16.- Feature Selection_04_19_vacunacion_completo_v_01

June 8, 2023

#

CU04_Optimización de vacunas

Citizenlab Data Science Methodology > III - Feature Engineering Domain *** > # 16.- Feature Selection

Feature Selection is the process where you automatically or manually select the most relevant features which contribute most to the correct output of the model.

0.1 Tasks

Perform Selection of Categorical-Input/Categorical-Output
- Encoding-Categorical-Features - Chi-Squared-Feature-Selection - Mutual-Information-Feature-Selection - Evaluate-a-Logistic-Regression-model

Perform Selection of Numerical-Input/Categorical-Output
- ANOVA-F-test-Feature-Selection - Mutual-Information-Feature-Selection - Evaluating-a-Logistic-Regression-model - Tuning-the-Number-of-Selected-Features

Perform Selection of Numerical-Input/Numerical-Output
- Correlation-with-the-outcome-Feature-Selection - Mutual-Information-Feature-Selection - Evaluate-a-Lineal-Regression-model - Tuning-the-Number-of-Selected-Features

Perform Selection of Any-data
- RFE-(Recursive-Feature-Elimination) - Tuning-the-Number-of-Selected-Features - Automatically-Select-the-Number-of-Features

Explore the use of diferent algorithms wrapped by RFE

Explore the use od Hybrid feature selection algorithms

0.2 Consideraciones casos CitizenLab programados en R

- Algunas de las tareas de este proceso se han realizado en los notebooks del proceso 05 Data Collection porque eran necesarias para las tareas ETL. En esos casos, en este notebook se referencia al notebook del proceso 05 correspondiente
- Otras tareas típicas de este proceso se realizan en los notebooks del dominio IV al ser más eficiente realizarlas en el propio pipeline de modelización.
- Por tanto en los notebooks de este proceso de manera general se incluyen las comprobaciones necesarias, y comentarios si procede
- Las tareas del proceso se van a aplicar solo a los archivos que forman parte del despliegue, ya que hay muchos archivos intermedios que no procede pasar por este proceso

- El nombre de archivo del notebook hace referencia al nombre de archivo del proceso 05 al que se aplica este proceso, por eso pueden no ser correlativa la numeración
- Las comprobaciones se van a realizar teniendo en cuenta que el lenguaje utilizado en el despliegue de este caso es R

0.3 File

- Input File: CU_04_08_20_vacunacion_gripe_train_and_test.csv
- Output File: No aplica

0.3.1 Encoding

Con la siguiente expresión se evitan problemas con el encoding al ejecutar el notebook. Es posible que deba ser eliminada o adaptada a la máquina en la que se ejecute el código.

[1]:
```
Sys.setlocale(category = "LC_ALL", locale = "es_ES.UTF-8")
```

```
Warning message in Sys.setlocale(category = "LC_ALL", locale = "es_ES.UTF-8"):
"OS reports request to set locale to "es_ES.UTF-8" cannot be honored"
"
```

0.4 Settings

0.4.1 Libraries to use

[9]:
```
library(caret)
library(readr)
library(dplyr)
library(tidyr)
library(forcats)
library(lubridate)
```

```
Loading required package: ggplot2

Loading required package: lattice
```

0.4.2 Paths

[3]:
```
iPath <- "Data/Input/"
oPath <- "Data/Output/"
```

0.5 Data Load

OPCION A: Seleccionar fichero en ventana para mayor comodidad

Data load using the {tcltk} package. Ucomment the line if using this option

[4]:
```
# file_data <- tcltk::tk_choose.files(multi = FALSE)
```

OPCION B: Especificar el nombre de archivo

[28]:
```
iFile <- "CU_04_08_20_vacunacion_gripe_train_and_test.csv"
file_data <- paste0(iPath, iFile)

if(file.exists(file_data)){
    cat("Se leerán datos del archivo: ", file_data)
} else{
    warning("Cuidado: el archivo no existe.")
}
```

```
Se leerán datos del archivo:
Data/Input/CU_04_08_20_vacunacion_gripe_train_and_test.csv
```

Data file to dataframe Usar la función adecuada según el formato de entrada (xlsx, csv, json, ...)

[29]:
```
data <- read_csv(file_data)
```

```
Rows: 21736 Columns: 49
  Column specification



Delimiter: ","
chr  (3): GEOCODIGO, DESBDT, nombre_zona
dbl (45): ano, semana, n_vacunas, n_citas, tmed, prec, velmedia,
presMax, be…
lgl  (1): is_train

 Use `spec()` to retrieve the full column specification for this
data.
 Specify the column types or set `show_col_types = FALSE` to quiet
this message.
```

Estructura de los datos:

[30]:
```
data |> glimpse()
```

```
Rows: 21,736
Columns: 49
$ GEOCODIGO         <chr> "259", "260", "041", "025", "046",
"159", "065", "09…
$ DESBDT            <chr> "V Centenario", "Valdeacederas",
"Canillejas", "Bara…
$ ano               <dbl> 2022, 2022, 2022, 2022, 2022, 2022,
2022, 2021, 2023…
$ semana            <dbl> 34, 8, 9, 49, 24, 3, 8, 47, 1, 2,
52, 39, 16, 50, 34…
```

```
$ n_vacunas         <dbl> 0, 0, 0, 292, 0, 524, 0, 248, 204,
205, NA, 0, 0, 51…
$ n_citas           <dbl> 0, 0, 0, 280, 0, 498, 0, 228, 198,
187, NA, 0, 0, 51…
$ tmed              <dbl> 27.278748, 9.577289, 8.536554,
9.065363, 29.905728, …
$ prec              <dbl> 0.169955881, 1.264910043,
3.122881160, 7.313886680, …
$ velmedia          <dbl> 2.297067, 1.890425, 2.418071,
1.562328, 2.564749, 1.…
$ presMax           <dbl> 940.0420, 944.1770, 949.7179,
941.8342, 940.5669, 95…
$ benzene           <dbl> 0.1764413, 0.4591543, 0.4099159,
0.4224172, 0.195865…
$ co                <dbl> 0.4987735, 0.3960647, 0.3951587,
NA, 0.2891224, 0.50…
$ no                <dbl> NA, 6.611337, 9.331224, 14.007722,
4.063517, 24.4756…
$ no2               <dbl> 14.21113, 34.67671, 30.29999,
32.54832, 26.06913, 44…
$ nox               <dbl> 18.00109, 48.94660, 45.22346,
56.75574, 30.35311, 74…
$ o3                <dbl> 80.90659, 42.06663, 48.88088,
26.68276, 64.55205, 31…
$ pm10              <dbl> 20.117087, 15.042152, 14.002432,
18.032354, 55.79346…
$ pm2.5             <dbl> 10.628064, 5.539590, 7.124192,
6.793868, 19.520373, …
$ so2               <dbl> 2.794934, 3.507164, 2.692125,
2.351139, 3.397640, 2.…
$ campana           <dbl> NA, NA, NA, 2022, NA, 2021, NA,
2021, 2022, 2021, 20…
$ scampana          <dbl> NA, NA, NA, 14, NA, 20, NA, 12, 18,
19, 17, 4, NA, 1…
$ capacidad_zona    <dbl> 7957, 6537, 7167, 5633, 3864,
12583, 8544, 5077, 494…
$ prop_riesgo       <dbl> 0.11393237, 0.15763986, 0.25500690,
0.14452370, 0.26…
$ tasa_riesgo       <dbl> 0.013477754, 0.015731142,
0.009177382, 0.013099129, …
$ tasa_mayores      <dbl> 0.023033610, 0.032817374,
0.028147027, 0.020829657, …
$ poblacion_mayores <dbl> 0.10330662, 0.14362062, 0.23161874,
0.13058449, 0.24…
$ nombre_zona       <chr> "V Centenario", "Valdeacederas",
"Canillejas", "Bara…
$ nsec              <dbl> 17, 18, 22, 13, 14, 42, 32, 13, 17,
11, NA, 15, 15, …
```

```
$ t3_1              <dbl> 36.73039, 41.41412, 45.44882,
39.78001, 46.13171, 46…
$ t1_1              <dbl> 31778, 26202, 28658, 22492, 15450,
50478, 34148, 202…
$ t2_1              <dbl> 0.5084658, 0.5329728, 0.5316594,
0.5189021, 0.551191…
$ t2_2              <dbl> 0.4915342, 0.4670272, 0.4683406,
0.4810979, 0.448809…
$ t4_1              <dbl> 0.22551283, 0.12790298, 0.12603707,
0.18104432, 0.11…
$ t4_2              <dbl> 0.6711962, 0.7284970, 0.6423306,
0.6883785, 0.641173…
$ t4_3              <dbl> 0.10330662, 0.14362062, 0.23161874,
0.13058449, 0.24…
$ t5_1              <dbl> 0.1063332, 0.2295250, 0.1655070,
0.1266086, 0.165893…
$ t6_1              <dbl> 0.1706875, 0.3477631, 0.2511757,
0.1998911, 0.261480…
$ t7_1              <dbl> 0.05131106, 0.04606911, 0.04379644,
0.05585777, 0.06…
$ t8_1              <dbl> 0.03892836, 0.03586418, 0.03207779,
0.04434976, 0.05…
$ t9_1              <dbl> 0.5151383, 0.3863876, 0.3129631,
0.4611972, 0.701812…
$ t10_1             <dbl> 0.09258503, 0.13151901, 0.13926119,
0.10460043, 0.06…
$ t11_1             <dbl> 0.6406787, 0.5451465, 0.4600730,
0.5920292, 0.471769…
$ t12_1             <dbl> 0.7028586, 0.6277335, 0.5346482,
0.6590530, 0.502531…
$ area              <dbl> 2100118.9, 1164622.0, 1597474.5,
3816572.0, 870986.8…
$ densidad_hab_km   <dbl> 15131.52443, 22498.28643,
17939.56640, 5893.24662, 1…
$ tuits_gripe       <dbl> 60, 56, 72, 196, 46, 382, 56, 280,
24, 508, NA, 126,…
$ interes_gripe     <dbl> 24, 15, 24, 77, 21, 42, 15, 64, 64,
69, NA, 42, 40, …
$ Target            <dbl> 0, 0, 0, 292, 0, 524, 0, 248, 204,
205, NA, 0, 0, 51…
$ is_train          <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE,
TRUE, TRUE, TRUE…
```

Muestra de los primeros datos:

[31]:
```
data |> slice_head(n = 5)
```

	GEOCODIGO <chr>	DESBDT <chr>	ano <dbl>	semana <dbl>	n_vacunas <dbl>	n_citas <dbl>	tmed <dbl>	p
	259	V Centenario	2022	34	0	0	27.278748	0
A spec_tbl_df: 5 × 49	260	Valdeacederas	2022	8	0	0	9.577289	1
	041	Canillejas	2022	9	0	0	8.536554	3
	025	Barajas	2022	49	292	280	9.065363	7
	046	Castelló	2022	24	0	0	29.905728	0

0.6 Selecting Categorical Input / Categorical Output

0.6.1 Encoding Categorical Features

[61]:
```
# Convert all character columns to factors
data <- mutate_if(data, is.character, as.factor)
data <- na.omit(data)

train_set <- subset(data[data$is_train == TRUE, ], select = -is_train)
train_set <- select_if(train_set, is.numeric)
test_set <- subset(data[data$is_train == FALSE, ], select = -is_train)
test_set <- select_if(test_set, is.numeric)
```

[ ]:
```
```

0.6.2 Chi-Squared Feature Selection

No aplica ya que el Target no es categórico.

[62]:
```
# # Set the threshold
# threshold = 0.05

# # Create an empty vector to store p-values
# p_values <- sapply(train_data[ , -which(names(train_data) == "Target")],
 ↪function(x) {
#   chisq <- chisq.test(table(x, train_data$Target))
#   return(chisq$p.value)
# })

# # Select variables with p-values less than a certain threshold (for example,
 ↪0.05)
# selected_features <- names(p_values)[p_values < threshold]

# # Print selected features
# print(selected_features)
```

0.6.3 Mutual Information Feature Selection

No aplica ya que el Target no es categórico.

[63]:
```
# # install the necessary packages if not already installed
# if (!require(FSelectorRcpp)) {
#   install.packages('FSelectorRcpp')
# }

# # Load necessary library
# library(FSelectorRcpp)

# # Calculate mutual information between each variable and the target
# mi_scores <- information_gain(train_data[, setdiff(names(train_data),␣
 ↪"Target")], train_data$Target)

# # Convert the top_features object into a dataframe
# mi_scores_df <- as.data.frame(mi_scores)

# # Rename the columns
# names(mi_scores_df) <- c("Feature", "Score")

# # Order the dataframe by Score in descending order
# mi_scores_df <- mi_scores_df[order(-mi_scores_df$Score),]

# # Create a bar plot
# ggplot(mi_scores_df, aes(x = reorder(Feature, Score), y = Score)) +
#   geom_bar(stat = "identity", fill = "steelblue") +
#   coord_flip() +
#   xlab("Features") +
#   ylab("Mutual Information Score") +
#   ggtitle("Top Features by Mutual Information") +
#   theme_minimal()
```

0.6.4 Evaluating a Logistic Regression model

No aplica ya que el Target no es categórico.

Select numer of Features to use

[64]:
```
# Select numer of Features to use
k <- 5
```

Operation

[65]:
```
# # Fit a linear regression model
# model_all_features <- glm(Target ~ ., data = select_if(train_set, is.numeric))

# # Predict on the test set
# predictions <- predict(model_all_features, newdata = test_set)

# # Evaluate the model
```

```
# postResample(pred = predictions, obs = test_set$Target)
```

[66]:
```
# # Select the top k features
# top_features <- names(top_features)[1:k]

# # Fit a linear regression model with only the top k features
# model_top_features <- glm(Target ~ ., data = select_if(train_set, is.
 ↪numeric)[, c(top_features, "Target")])

# # Predict on the test set
# predictions_top_features <- predict(model_top_features, newdata = test_set[,␣
 ↪top_features])

# # Evaluate the model
# postResample(pred = predictions_top_features, obs = test_set$Target)
```

0.7 Selecting Numerical Input / Categorical Output

No aplica ya que el Target no es categórico.

Operation

0.7.1 ANOVA F-test Feature Selection

0.7.2 Mutual Information Feature Selection

0.7.3 Evaluating a Logistic Regression model

Selecting feature to use

[67]:
```
# Select numer of Features to use
```

Operation

0.7.4 Tuning the Number of Selected Features

[ ]:
```

```

Know the best number of features to select

[ ]:
```

```

See the relationship between the number of selected features and accuracy

[ ]:
```

```

0.8 Selecting Numerical Input / Numerical Output

[ ]:

0.8.1 Correlation with the outcome Feature Selection

[68]:
```
# Calculate the correlation between each feature and the outcome variable
correlations <- sapply(select_if(train_set, is.numeric)[,
 ↪-which(names(select_if(train_set, is.numeric)) %in% "Target")], function(x)
 ↪cor(x, train_set$Target))

# Create a dataframe from the correlations
correlation_df <- data.frame(Feature = names(correlations), Correlation =
 ↪correlations)

# Sort the dataframe by the absolute values of the correlations in descending
 ↪order
correlation_df <- correlation_df[order(-abs(correlation_df$Correlation)), ]

# Print the correlation dataframe
print(correlation_df)
```

```
                         Feature  Correlation
n_vacunas                n_vacunas  1.000000000
n_citas                    n_citas  0.999130400
interes_gripe        interes_gripe  0.586498046
tmed                          tmed -0.585515629
scampana                scampana  0.539384124
capacidad_zona      capacidad_zona  0.518963147
t1_1                          t1_1  0.518958470
o3                              o3 -0.515131533
nsec                          nsec  0.415394240
benzene                    benzene  0.403689076
tuits_gripe            tuits_gripe  0.375181698
nox                            nox  0.282164710
no                              no  0.272879071
no2                            no2  0.259185375
so2                            so2 -0.201351922
t11_1                        t11_1  0.162866925
t3_1                          t3_1 -0.136262423
co                              co  0.135689359
t12_1                        t12_1  0.135599031
pm10                          pm10 -0.130991898
t4_1                          t4_1  0.130901447
t10_1                        t10_1 -0.130502874
t9_1                          t9_1  0.120369455
campana                    campana -0.115730085
```

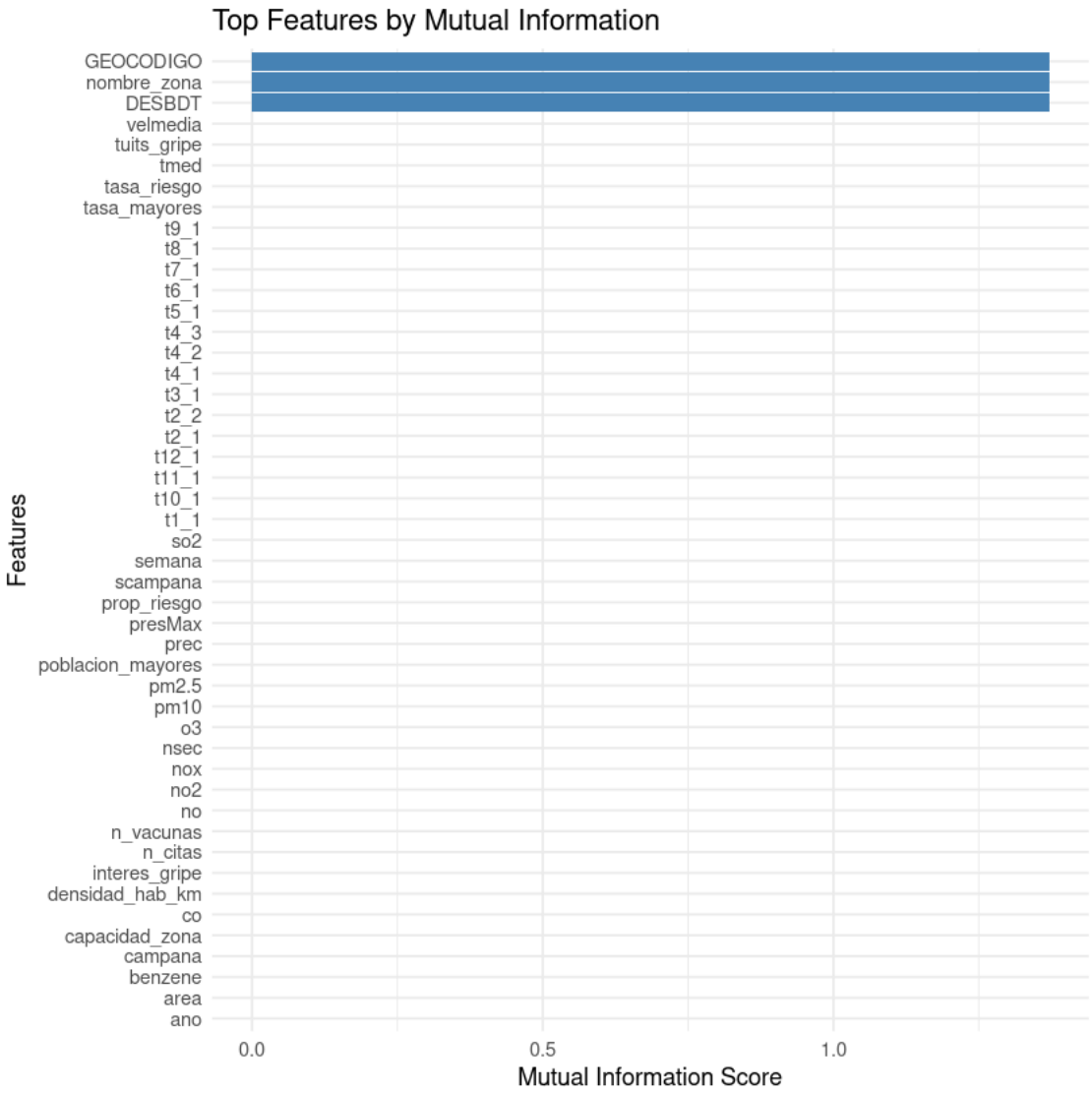
```
presMax                    presMax  0.108626452
prop_riesgo            prop_riesgo -0.105426121
poblacion_mayores poblacion_mayores -0.105092360
t4_3                          t4_3 -0.105092360
t8_1                          t8_1  0.088204743
t2_1                          t2_1  0.080789290
t7_1                          t7_1  0.078949266
area                          area -0.072657104
prec                          prec  0.068205501
t2_2                          t2_2 -0.066609878
semana                      semana  0.060823010
t5_1                          t5_1 -0.055804528
velmedia                  velmedia  0.055028022
tasa_riesgo            tasa_riesgo  0.041871987
t4_2                          t4_2  0.030293108
t6_1                          t6_1 -0.023654562
ano                            ano -0.015855092
pm2.5                        pm2.5 -0.009172425
densidad_hab_km  densidad_hab_km  0.004522135
tasa_mayores        tasa_mayores  0.002162538
```

0.8.2 Mutual Information Feature Selection

[69]:
```r
# install the necessary packages if not already installed
if (!require(FSelectorRcpp)) {
  install.packages('FSelectorRcpp')
}

# Load necessary library
library(FSelectorRcpp)

# Calculate mutual information between each variable and the target
mi_scores <- information_gain(train_data[, setdiff(names(train_data),␣
 ↪"Target")], train_data$Target)

# Convert the top_features object into a dataframe
mi_scores_df <- as.data.frame(mi_scores)

# Rename the columns
names(mi_scores_df) <- c("Feature", "Score")

# Order the dataframe by Score in descending order
mi_scores_df <- mi_scores_df[order(-mi_scores_df$Score),]

# Create a bar plot
ggplot(mi_scores_df, aes(x = reorder(Feature, Score), y = Score)) +
  geom_bar(stat = "identity", fill = "steelblue") +
```

```
  coord_flip() +
  xlab("Features") +
  ylab("Mutual Information Score") +
  ggtitle("Top Features by Mutual Information") +
  theme_minimal()
```

```
Warning message in .information_gain.data.frame(formula, data, type = type,
equal = equal, :
"There are missing values in the dependent variable information_gain will remove
them."
Warning message in .information_gain.data.frame(formula, data, type = type,
equal = equal, :
"Dependent variable is a numeric! It will be converted to factor with simple
factor(y). We do not discretize dependent variable in FSelectorRcpp by default!
You can choose equal frequency binning discretization by setting equal argument
to TRUE."
```



0.8.3 Evaluating a Lineal Regression model

[ ]:

Selecting feature to use

[70]:
```
# Select numer of Features to use
k <- 5
```

Operation

[71]:
```
# Fit a linear regression model
model_all_features <- lm(Target ~ ., data = train_set)
```

```r
# Predict on the test set
predictions <- predict(model_all_features, newdata = test_set)

# Evaluate the model
postResample(pred = predictions, obs = test_set$Target)
```

```
Warning message in predict.lm(model_all_features, newdata = test_set):
"prediction from a rank-deficient fit may be misleading"
```

RMSE 8.93086012220612e-13 **Rsquared** 1 **MAE** 7.18003164589858e-13

[72]:
```r
# Select the top k features
top_features <- names(mi_scores)[1:k]

# Fit a linear regression model with only the top k features
model_top_features <- lm(Target ~ ., data = train_set[, c(top_features,␣
 ↪"Target")])

# Predict on the test set
predictions_top_features <- predict(model_top_features, newdata = test_set[,␣
 ↪top_features])

# Evaluate the model
postResample(pred = predictions_top_features, obs = test_set$Target)
```

```
Error in `train_set[, c(top_features, "Target")]`:
! Can't subset columns that don't exist.
  Columns NA don't exist.
Traceback:

1. lm(Target ~ ., data = train_set[, c(top_features, "Target")])
2. eval(mf, parent.frame())
3. eval(mf, parent.frame())
4. stats::model.frame(formula = Target ~ ., data = train_set[, c(top_features,
 .     "Target")], drop.unused.levels = TRUE)
5. model.frame.default(formula = Target ~ ., data = train_set[,
 .     c(top_features, "Target")], drop.unused.levels = TRUE)
6. is.data.frame(data)
7. train_set[, c(top_features, "Target")]
8. `[.tbl_df`(train_set, , c(top_features, "Target"))
9. vectbl_as_col_location(j, length(x), names(x), j_arg = j_arg,
 .     assign = FALSE)
10. subclass_col_index_errors(vec_as_location(j, n, names, missing = "error",
  .     call = call), j_arg = j_arg, assign = assign)
11. withCallingHandlers(expr, vctrs_error_subscript = function(cnd) {
  .     cnd$subscript_arg <- j_arg
  .     cnd$subscript_elt <- "column"
```

```
  .     if (isTRUE(assign) && !isTRUE(cnd$subscript_action %in% c("negate"))) {
  .         cnd$subscript_action <- "assign"
  .     }
  .     cnd_signal(cnd)
  . })
12. vec_as_location(j, n, names, missing = "error", call = call)
13. (function ()
  . stop_subscript_oob(i = i, subscript_type = subscript_type, names = names,
  .     subscript_action = subscript_action, subscript_arg = subscript_arg,
  .     call = call))()
14. stop_subscript_oob(i = i, subscript_type = subscript_type, names = names,
  .     subscript_action = subscript_action, subscript_arg = subscript_arg,
  .     call = call)
15. stop_subscript(class = "vctrs_error_subscript_oob", i = i, subscript_type =
 ↪subscript_type,
  .     …, call = call)
16. abort(class = c(class, "vctrs_error_subscript"), i = i, …,
  .     call = call)
17. signal_abort(cnd, .file)
18. signalCondition(cnd)
19. (function (cnd)
  . {
  .     cnd$subscript_arg <- j_arg
  .     cnd$subscript_elt <- "column"
  .     if (isTRUE(assign) && !isTRUE(cnd$subscript_action %in% c("negate"))) {
  .         cnd$subscript_action <- "assign"
  .     }
  .     cnd_signal(cnd)
  . })(structure(list(message = "", trace = structure(list(call = list(
  .     IRkernel::main(), kernel$run(), handle_shell(), executor$execute(msg),
  .     tryCatch(evaluate(request$content$code, envir = .GlobalEnv,
  .         output_handler = oh, stop_on_error = 1L), interrupt = function(cond
 ↪{
  .         log_debug("Interrupt during execution")
  .         interrupted <<- TRUE
  .     }, error = .self$handle_error), tryCatchList(expr, classes,
  .         parentenv, handlers), tryCatchOne(tryCatchList(expr,
  .         names[-nh], parentenv, handlers[-nh]), names[nh], parentenv,
  .         handlers[[nh]]), doTryCatch(return(expr), name, parentenv,
  .         handler), tryCatchList(expr, names[-nh], parentenv, handlers[-nh]),
  .     tryCatchOne(expr, names, parentenv, handlers[[1L]]),
 ↪doTryCatch(return(expr),
  .         name, parentenv, handler), evaluate(request$content$code,
  .         envir = .GlobalEnv, output_handler = oh, stop_on_error = 1L),
  .     evaluate_call(expr, parsed$src[[i]], envir = envir, enclos = enclos,
  .         debug = debug, last = i == length(out), use_try = stop_on_error !=
  .             2L, keep_warning = keep_warning, keep_message = keep_message,
```

```
.         log_echo = log_echo, log_warning = log_warning, output_handler =␣
↪output_handler,
.         include_timing = include_timing), timing_fn(handle(ev <-␣
↪withCallingHandlers(withVisible(eval_with_user_handlers(expr,
.         envir, enclos, user_handlers)), warning = wHandler, error = eHandle ,
.         message = mHandler))), handle(ev <-␣
↪withCallingHandlers(withVisible(eval_with_user_handlers(expr,
.         envir, enclos, user_handlers)), warning = wHandler, error = eHandle ,
.         message = mHandler)), try(f, silent = TRUE), tryCatch(expr,
.         error = function(e) {
.             call <- conditionCall(e)
.             if (!is.null(call)) {
.                 if (identical(call[[1L]], quote(doTryCatch)))
.                   call <- sys.call(-4L)
.                 dcall <- deparse(call, nlines = 1L)
.                 prefix <- paste("Error in", dcall, ": ")
.                 LONG <- 75L
.                 sm <- strsplit(conditionMessage(e), "\n")[[1L]]
.                 w <- 14L + nchar(dcall, type = "w") + nchar(sm[1L],
.                   type = "w")
.                 if (is.na(w))
.                   w <- 14L + nchar(dcall, type = "b") + nchar(sm[1L],
.                     type = "b")
.                 if (w > LONG)
.                   prefix <- paste0(prefix, "\n  ")
.             }
.             else prefix <- "Error : "
.             msg <- paste0(prefix, conditionMessage(e), "\n")
.             .Internal(seterrmessage(msg[1L]))
.             if (!silent && isTRUE(getOption("show.error.messages"))) {
.                 cat(msg, file = outFile)
.                 .Internal(printDeferredWarnings())
.             }
.             invisible(structure(msg, class = "try-error", condition = e))
.         }), tryCatchList(expr, classes, parentenv, handlers),
.     tryCatchOne(expr, names, parentenv, handlers[[1L]]),␣
↪doTryCatch(return(expr),
.         name, parentenv, handler),␣
↪withCallingHandlers(withVisible(eval_with_user_handlers(expr,
.         envir, enclos, user_handlers)), warning = wHandler, error = eHandle ,
.         message = mHandler), withVisible(eval_with_user_handlers(expr,
.         envir, enclos, user_handlers)), eval_with_user_handlers(expr,
.         envir, enclos, user_handlers), eval(expr, envir, enclos),
.     eval(expr, envir, enclos), lm(Target ~ ., data = train_set[,
.         c(top_features, "Target")]), eval(mf, parent.frame()),
.     eval(mf, parent.frame()), stats::model.frame(formula = Target ~
.         ., data = train_set[, c(top_features, "Target")], drop.unused.level ␣
↪= TRUE),
```

```
 .     model.frame.default(formula = Target ~ ., data = train_set[,
 .         c(top_features, "Target")], drop.unused.levels = TRUE),
 .     is.data.frame(data), train_set[, c(top_features, "Target")],
 .     `[.tbl_df`(train_set, , c(top_features, "Target")),␣
↪vectbl_as_col_location(j,
 .         length(x), names(x), j_arg = j_arg, assign = FALSE),
 .     subclass_col_index_errors(vec_as_location(j, n, names, missing = "error ,
 .         call = call), j_arg = j_arg, assign = assign),␣
↪withCallingHandlers(expr,
 .         vctrs_error_subscript = function(cnd) {
 .             cnd$subscript_arg <- j_arg
 .             cnd$subscript_elt <- "column"
 .             if (isTRUE(assign) && !isTRUE(cnd$subscript_action %in%
 .                 c("negate"))) {
 .                 cnd$subscript_action <- "assign"
 .             }
 .             cnd_signal(cnd)
 .         }), vec_as_location(j, n, names, missing = "error", call = call),
 .     `<fn>`(), stop_subscript_oob(i = i, subscript_type = subscript_type,
 .         names = names, subscript_action = subscript_action, subscript_arg =
↪subscript_arg,
 .         call = call), stop_subscript(class = "vctrs_error_subscript_oob",
 .         i = i, subscript_type = subscript_type, …, call = call),
 .     abort(class = c(class, "vctrs_error_subscript"), i = i, …,
 .         call = call)), parent = c(0L, 1L, 2L, 3L, 4L, 5L, 6L,
 . 7L, 6L, 9L, 10L, 4L, 12L, 13L, 13L, 15L, 16L, 17L, 18L, 19L,
 . 13L, 13L, 13L, 23L, 24L, 0L, 26L, 27L, 0L, 0L, 30L, 0L, 0L, 33L,
 . 34L, 35L, 34L, 0L, 38L, 39L, 40L), visible = c(TRUE, TRUE, TRUE,
 . TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE,
 . TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE,
 . TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, FALSE, FALSE,
 . FALSE, FALSE, FALSE, FALSE, FALSE, FALSE), namespace = c("IRkernel",
 . NA, "IRkernel", NA, "base", "base", "base", "base", "base", "base",
 . "base", "evaluate", "evaluate", "evaluate", "evaluate", "base",
 . "base", "base", "base", "base", "base", "base", "evaluate", "base",
 . "base", "stats", "base", "base", "stats", "stats", "base", NA,
 . "tibble", "tibble", "tibble", "base", "vctrs", "vctrs", "vctrs",
 . "vctrs", "rlang"), scope = c("::", NA, "local", NA, "::", "local",
 . "local", "local", "local", "local", "local", "::", ":::", "local",
 . "local", "::", "::", "local", "local", "local", "::", "::", ":::",
 . "::", "::", "::", "::", "::", "::", "::", "::", NA, ":::", ":::",
 . ":::", "::", "::", "local", ":::", ":::", "::"), error_frame = c(FALSE,
 . FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE,
 . FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE,
 . FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE,
 . FALSE, FALSE, FALSE, FALSE, TRUE, FALSE, FALSE, FALSE, FALSE,
 . FALSE, FALSE, FALSE, FALSE)), row.names = c(NA, -41L), version = 2L, class =␣
↪c("rlang_trace",
```

```
  . "rlib_trace", "tbl", "data.frame")), parent = NULL, i = c("attributes",
  . "importance", NA, NA, NA, "Target"), subscript_type = "character",
  .     names = c("ano", "semana", "n_vacunas", "n_citas", "tmed",
  .     "prec", "velmedia", "presMax", "benzene", "co", "no", "no2",
  .     "nox", "o3", "pm10", "pm2.5", "so2", "campana", "scampana",
  .     "capacidad_zona", "prop_riesgo", "tasa_riesgo", "tasa_mayores",
  .     "poblacion_mayores", "nsec", "t3_1", "t1_1", "t2_1", "t2_2",
  .     "t4_1", "t4_2", "t4_3", "t5_1", "t6_1", "t7_1", "t8_1", "t9_1",
  .     "t10_1", "t11_1", "t12_1", "area", "densidad_hab_km", "tuits_gripe",
  .     "interes_gripe", "Target"), subscript_action = NULL, subscript_arg = "j",
  .     rlang = list(inherit = TRUE), call = train_set[, c(top_features,
  .         "Target")]), class = c("vctrs_error_subscript_oob",␣
 ↪"vctrs_error_subscript",
  . "rlang_error", "error", "condition")))
20. cnd_signal(cnd)
21. signal_abort(cnd)
```

0.9 Any data: RFE (Recursive Feature Elimination)

0.9.1 RFE for Classification

No aplica ya que el Target no es categórico.

Selecting feature to use

[ ]:
```
# Select numer of Features to use
```

Operation

[ ]:
```

```

0.9.2 RFE for Regression

Selecting feature to use

[73]:
```
# Select numer of Features to use
k <- 5
```

Operation

[74]:
```
# Define control parameters for rfe function
ctrl <- rfeControl(functions=lmFuncs, method="cv", number=10)

# Determine number of predictors
predictors_number <- ncol(train_set) - 1 # Assuming the last column is the␣
 ↪target variable

# Apply the RFE algorithm with cross validation.
```

```r
result <- rfe(train_set[, !names(train_set) %in% "Target"], train_set$Target,
  sizes=c(1:predictors_number), rfeControl=ctrl)

# Print the result
print(result)

# Top ranking variables in the optimal subset size
top_features <- predictors(result, result$optsize)
```

```
Warning message in predict.lm(object, x):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(object, x):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(object, x):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(object, x):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(object, x):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(object, x):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(object, x):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(object, x):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(object, x):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(object, x):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(object, x):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(object, x):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(object, x):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(object, x):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(object, x):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(object, x):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(object, x):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(object, x):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(object, x):
"prediction from a rank-deficient fit may be misleading"
```

```
Warning message in predict.lm(object, x):
"prediction from a rank-deficient fit may be misleading"


Recursive feature selection

Outer resampling method: Cross-Validated (10 fold)

Resampling performance over subset size:

 Variables      RMSE Rsquared       MAE    RMSESD RsquaredSD     MAESD Selected
         1 3.308e-12        1 2.705e-12 4.069e-12          0 3.326e-12
         2 3.307e-12        1 2.706e-12 4.065e-12          0 3.326e-12
         3 3.306e-12        1 2.703e-12 4.070e-12          0 3.328e-12
         4 3.306e-12        1 2.702e-12 4.069e-12          0 3.327e-12
         5 3.304e-12        1 2.703e-12 4.062e-12          0 3.320e-12
         6 3.306e-12        1 2.705e-12 4.071e-12          0 3.328e-12
         7 3.308e-12        1 2.704e-12 4.071e-12          0 3.328e-12
         8 3.306e-12        1 2.703e-12 4.077e-12          0 3.334e-12
         9 3.303e-12        1 2.700e-12 4.066e-12          0 3.325e-12
        10 3.305e-12        1 2.702e-12 4.068e-12          0 3.326e-12
        11 3.299e-12        1 2.698e-12 4.053e-12          0 3.314e-12
        12 3.315e-12        1 2.711e-12 4.066e-12          0 3.326e-12
        13 3.322e-12        1 2.716e-12 4.064e-12          0 3.325e-12
        14 3.308e-12        1 2.707e-12 4.066e-12          0 3.326e-12
        15 3.320e-12        1 2.717e-12 4.062e-12          0 3.321e-12
        16 3.309e-12        1 2.707e-12 4.080e-12          0 3.338e-12
        17 3.313e-12        1 2.712e-12 4.076e-12          0 3.333e-12
        18 3.312e-12        1 2.709e-12 4.061e-12          0 3.324e-12
        19 3.316e-12        1 2.713e-12 4.075e-12          0 3.339e-12
        20 3.321e-12        1 2.718e-12 4.079e-12          0 3.341e-12
        21 3.311e-12        1 2.710e-12 4.067e-12          0 3.332e-12
        22 3.309e-12        1 2.708e-12 4.057e-12          0 3.324e-12
        23 3.326e-12        1 2.724e-12 4.073e-12          0 3.339e-12
        24 3.317e-12        1 2.717e-12 4.054e-12          0 3.324e-12
        25 3.318e-12        1 2.719e-12 4.061e-12          0 3.331e-12
        26 3.327e-12        1 2.722e-12 4.077e-12          0 3.342e-12
        27 3.319e-12        1 2.719e-12 4.053e-12          0 3.322e-12
        28 3.317e-12        1 2.719e-12 4.047e-12          0 3.317e-12
        29 3.322e-12        1 2.722e-12 4.042e-12          0 3.315e-12
        30 3.312e-12        1 2.715e-12 4.047e-12          0 3.319e-12
        31 3.309e-12        1 2.709e-12 4.061e-12          0 3.333e-12
        32 3.320e-12        1 2.717e-12 4.054e-12          0 3.326e-12
        33 3.323e-12        1 2.720e-12 4.054e-12          0 3.324e-12
        34 3.308e-12        1 2.708e-12 4.053e-12          0 3.322e-12
        35 3.320e-12        1 2.717e-12 4.058e-12          0 3.325e-12
        36 3.316e-12        1 2.711e-12 4.060e-12          0 3.327e-12
        37 3.315e-12        1 2.712e-12 4.054e-12          0 3.322e-12
```

```
          38 3.317e-12        1 2.715e-12 4.052e-12          0 3.320e-12
          39 3.314e-12        1 2.710e-12 4.053e-12          0 3.320e-12
          40 3.323e-12        1 2.719e-12 4.048e-12          0 3.315e-12
          41 3.323e-12        1 2.722e-12 4.051e-12          0 3.320e-12
          42 3.312e-12        1 2.712e-12 4.053e-12          0 3.320e-12
          43 3.312e-12        1 2.712e-12 4.053e-12          0 3.320e-12
          44 1.302e-12        1 1.060e-12 1.018e-12          0 8.380e-13        *

    The top 5 variables (out of 44):
       n_vacunas, t2_1, t4_2, t2_2, t4_1
```

[ ]:

[ ]: