  0.6774680
##   0.64    0.8585066  0.3454949  0.6775016
##   0.65    0.8586070  0.3453840  0.6775364
##   0.66    0.8587121  0.3452732  0.6775840
##   0.67    0.8588145  0.3451682  0.6776322
```

```
##   0.68    0.8589204  0.3450547  0.6776796
##   0.69    0.8590300  0.3449359  0.6777233
##   0.70    0.8591430  0.3448137  0.6777668
##   0.71    0.8592590  0.3446886  0.6778195
##   0.72    0.8593787  0.3445629  0.6778788
##   0.73    0.8594957  0.3444454  0.6779458
##   0.74    0.8596138  0.3443255  0.6780161
##   0.75    0.8597343  0.3441995  0.6780951
##   0.76    0.8598581  0.3440694  0.6781781
##   0.77    0.8599840  0.3439372  0.6782599
##   0.78    0.8601124  0.3438030  0.6783419
##   0.79    0.8602434  0.3436693  0.6784252
##   0.80    0.8603706  0.3435453  0.6785064
##   0.81    0.8604975  0.3434215  0.6785859
##   0.82    0.8606253  0.3432924  0.6786702
##   0.83    0.8607555  0.3431602  0.6787566
##   0.84    0.8608881  0.3430253  0.6788428
##   0.85    0.8610222  0.3428895  0.6789286
##   0.86    0.8611565  0.3427542  0.6790139
##   0.87    0.8612941  0.3426207  0.6791144
##   0.88    0.8614275  0.3424949  0.6792200
##   0.89    0.8615620  0.3423665  0.6793244
##   0.90    0.8616979  0.3422328  0.6794273
##   0.91    0.8618359  0.3420961  0.6795280
##   0.92    0.8619764  0.3419568  0.6796286
##   0.93    0.8621179  0.3418166  0.6797288
##   0.94    0.8622608  0.3416754  0.6798317
##   0.95    0.8624051  0.3415356  0.6799357
##   0.96    0.8625473  0.3414029  0.6800388
##   0.97    0.8626878  0.3412737  0.6801417
##   0.98    0.8628279  0.3411414  0.6802446
##   0.99    0.8629684  0.3410075  0.6803577
##   1.00    0.8631103  0.3408720  0.6804786
##   1.01    0.8632541  0.3407344  0.6806040
##   1.02    0.8633990  0.3405961  0.6807290
##   1.03    0.8635450  0.3404568  0.6808575
##   1.04    0.8636906  0.3403203  0.6809856
##   1.05    0.8638371  0.3401871  0.6811125
##   1.06    0.8639798  0.3400605  0.6812384
##   1.07    0.8641225  0.3399331  0.6813631
##   1.08    0.8642650  0.3398023  0.6814933
##   1.09    0.8644077  0.3396706  0.6816223
##   1.10    0.8645522  0.3395371  0.6817569
##   1.11    0.8646982  0.3394020  0.6819037
##   1.12    0.8648451  0.3392662  0.6820508
##   1.13    0.8649929  0.3391297  0.6822066
##   1.14    0.8651395  0.3389964  0.6823611
##   1.15    0.8652881  0.3388647  0.6825162
##   1.16    0.8654325  0.3387404  0.6826650
##   1.17    0.8655770  0.3386166  0.6828123
##   1.18    0.8657201  0.3384903  0.6829584
##   1.19    0.8658634  0.3383627  0.6831060
##   1.20    0.8660074  0.3382342  0.6832587
##   1.21    0.8661530  0.3381040  0.6834115
```

```
##   1.22    0.8662996  0.3379728  0.6835638
##   1.23    0.8664471  0.3378409  0.6837156
##   1.24    0.8665951  0.3377086  0.6838671
##   1.25    0.8667415  0.3375798  0.6840167
##   1.26    0.8668907  0.3374510  0.6841701
##   1.27    0.8670357  0.3373299  0.6843192
##   1.28    0.8671802  0.3372101  0.6844666
##   1.29    0.8673236  0.3370895  0.6846124
##   1.30    0.8674668  0.3369664  0.6847576
##   1.31    0.8676095  0.3368436  0.6849001
##   1.32    0.8677535  0.3367194  0.6850425
##   1.33    0.8678986  0.3365938  0.6851850
##   1.34    0.8680447  0.3364675  0.6853269
##   1.35    0.8681914  0.3363405  0.6854686
##   1.36    0.8683387  0.3362133  0.6856101
##   1.37    0.8684837  0.3360897  0.6857492
##   1.38    0.8686323  0.3359647  0.6858922
##   1.39    0.8687773  0.3358467  0.6860346
##   1.40    0.8689204  0.3357321  0.6861744
##   1.41    0.8690639  0.3356170  0.6863155
##   1.42    0.8692054  0.3355001  0.6864582
##   1.43    0.8693471  0.3353823  0.6866008
##   1.44    0.8694887  0.3352645  0.6867418
##   1.45    0.8696313  0.3351456  0.6868828
##   1.46    0.8697749  0.3350255  0.6870237
##   1.47    0.8699193  0.3349048  0.6871643
##   1.48    0.8700644  0.3347834  0.6873047
##   1.49    0.8702095  0.3346622  0.6874448
##   1.50    0.8703534  0.3345434  0.6875830
##   1.51    0.8704986  0.3344248  0.6877224
##   1.52    0.8706434  0.3343089  0.6878647
##   1.53    0.8707856  0.3341980  0.6880102
##   1.54    0.8709272  0.3340882  0.6881545
##   1.55    0.8710684  0.3339783  0.6882980
##   1.56    0.8712078  0.3338665  0.6884404
##   1.57    0.8713473  0.3337545  0.6885879
##   1.58    0.8714867  0.3336425  0.6887357
##   1.59    0.8716269  0.3335296  0.6888862
##   1.60    0.8717680  0.3334158  0.6890381
##   1.61    0.8719098  0.3333012  0.6891913
##   1.62    0.8720523  0.3331860  0.6893491
##   1.63    0.8721948  0.3330710  0.6895068
##   1.64    0.8723374  0.3329562  0.6896641
##   1.65    0.8724781  0.3328445  0.6898191
##   1.66    0.8726215  0.3327317  0.6899757
##   1.67    0.8727628  0.3326233  0.6901287
##   1.68    0.8729023  0.3325183  0.6902792
##   1.69    0.8730415  0.3324139  0.6904315
##   1.70    0.8731803  0.3323097  0.6905828
##   1.71    0.8733170  0.3322040  0.6907322
##   1.72    0.8734540  0.3320977  0.6908812
##   1.73    0.8735902  0.3319921  0.6910285
##   1.74    0.8737272  0.3318857  0.6911800
##   1.75    0.8738650  0.3317785  0.6913378
```

```
##   1.76    0.8740034  0.3316705  0.6914955
##   1.77    0.8741424  0.3315621  0.6916530
##   1.78    0.8742821  0.3314529  0.6918105
##   1.79    0.8744213  0.3313444  0.6919675
##   1.80    0.8745607  0.3312360  0.6921240
##   1.81    0.8746982  0.3311307  0.6922786
##   1.82    0.8748380  0.3310244  0.6924344
##   1.83    0.8749768  0.3309207  0.6925882
##   1.84    0.8751137  0.3308206  0.6927392
##   1.85    0.8752496  0.3307222  0.6928889
##   1.86    0.8753860  0.3306232  0.6930386
##   1.87    0.8755203  0.3305247  0.6931861
##   1.88    0.8756538  0.3304247  0.6933328
##   1.89    0.8757873  0.3303248  0.6934788
##   1.90    0.8759204  0.3302252  0.6936236
##   1.91    0.8760541  0.3301250  0.6937683
##   1.92    0.8761883  0.3300242  0.6939131
##   1.93    0.8763232  0.3299227  0.6940579
##   1.94    0.8764586  0.3298206  0.6942026
##   1.95    0.8765945  0.3297181  0.6943471
##   1.96    0.8767304  0.3296157  0.6944913
##   1.97    0.8768662  0.3295134  0.6946353
##   1.98    0.8770009  0.3294134  0.6947779
##   1.99    0.8771354  0.3293145  0.6949199
##   2.00    0.8772718  0.3292149  0.6950631
##   2.01    0.8774064  0.3291186  0.6952038
##   2.02    0.8775399  0.3290245  0.6953428
##   2.03    0.8776725  0.3289318  0.6954830
##   2.04    0.8778057  0.3288386  0.6956233
##   2.05    0.8779372  0.3287461  0.6957619
##   2.06    0.8780671  0.3286523  0.6958992
##   2.07    0.8781976  0.3285580  0.6960365
##   2.08    0.8783270  0.3284648  0.6961721
##   2.09    0.8784568  0.3283712  0.6963077
##   2.10    0.8785872  0.3282769  0.6964433
##   2.11    0.8787180  0.3281822  0.6965789
##   2.12    0.8788494  0.3280868  0.6967145
##   2.13    0.8789811  0.3279913  0.6968498
##   2.14    0.8791132  0.3278953  0.6969849
##   2.15    0.8792451  0.3277996  0.6971197
##   2.16    0.8793769  0.3277043  0.6972543
##   2.17    0.8795083  0.3276097  0.6973880
##   2.18    0.8796386  0.3275166  0.6975229
##   2.19    0.8797703  0.3274230  0.6976641
##   2.20    0.8799023  0.3273298  0.6978049
##   2.21    0.8800317  0.3272412  0.6979425
##   2.22    0.8801605  0.3271540  0.6980792
##   2.23    0.8802890  0.3270673  0.6982154
##   2.24    0.8804180  0.3269800  0.6983537
##   2.25    0.8805450  0.3268938  0.6984908
##   2.26    0.8806709  0.3268062  0.6986267
##   2.27    0.8807972  0.3267180  0.6987629
##   2.28    0.8809228  0.3266308  0.6988995
##   2.29    0.8810485  0.3265434  0.6990373
```

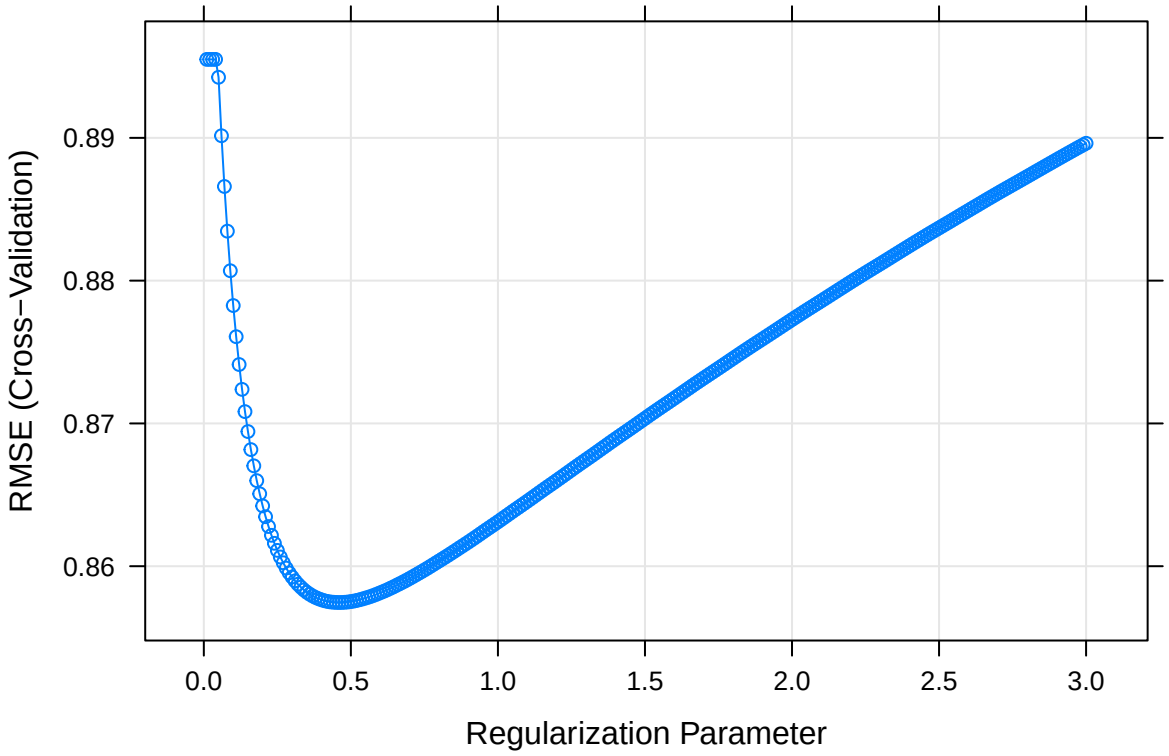
```
##   2.30    0.8811746  0.3264555  0.6991758
##   2.31    0.8813011  0.3263671  0.6993161
##   2.32    0.8814281  0.3262783  0.6994565
##   2.33    0.8815555  0.3261890  0.6995968
##   2.34    0.8816834  0.3260989  0.6997371
##   2.35    0.8818118  0.3260084  0.6998775
##   2.36    0.8819398  0.3259184  0.7000175
##   2.37    0.8820677  0.3258285  0.7001573
##   2.38    0.8821953  0.3257392  0.7002965
##   2.39    0.8823217  0.3256522  0.7004341
##   2.40    0.8824489  0.3255648  0.7005722
##   2.41    0.8825771  0.3254770  0.7007112
##   2.42    0.8827035  0.3253924  0.7008545
##   2.43    0.8828294  0.3253089  0.7009995
##   2.44    0.8829540  0.3252274  0.7011431
##   2.45    0.8830791  0.3251454  0.7012892
##   2.46    0.8832042  0.3250633  0.7014383
##   2.47    0.8833272  0.3249819  0.7015853
##   2.48    0.8834488  0.3248999  0.7017313
##   2.49    0.8835707  0.3248176  0.7018777
##   2.50    0.8836918  0.3247361  0.7020226
##   2.51    0.8838128  0.3246549  0.7021669
##   2.52    0.8839342  0.3245734  0.7023112
##   2.53    0.8840558  0.3244914  0.7024556
##   2.54    0.8841779  0.3244090  0.7025999
##   2.55    0.8843002  0.3243263  0.7027442
##   2.56    0.8844230  0.3242431  0.7028884
##   2.57    0.8845460  0.3241596  0.7030325
##   2.58    0.8846693  0.3240757  0.7031766
##   2.59    0.8847923  0.3239924  0.7033204
##   2.60    0.8849150  0.3239093  0.7034638
##   2.61    0.8850380  0.3238262  0.7036071
##   2.62    0.8851592  0.3237458  0.7037487
##   2.63    0.8852809  0.3236651  0.7038927
##   2.64    0.8854037  0.3235841  0.7040369
##   2.65    0.8855263  0.3235036  0.7041806
##   2.66    0.8856470  0.3234265  0.7043221
##   2.67    0.8857675  0.3233499  0.7044631
##   2.68    0.8858873  0.3232744  0.7046032
##   2.69    0.8860074  0.3231985  0.7047434
##   2.70    0.8861275  0.3231227  0.7048830
##   2.71    0.8862457  0.3230475  0.7050202
##   2.72    0.8863624  0.3229718  0.7051559
##   2.73    0.8864795  0.3228956  0.7052916
##   2.74    0.8865963  0.3228198  0.7054266
##   2.75    0.8867126  0.3227447  0.7055605
##   2.76    0.8868291  0.3226693  0.7056944
##   2.77    0.8869459  0.3225935  0.7058283
##   2.78    0.8870630  0.3225174  0.7059622
##   2.79    0.8871803  0.3224409  0.7060962
##   2.80    0.8872980  0.3223642  0.7062301
##   2.81    0.8874159  0.3222871  0.7063639
##   2.82    0.8875341  0.3222096  0.7064976
##   2.83    0.8876526  0.3221319  0.7066313
```

```
##   2.84    0.8877708  0.3220545  0.7067648
##   2.85    0.8878886  0.3219777  0.7068978
##   2.86    0.8880067  0.3219005  0.7070308
##   2.87    0.8881237  0.3218251  0.7071624
##   2.88    0.8882402  0.3217505  0.7072932
##   2.89    0.8883575  0.3216758  0.7074242
##   2.90    0.8884753  0.3216008  0.7075554
##   2.91    0.8885925  0.3215270  0.7076857
##   2.92    0.8887085  0.3214554  0.7078144
##   2.93    0.8888242  0.3213844  0.7079427
##   2.94    0.8889392  0.3213144  0.7080702
##   2.95    0.8890545  0.3212441  0.7081977
##   2.96    0.8891700  0.3211736  0.7083253
##   2.97    0.8892843  0.3211041  0.7084508
##   2.98    0.8893964  0.3210342  0.7085744
##   2.99    0.8895085  0.3209640  0.7086993
##   3.00    0.8896208  0.3208935  0.7088244
##
## Tuning parameter 'alpha' was held constant at a value of 0
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were alpha = 0 and lambda = 0.46.
```

```
ridge_mod$bestTune
```

```
##    alpha lambda
## 46     0   0.46
```

```
plot(ridge_mod)
```




```
predict_te_ridge <- predict(ridge_mod, ind_te)
rsq_te <- cor(ind_te$id_overall,predict_te_ridge)^2
rsq_te
```

```
## [1] 0.3010392
```

```
mae_te <- mean(abs(ind_te$id_overall - predict_te_ridge))
mae_te
```

```
## [1] 0.6469945
```

```
rmse_te <- sqrt(mean((ind_te$id_overall - predict_te_ridge)^2))
rmse_te
```

```
## [1] 0.8004419
```

## Model 2 Random forest

```
gridrf <- expand.grid(mtry = 30,splitrule='variance',min.node.size=2)
gridrf
```

**randomly sample 30 predictors**

```
##   mtry splitrule min.node.size
## 1   30  variance             2
```

```
# Run the Random Forest by iterating over num.trees using the
# values 5, 20, 40, 60,  ..., 200

 nbags <- c(5,seq(from = 20,to = 200, by = 20))

  bags <- vector('list',length(nbags))

    for(i in 1:length(nbags)){

      bags[[i]] <- caret::train(blueprint,
                                data       = ind_tr,
                                method     = 'ranger',
                                trControl  = cv,
                                tuneGrid   = gridrf,
                                num.trees  = nbags[i],
                                importance = "impurity",
                                max.depth  = 60)

      print(i)
 }
```

```
## Loading required namespace: e1071
```

```
##
## Attaching package: 'e1071'
```

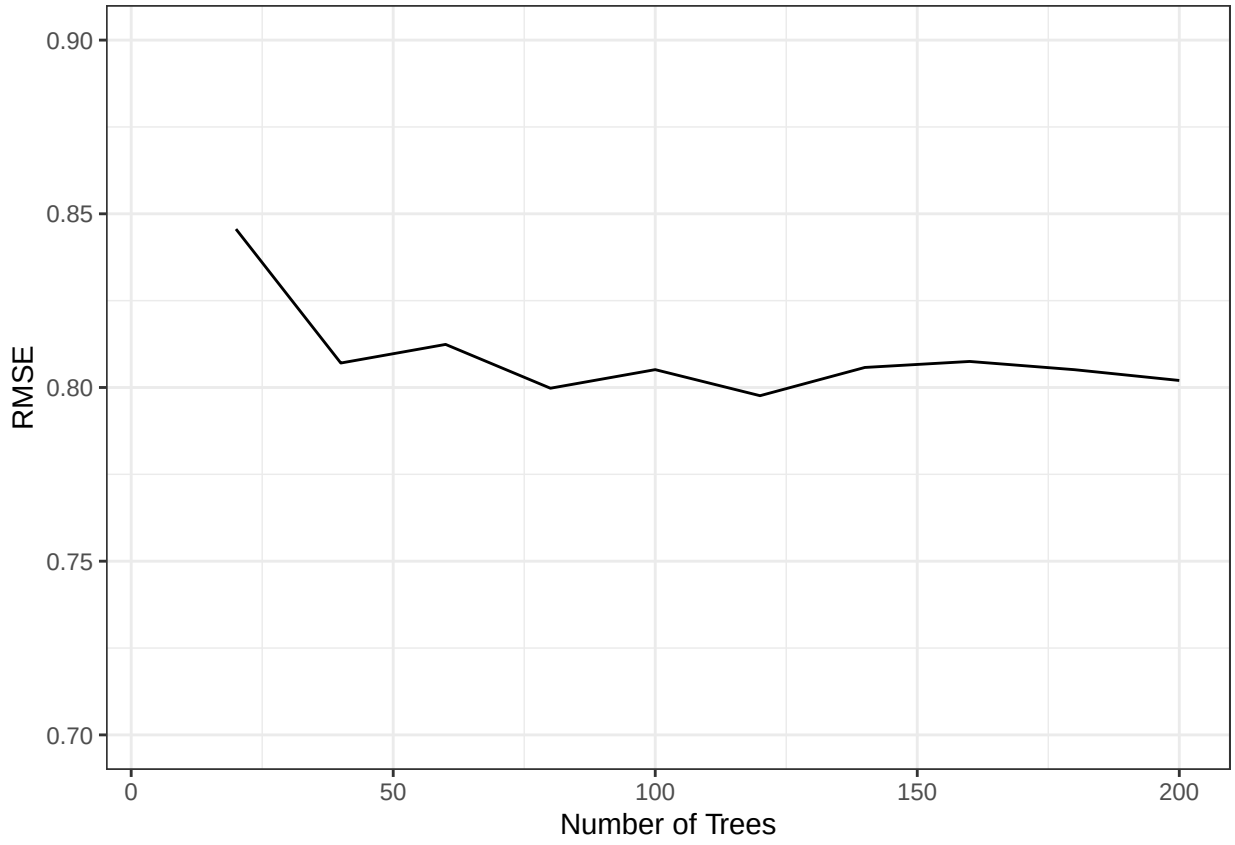
```
## The following object is masked from 'package:Hmisc':
##
##     impute
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## [1] 11
```

```
rmses <- c()

for(i in 1:length(nbags)){

  rmses[i] = bags[[i]]$results$RMSE

}
```

```
ggplot()+
  geom_line(aes(x=nbags,y=rmses))+
  xlab('Number of Trees')+
  ylab('RMSE')+
  ylim(c(0.7,0.90))+
  theme_bw()
```

```
## Warning: Removed 1 row containing missing values (`geom_line()`).
```




```
nbags[which.min(rmses)]
```

```
## [1] 120
```

```
nbags
```

```
##  [1]   5  20  40  60  80 100 120 140 160 180 200
```

**Although the model with 80 trees has the lowest RMSE for training test I test the rest of number of trees and found that the 200 trees (bag 11th) yielded largest predictive power for the test data.**

```
predicted_te <- predict(bags[[11]],ind_te)

# MAE
mean(abs(ind_te$id_overall - predicted_te))
```

```
## [1] 0.6116961
```

```
# RMSE
sqrt(mean((ind_te$id_overall - predicted_te)^2))
```

```
## [1] 0.74823
```

```
# R-square
cor(ind_te$id_overall,predicted_te)^2
```

```
## [1] 0.4018192
```

### Model 3 Gradient Boosting Trees

```
require(doParallel)
```

```
## Loading required package: doParallel

## Loading required package: foreach

##
## Attaching package: 'foreach'

## The following objects are masked from 'package:purrr':
##
##     accumulate, when

## Loading required package: iterators

## Loading required package: parallel
```

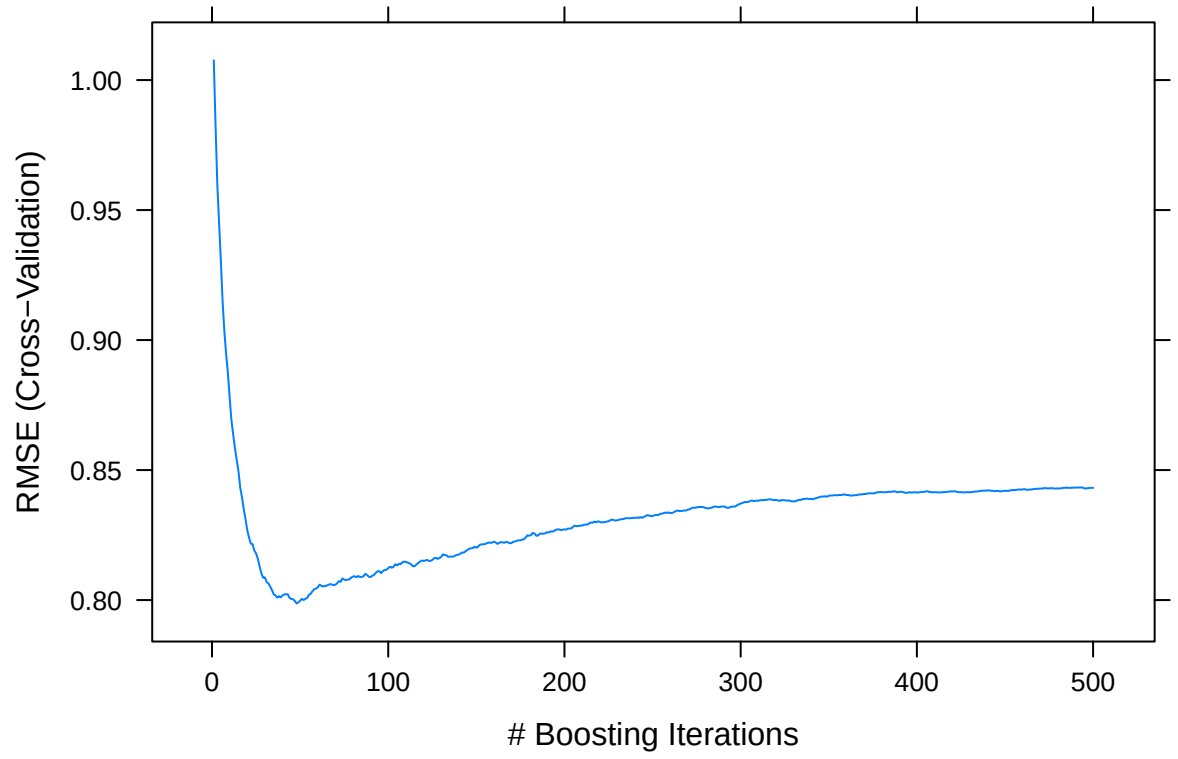
```
ncores <- 10

cl <- makePSOCKcluster(ncores)

registerDoParallel(cl)
```

```
# Grid Settings

grid <- expand.grid(shrinkage         = 0.1,
                    n.trees           = 1:500,
                    interaction.depth = 5,
```

```
                    n.minobsinnode    = 10)


gbm1 <- caret::train(blueprint,
                    data          = ind_tr,
                    method        = 'gbm',
                    trControl     = cv,
                    tuneGrid      = grid,
                    bag.fraction = 1,
                    verbose       = FALSE)

gbm1$times
```

```
## $everything
##    user  system elapsed
##   1.076   0.235  20.162
##
## $final
##    user  system elapsed
##   0.333   0.004   0.338
##
## $prediction
## [1] NA NA NA
```

```
plot(gbm1,type='l')
```



**Tune the interaction depth and n.minobsinnode**

```
gbm1$results[which.min(gbm1$results$RMSE),]
```

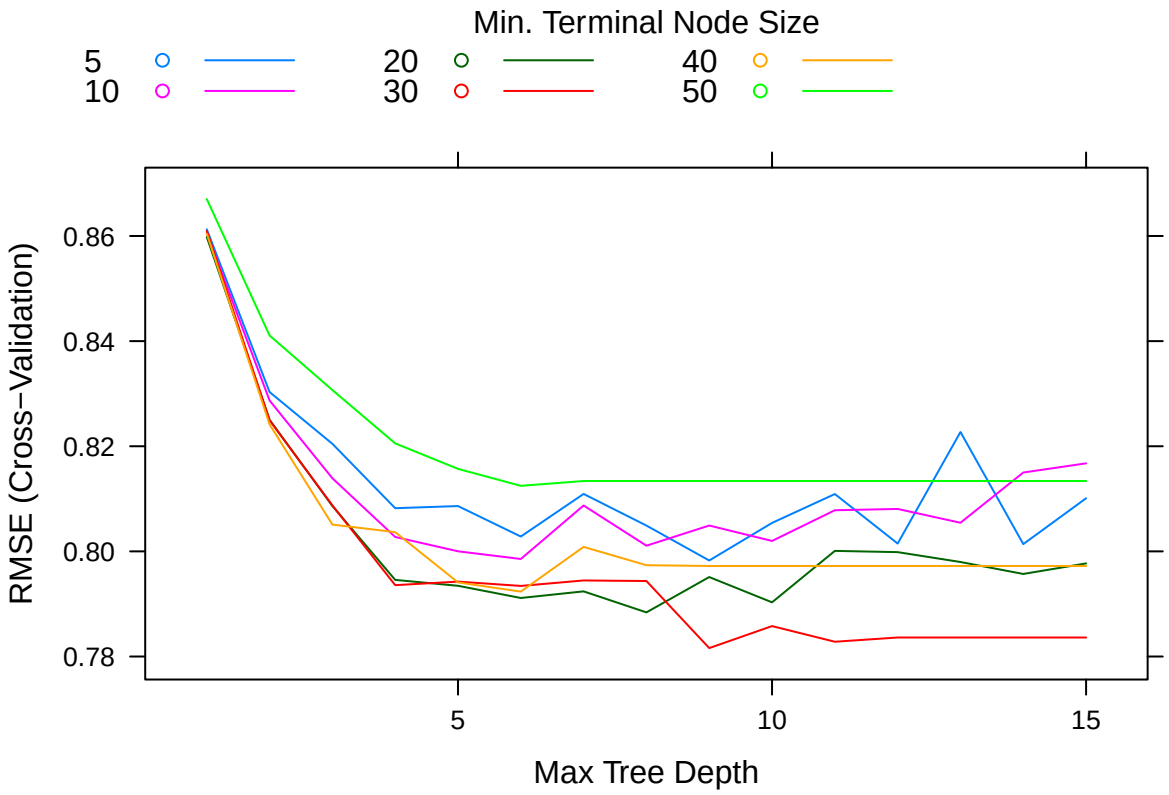
```
##    shrinkage interaction.depth n.minobsinnode n.trees      RMSE  Rsquared
## 48       0.1                 5             10      48 0.7986829 0.4283728
##          MAE    RMSESD RsquaredSD      MAESD
## 48 0.6228667 0.0633913  0.1093515 0.06315197
```

```
#We will fix the n.of trees to 52 onwards
```

```
grid <- expand.grid(shrinkage         = 0.1,
                    n.trees           = 52,
                    interaction.depth = 1:15,
                    n.minobsinnode    = c(5,10,20,30,40,50))


gbm2 <- caret::train(blueprint,
                     data      = ind_tr,
                     method    = 'gbm',
                     trControl = cv,
                     tuneGrid  = grid,
                     bag.fraction = 1,
                     verbose = FALSE)

gbm2$times
```

```
## $everything
##    user  system elapsed
##   1.487   1.036 202.338
##
## $final
##    user  system elapsed
##   0.445   0.007   0.452
##
## $prediction
## [1] NA NA NA
```

```
plot(gbm2,type='l')
```




```
gbm2$bestTune
```

```
##    n.trees interaction.depth shrinkage n.minobsinnode
## 52      52                 9       0.1             30
```

```
gbm2$results[which.min(gbm2$results$RMSE),]
```

```
##    shrinkage interaction.depth n.minobsinnode n.trees      RMSE  Rsquared
## 52       0.1                 9             30      52 0.7816028 0.4549474
##          MAE     RMSESD RsquaredSD      MAESD
## 52 0.6009909 0.07665079  0.1269779 0.07179077
```

```
predicted_te <- predict(gbm2,ind_te)

# MAE
mean(abs(ind_te$id_overall - predicted_te))
```

```
## [1] 0.5840144
```

```
# RMSE
sqrt(mean((ind_te$id_overall - predicted_te)^2))
```

```
## [1] 0.7309066
```

```r
# R-square
cor(ind_te$id_overall,predicted_te)^2
```

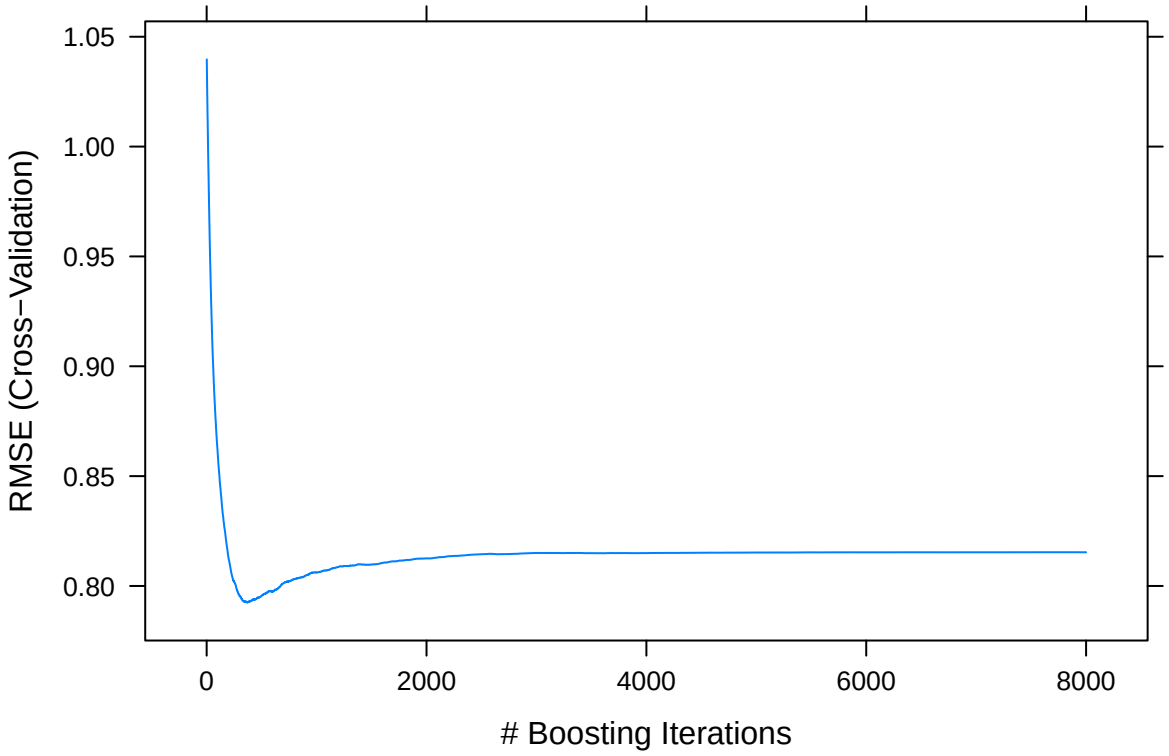
```
## [1] 0.4207848
```

**Tune in the n.trees to be 1 to 8000**

```r
grid <- expand.grid(shrinkage         = 0.01,
                    n.trees           = 1:8000,
                    interaction.depth = 13,
                    n.minobsinnode    = 20)


gbm3 <- caret::train(blueprint,
                     data      = ind_tr,
                     method    = 'gbm',
                     trControl = cv,
                     tuneGrid  = grid,
                     bag.fraction = 1,
                     verbose= FALSE)

gbm3$times
```

```
## $everything
##    user  system elapsed
##   4.445   0.246 149.137
##
## $final
##    user  system elapsed
##   2.642   0.017   2.667
##
## $prediction
## [1] NA NA NA
```

```r
plot(gbm3,type='l')
```




```
gbm3$results[which.min(gbm3$results$RMSE),]
```

```
##     shrinkage interaction.depth n.minobsinnode n.trees      RMSE  Rsquared
## 372      0.01                13             20     372 0.7924743 0.4385012
##           MAE     RMSESD RsquaredSD      MAESD
## 372 0.6164906 0.07177164  0.1237469 0.06140102
```

```
predicted_te <- predict(gbm3,ind_te)

# MAE
mean(abs(ind_te$id_overall - predicted_te))
```

```
## [1] 0.5944563
```

```
# RMSE
sqrt(mean((ind_te$id_overall - predicted_te)^2))
```

```
## [1] 0.7432849
```

```
# R-square
cor(ind_te$id_overall,predicted_te)^2
```

```
## [1] 0.4006313
```

```
perf <- matrix(c(0.434, 0.598,0.731,0.419,0.584,0.731,0.301,0.646,0.80),ncol=3,byrow=TRUE)
colnames(perf) <- c("R-Squared","MAE","RMSE")
rownames(perf) <- c("Random Forest","Gradient Boosting Tree", "Ridge Regularized Regression")
perf <- as.table(perf)
perf
```

```
##                               R-Squared   MAE  RMSE
## Random Forest                     0.434 0.598 0.731
## Gradient Boosting Tree            0.419 0.584 0.731
## Ridge Regularized Regression      0.301 0.646 0.800
```

```
vip(bags[[11]],num_features = 10, geom = "point") + theme_bw()
```