Classifiers

```r
set.seed(42)

library(caret)
```

```
## Loading required package: ggplot2
## Loading required package: lattice
```

```r
library(party)
```

```
## Loading required package: grid
## Loading required package: mvtnorm
## Loading required package: modeltools
## Loading required package: stats4
## Loading required package: strucchange
## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
## Loading required package: sandwich
```

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x stringr::boundary() masks strucchange::boundary()
## x dplyr::filter()     masks stats::filter()
## x dplyr::lag()        masks stats::lag()
## x purrr::lift()       masks caret::lift()
## x dplyr::where()      masks party::where()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(tidymodels)
```

```
## -- Attaching packages --------------------------------------- tidymodels 1.2.0 --
## v broom        1.0.5     v rsample      1.2.1
## v dials        1.3.0     v tune         1.2.1
## v infer        1.0.7     v workflows    1.1.4
## v modeldata    1.4.0     v workflowsets 1.1.0
```

```
## v parsnip      1.2.1     v yardstick    1.3.2
## v recipes      1.1.0
## -- Conflicts ---------------------------------------- tidymodels_conflicts() --
## x scales::discard()        masks purrr::discard()
## x dplyr::filter()          masks stats::filter()
## x parsnip::fit()           masks infer::fit(), party::fit(), modeltools::fit()
## x recipes::fixed()         masks stringr::fixed()
## x dplyr::lag()             masks stats::lag()
## x purrr::lift()            masks caret::lift()
## x tune::parameters()       masks dials::parameters(), modeltools::parameters()
## x yardstick::precision()   masks caret::precision()
## x yardstick::recall()      masks caret::recall()
## x yardstick::sensitivity() masks caret::sensitivity()
## x yardstick::spec()        masks readr::spec()
## x yardstick::specificity() masks caret::specificity()
## x recipes::step()          masks stats::step()
## x recipes::update()        masks stats4::update(), stats::update()
## x dplyr::where()           masks party::where()
## * Search for functions across packages at https://www.tidymodels.org/find/
```

Load and tidy data

```
pretty_names <- read_csv("../feat_name_mapping.csv")
```

```
## Rows: 85 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): name_orig, name_pretty
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
data <- read_csv("../measurements/measurements.csv")
```

```
## Rows: 754 Columns: 108
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (20): fpath, KUK_ID, FileName, FileFormat, FolderPath, subcorpus, Source...
## dbl (85): RuleAbstractNouns, RuleAmbiguousRegards, RuleAnaphoricReferences, ...
## lgl  (3): ClarityPursuit, SyllogismBased, Bindingness
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
data_no_nas <- data %>%
  select(!c(
    fpath,
    # KUK_ID,
    # FileName,
    FolderPath,
    # subcorpus,
    DocumentTitle,
    ClarityPursuit,
    Readability,
```

```
  SyllogismBased,
  SourceDB
)) %>%
# replace -1s in variation coefficients with NAs
mutate(across(c(
  `RuleDoubleAdpos.max_allowable_distance.v`,
  `RuleTooManyNegations.max_negation_frac.v`,
  `RuleTooManyNegations.max_allowable_negations.v`,
  `RuleTooManyNominalConstructions.max_noun_frac.v`,
  `RuleTooManyNominalConstructions.max_allowable_nouns.v`,
  `RuleCaseRepetition.max_repetition_count.v`,
  `RuleCaseRepetition.max_repetition_frac.v`,
  `RulePredSubjDistance.max_distance.v`,
  `RulePredObjDistance.max_distance.v`,
  `RuleInfVerbDistance.max_distance.v`,
  `RuleMultiPartVerbs.max_distance.v`,
  `RuleLongSentences.max_length.v`,
  `RulePredAtClauseBeginning.max_order.v`,
  `mattr.v`,
  `maentropy.v`
), ~ na_if(.x, -1))) %>%
# replace NAs with 0s
replace_na(list(
  RuleGPcoordovs = 0,
  RuleGPdeverbaddr = 0,
  RuleGPpatinstr = 0,
  RuleGPdeverbsubj = 0,
  RuleGPadjective = 0,
  RuleGPpatbenperson = 0,
  RuleGPwordorder = 0,
  RuleDoubleAdpos = 0,
  RuleDoubleAdpos.max_allowable_distance = 0,
  RuleDoubleAdpos.max_allowable_distance.v = 0,
  RuleAmbiguousRegards = 0,
  RuleReflexivePassWithAnimSubj = 0,
  RuleTooManyNegations = 0,
  RuleTooManyNegations.max_negation_frac = 0,
  RuleTooManyNegations.max_negation_frac.v = 0,
  RuleTooManyNegations.max_allowable_negations = 0,
  RuleTooManyNegations.max_allowable_negations.v = 0,
  RuleTooManyNominalConstructions.max_noun_frac.v = 0,
  RuleTooManyNominalConstructions.max_allowable_nouns.v = 0,
  RuleFunctionWordRepetition = 0,
  RuleCaseRepetition.max_repetition_count.v = 0,
  RuleCaseRepetition.max_repetition_frac.v = 0,
  RuleWeakMeaningWords = 0,
  RuleAbstractNouns = 0,
  RuleRelativisticExpressions = 0,
  RuleConfirmationExpressions = 0,
  RuleRedundantExpressions = 0,
  RuleTooLongExpressions = 0,
  RuleAnaphoricReferences = 0,
  RuleLiteraryStyle = 0,
```

```
    RulePassive = 0,
    RulePredSubjDistance = 0,
    RulePredSubjDistance.max_distance = 0,
    RulePredSubjDistance.max_distance.v = 0,
    RulePredObjDistance = 0,
    RulePredObjDistance.max_distance = 0,
    RulePredObjDistance.max_distance.v = 0,
    RuleInfVerbDistance = 0,
    RuleInfVerbDistance.max_distance = 0,
    RuleInfVerbDistance.max_distance.v = 0,
    RuleMultiPartVerbs = 0,
    RuleMultiPartVerbs.max_distance = 0,
    RuleMultiPartVerbs.max_distance.v = 0,
    RuleLongSentences.max_length.v = 0,
    RulePredAtClauseBeginning.max_order.v = 0,
    RuleVerbalNouns = 0,
    RuleDoubleComparison = 0,
    RuleWrongValencyCase = 0,
    RuleWrongVerbonominalCase = 0,
    RuleIncompleteConjunction = 0
  ))

data_clean <- data_no_nas %>%
  # norm data expected to correlate with text length
  mutate(across(c(
    RuleGPcoordovs,
    RuleGPdeverbaddr,
    RuleGPpatinstr,
    RuleGPdeverbsubj,
    RuleGPadjective,
    RuleGPpatbenperson,
    RuleGPwordorder,
    RuleDoubleAdpos,
    RuleAmbiguousRegards,
    RuleFunctionWordRepetition,
    RuleWeakMeaningWords,
    RuleAbstractNouns,
    RuleRelativisticExpressions,
    RuleConfirmationExpressions,
    RuleRedundantExpressions,
    RuleTooLongExpressions,
    RuleAnaphoricReferences,
    RuleLiteraryStyle,
    RulePassive,
    RuleVerbalNouns,
    RuleDoubleComparison,
    RuleWrongValencyCase,
    RuleWrongVerbonominalCase,
    RuleIncompleteConjunction,
    num_hapax,
    RuleReflexivePassWithAnimSubj,
    RuleTooManyNominalConstructions,
    RulePredSubjDistance,
```

```
    RuleMultiPartVerbs,
    RulePredAtClauseBeginning
  ), ~ .x / word_count)) %>%
  mutate(across(c(
    RuleTooFewVerbs,
    RuleTooManyNegations,
    RuleCaseRepetition,
    RuleLongSentences,
    RulePredObjDistance,
    RuleInfVerbDistance
  ), ~ .x / sent_count)) %>%
  # remove variables identified as "u counts"
  select(!c(
    RuleTooFewVerbs,
    RuleTooManyNegations,
    RuleTooManyNominalConstructions,
    RuleCaseRepetition,
    RuleLongSentences,
    RulePredAtClauseBeginning,
    sent_count,
    word_count,
    syllab_count,
    char_count
  )) %>%
  # remove variables identified as unreliable
  select(!c(
    RuleAmbiguousRegards,
    RuleFunctionWordRepetition,
    RuleDoubleComparison,
    RuleWrongValencyCase,
    RuleWrongVerbonominalCase
  )) %>%
  # remove artificially limited variables
  select(!c(
    RuleCaseRepetition.max_repetition_frac,
    RuleCaseRepetition.max_repetition_frac.v
  )) %>%
  # remove further variables belonging to the 'acceptability' category
  select(!c(RuleIncompleteConjunction)) %>%
  unite("strata", c(subcorpus, class), sep = "_", remove = FALSE) %>%
  mutate(across(c(class), ~ as.factor(.x)))

# no NAs should be present now
data_clean[!complete.cases(data_clean), ]
```

```
## # A tibble: 754 x 84
##    KUK_ID          FileName FileFormat strata subcorpus SourceID DocumentVersion
##    <chr>           <chr>    <chr>      <chr>  <chr>     <chr>    <chr>
##  1 673b7a37c6537d~ 002_Kom~ TXT        KUKY_~ KUKY      <NA>     Original
##  2 673b7a37c6537d~ 006_Chc~ TXT        KUKY_~ KUKY      <NA>     Redesign
##  3 673b7a37c6537d~ 004_Nev~ TXT        KUKY_~ KUKY      <NA>     Original
##  4 673b7a37c6537d~ 008_Pol~ TXT        KUKY_~ KUKY      <NA>     Original
##  5 673b7a37c6537d~ 005_Och~ TXT        KUKY_~ KUKY      <NA>     Original
##  6 673b7a37c6537d~ 016_Obc~ TXT        KUKY_~ KUKY      <NA>     Original
```

```
##  7 673b7a37c6537d~ 019_Dět~ TXT        KUKY_~ KUKY      <NA>     Redesign
##  8 673b7a37c6537d~ 007_DŮC~ TXT        KUKY_~ KUKY      <NA>     Redesign
##  9 673b7a37c6537d~ 024_Opa~ TXT        KUKY_~ KUKY      <NA>     Original
## 10 673b7a37c6537d~ 047_Dav~ TXT        KUKY_~ KUKY      <NA>     Original
## # i 744 more rows
## # i 77 more variables: ParentDocumentID <chr>, LegalActType <chr>,
## #   Objectivity <chr>, Bindingness <lgl>, AuthorType <chr>,
## #   RecipientType <chr>, RecipientIndividuation <chr>, Anonymized <chr>,
## #   `Recipient Type` <chr>, class <fct>, RuleAbstractNouns <dbl>,
## #   RuleAnaphoricReferences <dbl>,
## #   RuleCaseRepetition.max_repetition_count <dbl>, ...
```

```
.firstnonmetacolumn <- 18

prettify_feat_name <- function(x) {
  name <- pull(pretty_names %>%
    filter(name_orig == x), name_pretty)
  if (length(name) == 1) {
    return(name)
  } else {
    return(x)
  }
}

prettify_feat_name_vector <- function(x) {
  map(
    x,
    prettify_feat_name
  ) %>% unlist()
}

colnames(data_clean) <- prettify_feat_name_vector(colnames(data_clean))
```

Filter for features identified as important

This may not be necessary, as the identification was crucial to the EFA above all, so that features irrelevant for readability would not appear in the model. It may be useful to compare the importances of a model trained on all features and on a selected-feature model.

```
selected_features_tibble <- read_csv("../efa/selected_features.csv") %>%
  mutate(across(feat_name, prettify_feat_name_vector))
```

```
## Rows: 67 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): feat_name
## dbl (1): p_value
## lgl (1): selected
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
formula_all <- reformulate(
  selected_features_tibble %>% pull(feat_name), "class"
)
```

```
formula_selected <- reformulate(
  selected_features_tibble %>% filter(selected) %>% pull(feat_name), "class"
)
```

Split and folds

```
.split_prop <- 4 / 5
.no_folds <- 10

split <- data_clean %>% initial_split(prop = .split_prop, strata = strata)

training_set <- training(split)
testing_set <- testing(split)

folds <- vfold_cv(training_set, v = .no_folds, strata = strata)

nrow(training_set)
```

```
## [1] 601
```

```
training_set %>%
  select(subcorpus, class) %>%
  table()
```

```
##             class
## subcorpus    bad good
##   CzCDC      170    0
##   FrBo        62  183
##   KUKY        65   87
##   LiFRLaw      2    0
##   OmbuFlyers  32    0
```

```
nrow(testing_set)
```

```
## [1] 153
```

```
testing_set %>%
  select(subcorpus, class) %>%
  table()
```

```
##             class
## subcorpus    bad good
##   CzCDC       44    0
##   FrBo        16   46
##   KUKY        17   23
##   LiFRLaw      1    0
##   OmbuFlyers   6    0
```

Experimental model

To familiarize myself with the library and CRFs.

```
training_split <- training_set %>%
  initial_split(prop = .split_prop, strata = strata)
train_subset <- training(training_split)
```

```r
devtest_subset <- testing(training_split)

model_rf_exp <- cforest(
  formula_selected,
  data = train_subset, controls = cforest_control(ntree = 1000)
)

predictions_exp <- predict(model_rf_exp, newdata = devtest_subset)
confusionMatrix(
  predictions_exp, devtest_subset$class,
  positive = "good", mode = "everything"
)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction bad good
##       bad   49   18
##       good  19   37
##
##                Accuracy : 0.6992
##                  95% CI : (0.61, 0.7786)
##     No Information Rate : 0.5528
##     P-Value [Acc > NIR] : 0.00063
##
##                   Kappa : 0.3926
##
##  Mcnemar's Test P-Value : 1.00000
##
##             Sensitivity : 0.6727
##             Specificity : 0.7206
##          Pos Pred Value : 0.6607
##          Neg Pred Value : 0.7313
##               Precision : 0.6607
##                  Recall : 0.6727
##                      F1 : 0.6667
##              Prevalence : 0.4472
##          Detection Rate : 0.3008
##    Detection Prevalence : 0.4553
##       Balanced Accuracy : 0.6967
##
##        'Positive' Class : good
##
```

```r
# computationally expensive
# importances_exp <- varimp(model_rf_exp)

# even more computationally expensive
# cimportances_exp <- varimp(model_rf_exp, conditional = TRUE)
```

MFV model

```r
(nrow(data_clean %>% filter(class == "bad")) / nrow(data_clean)) %>%
  round(3)
```

```
## [1] 0.55
```

```r
(nrow(training_set %>% filter(class == "bad")) / nrow(training_set)) %>%
  round(3)
```

```
## [1] 0.551
```

```r
(nrow(testing_set %>% filter(class == "bad")) / nrow(testing_set)) %>%
  round(3)
```

```
## [1] 0.549
```

Helpers

```r
ntree_tune_levels <- 500 + 0:8 * 250

tune_crf <- function(formula, folds, ntree_tune_levels) {
  accuracy_column <- numeric()
  ntree_column <- numeric()
  fold_column <- numeric()

  for (ntree_ in ntree_tune_levels) {
    message(paste0(c("ntree_ ", ntree_), collapse = " "))
    ctrl <- cforest_control(ntree = ntree_)

    for (i in seq_len(nrow(folds))) {
      alldata <- pull(folds[i, 1])[[1]]$data
      trindices <- pull(folds[i, 1])[[1]]$in_id
      trdata <- alldata[trindices, ]
      tsdata <- alldata[-trindices, ]

      model <- cforest(formula, data = trdata, controls = ctrl)
      pred <- predict(model, newdata = tsdata)

      cm <- confusionMatrix(pred, tsdata$class, positive = "good")

      ntree_column <- c(ntree_column, ntree_)
      fold_column <- c(fold_column, i)
      accuracy_column <- c(accuracy_column, cm$overall["Accuracy"])
    }
  }

  data.frame(
    ntree = ntree_column,
    fold = fold_column,
    accuracy = accuracy_column
  )
}

get_mismatch_details <- function(data_with_predictions) {
  print(
```

```
    data_with_predictions %>%
      ggplot(aes(x = .prob, y = class, color = subcorpus)) +
      geom_jitter(height = 0.2, width = 0)
  )

  cat("Confusion matrices by subcorpora:\n")
  data_with_predictions %>%
    select(.pred, class, subcorpus) %>%
    table() %>%
    print()

  cat("\n")

  deviations <- data_with_predictions %>%
    filter(.pred != class) %>%
    mutate(abs_dev = abs(.prob - 0.5)) %>%
    arrange(-abs_dev)

  cat("Greatest deviations:\n")
  deviations %>%
    select(abs_dev, .prob, class, subcorpus, FileName) %>%
    mutate(across(c(.prob, abs_dev), ~ round(.x, 3))) %>%
    print(n = round(nrow(data_with_predictions) / 5))

  cat("Names of highest-deviating documents:\n")
  highest_deviation_names <- deviations %>%
    filter(abs_dev >= 0.25) %>%
    arrange(-abs_dev) %>%
    pull(FileName)

  print(highest_deviation_names)

  return(list(
    deviations = deviations, highest_deviations = highest_deviation_names
  ))
}

plot_outlier <- function(doc_name, variable_importances, dataset) {
  important_variables <- sort(variable_importances) %>% tail(n = 9)
  varnames <- names(important_variables)

  dmut <- dataset %>%
    select(KUK_ID, FileName, class, all_of(varnames)) %>%
    mutate(across(all_of(varnames), ~ scale(.x))) %>%
    pivot_longer(
      all_of(varnames),
      names_to = "feature", values_to = "value"
    ) %>%
    mutate(across(value, ~ .x[, 1]))

  cat(nrow(dmut %>% filter(value > 5)), "observation(s) removed from the plot\n")
  dmutf <- dmut %>% filter(value <= 5)
```

```
  dmutf %>%
    ggplot(aes(x = class, y = value)) +
    facet_wrap(~feature) +
    geom_boxplot() +
    geom_point(
      data = dmut %>% filter(FileName == doc_name), color = "red", size = 5
    ) +
    labs(y = "measurements (scaled)")
}
```

Selected-features model

Tune

```
tune_df_sel <- tune_crf(formula_selected, folds, ntree_tune_levels)
```

```
## ntree_  500
## ntree_  750
## ntree_  1000
## ntree_  1250
## ntree_  1500
## ntree_  1750
## ntree_  2000
## ntree_  2250
## ntree_  2500
```

```
tune_df_sel %>%
  group_by(ntree) %>%
  summarize(mean_acc = mean(accuracy), sd_acc = sd(accuracy))
```

```
## # A tibble: 9 x 3
##   ntree mean_acc sd_acc
##   <dbl>    <dbl>  <dbl>
## 1   500    0.769 0.0319
## 2   750    0.760 0.0357
## 3  1000    0.761 0.0363
## 4  1250    0.766 0.0268
## 5  1500    0.761 0.0338
## 6  1750    0.756 0.0350
## 7  2000    0.757 0.0365
## 8  2250    0.762 0.0320
## 9  2500    0.761 0.0319
```

```
tune_df_sel %>%
  group_by(fold) %>%
  summarize(mean_acc = mean(accuracy), sd_acc = sd(accuracy))
```

```
## # A tibble: 10 x 3
##     fold mean_acc  sd_acc
##    <dbl>    <dbl>   <dbl>
```

```
##  1     1    0.802 0.0140
##  2     2    0.738 0.00711
##  3     3    0.754 0
##  4     4    0.739 0.0118
##  5     5    0.780 0.0111
##  6     6    0.713 0.0232
##  7     7    0.741 0.00878
##  8     8    0.746 0
##  9     9    0.807 0.00760
## 10    10    0.793 0
```

```
best_ntree_sel <- tune_df_sel %>%
  group_by(ntree) %>%
  summarize(mean_acc = mean(accuracy)) %>%
  arrange(-mean_acc) %>%
  head(n = 1) %>%
  pull(ntree)
```

Fit

```
model_crf_sel <- cforest(
  formula_selected, training_set,
  controls = cforest_control(ntree = best_ntree_sel)
)

predictions_sel_prob <- predict(
  model_crf_sel,
  newdata = testing_set, type = "prob"
) %>%
  map(function(x) x[1, 2]) %>%
  unlist() %>%
  as.vector()
predictions_sel <- if_else(predictions_sel_prob > 0.5, "good", "bad") %>%
  as.factor()

confusionMatrix(
  predictions_sel, testing_set$class,
  positive = "good", mode = "everything"
)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction bad good
##       bad   63   14
##       good  21   55
##
##                Accuracy : 0.7712
##                  95% CI : (0.6965, 0.8352)
##     No Information Rate : 0.549
##     P-Value [Acc > NIR] : 9.382e-09
##
##                   Kappa : 0.5422
##
```

```
##  Mcnemar's Test P-Value : 0.3105
##
##             Sensitivity : 0.7971
##             Specificity : 0.7500
##          Pos Pred Value : 0.7237
##          Neg Pred Value : 0.8182
##               Precision : 0.7237
##                  Recall : 0.7971
##                      F1 : 0.7586
##              Prevalence : 0.4510
##          Detection Rate : 0.3595
##    Detection Prevalence : 0.4967
##       Balanced Accuracy : 0.7736
##
##        'Positive' Class : good
##
```

```
cimportances_sel <- varimp(model_crf_sel, conditional = TRUE, nperm = 20)
cimportances_sel %>%
  sort() %>%
  as.data.frame() %>%
  print(digits = 3)
```

```
##                              .
## NEGcount.v           -5.64e-05
## xcomp                -5.17e-05
## NOUNfrac.v           -5.17e-05
## NEGfrac.v            -4.33e-05
## predobjdist.v        -3.98e-05
## predsubjdist.m       -3.66e-05
## caserepcount.v       -2.67e-05
## obj                  -1.86e-05
## predsubjdist.v       -1.33e-05
## GPwordorder          -1.32e-05
## predobjdist.m        -1.14e-05
## predorder.v          -9.07e-06
## hapaxes              -1.01e-06
## redundexprs           0.00e+00
## hpoint                3.09e-06
## compoundVERBs         1.00e-05
## fre                   1.34e-05
## doubleADPdist.m       1.48e-05
## NEGfrac.m             1.79e-05
## maentropy             1.94e-05
## weakmeaning           1.97e-05
## analyticVERBsdist.m   2.41e-05
## VERBfrac.v            3.34e-05
## GPdeverbsubj          3.50e-05
## relativisticexprs     3.75e-05
## cli                   4.80e-05
## smog                  5.02e-05
## fkgl                  6.01e-05
## NEGcount.m            6.19e-05
## abstractNOUNs         6.84e-05
## doubleADPdist.v       7.00e-05
```

```
## subj                 7.19e-05
## gf                   7.45e-05
## entropy              7.67e-05
## analyticVERBsdist.v  8.82e-05
## NOUNcount.v          9.04e-05
## GPdeverbaddr         9.06e-05
## mamr                 9.23e-05
## predorder.m          1.05e-04
## atl                  1.24e-04
## ari                  1.41e-04
## verbalNOUNs          1.47e-04
## mattr                1.51e-04
## NOUNcount.m          1.90e-04
## entropy.v            1.93e-04
## VERBfrac.m           2.19e-04
## passives             2.65e-04
## sentlen.m            3.49e-04
## verbdist             4.27e-04
## literary             4.81e-04
## activity             7.85e-04
```

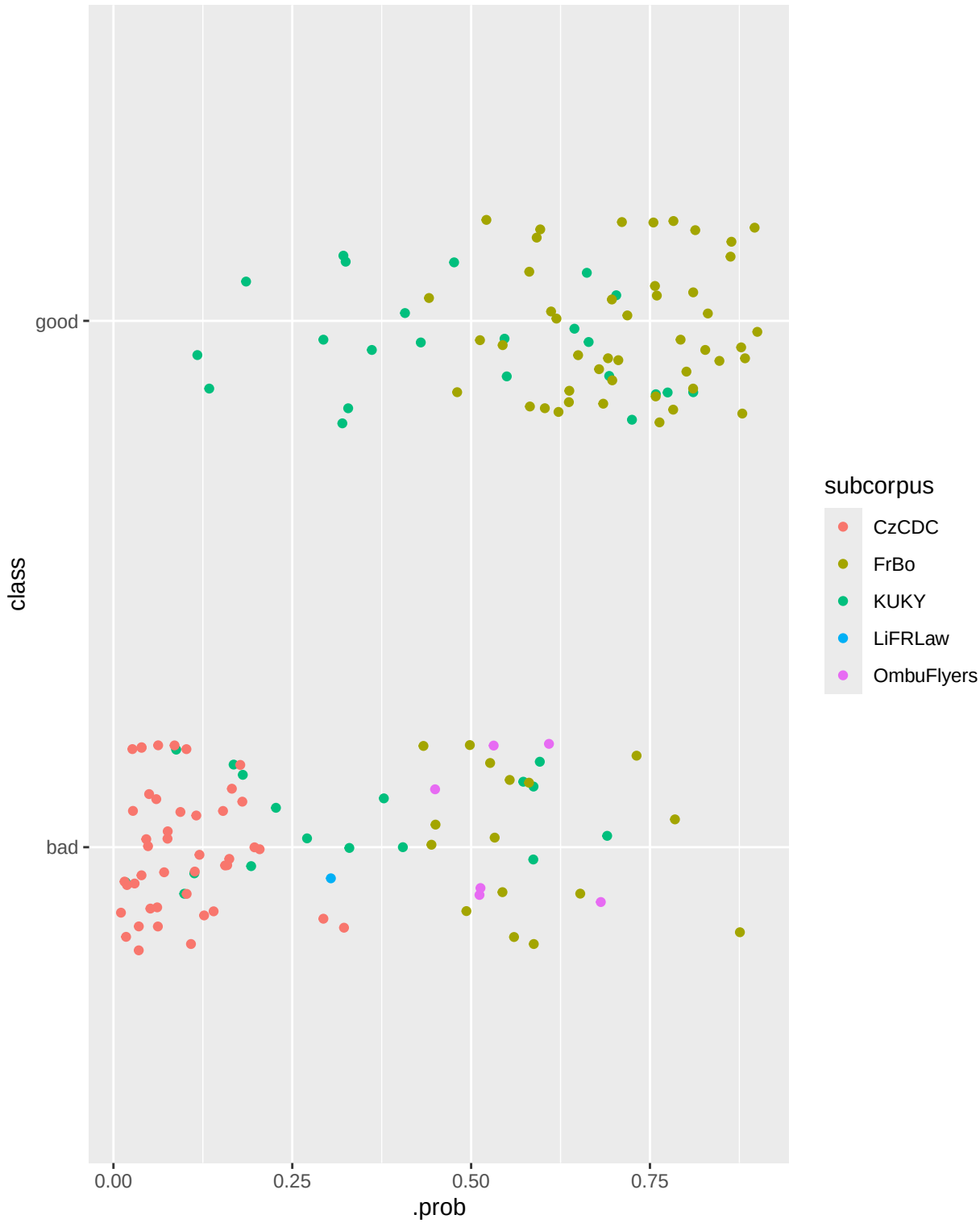
```
cimportances_sel %>%
  abs() %>%
  sort(decreasing = TRUE) %>%
  as.data.frame() %>%
  print(digits = 3)
```

```
##                          .
## activity            7.85e-04
## literary            4.81e-04
## verbdist            4.27e-04
## sentlen.m           3.49e-04
## passives            2.65e-04
## VERBfrac.m          2.19e-04
## entropy.v           1.93e-04
## NOUNcount.m         1.90e-04
## mattr               1.51e-04
## verbalNOUNs         1.47e-04
## ari                 1.41e-04
## atl                 1.24e-04
## predorder.m         1.05e-04
## mamr                9.23e-05
## GPdeverbaddr        9.06e-05
## NOUNcount.v         9.04e-05
## analyticVERBsdist.v 8.82e-05
## entropy             7.67e-05
## gf                  7.45e-05
## subj                7.19e-05
## doubleADPdist.v     7.00e-05
## abstractNOUNs       6.84e-05
## NEGcount.m          6.19e-05
## fkgl                6.01e-05
## NEGcount.v          5.64e-05
## xcomp               5.17e-05
## NOUNfrac.v          5.17e-05
```

```
## smog                5.02e-05
## cli                 4.80e-05
## NEGfrac.v           4.33e-05
## predobjdist.v       3.98e-05
## relativisticexprs   3.75e-05
## predsubjdist.m      3.66e-05
## GPdeverbsubj        3.50e-05
## VERBfrac.v          3.34e-05
## caserepcount.v      2.67e-05
## analyticVERBsdist.m 2.41e-05
## weakmeaning         1.97e-05
## maentropy           1.94e-05
## obj                 1.86e-05
## NEGfrac.m           1.79e-05
## doubleADPdist.m     1.48e-05
## fre                 1.34e-05
## predsubjdist.v      1.33e-05
## GPwordorder         1.32e-05
## predobjdist.m       1.14e-05
## compoundVERBs       1.00e-05
## predorder.v         9.07e-06
## hpoint              3.09e-06
## hapaxes             1.01e-06
## redundexprs         0.00e+00
```

```
testing_set_sel <- testing_set %>%
  mutate(.prob = predictions_sel_prob, .pred = predictions_sel)
```

```
mismatches_sel <- get_mismatch_details(testing_set_sel)
```



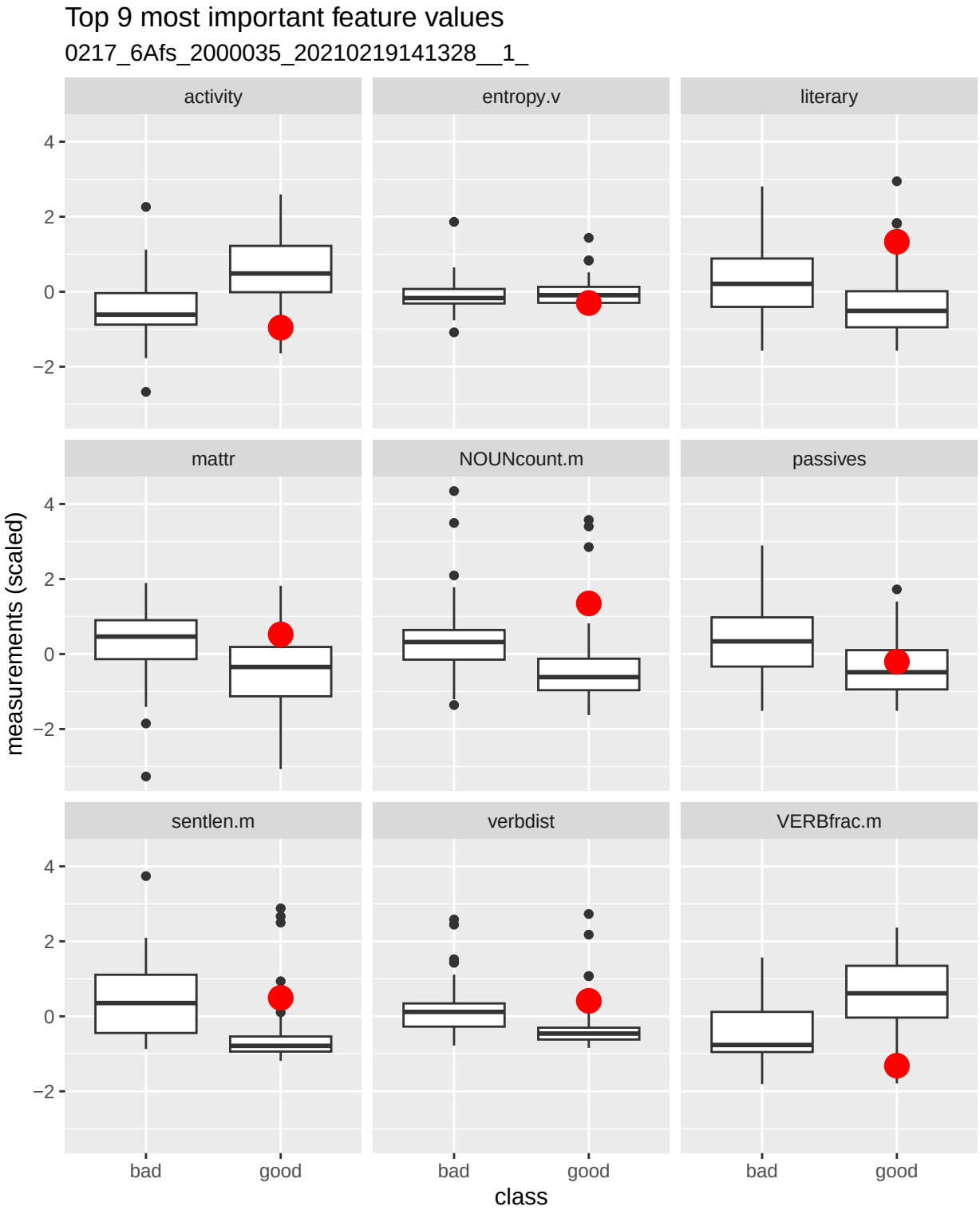

```
## Confusion matrices by subcorpora:
## , , subcorpus = CzCDC
##
##       class
## .pred  bad good
##   bad   44    0
```

```
##   good   0    0
##
## , , subcorpus = FrBo
##
##       class
## .pred  bad good
##   bad    5    2
##   good  11   44
##
## , , subcorpus = KUKY
##
##       class
## .pred  bad good
##   bad   12   12
##   good   5   11
##
## , , subcorpus = LiFRLaw
##
##       class
## .pred  bad good
##   bad    1    0
##   good   0    0
##
## , , subcorpus = OmbuFlyers
##
##       class
## .pred  bad good
##   bad    1    0
##   good   5    0
##
##
## Greatest deviations:
## # A tibble: 35 x 5
##    abs_dev .prob class subcorpus  FileName
##      <dbl> <dbl> <fct> <chr>      <chr>
##  1   0.383 0.117 good  KUKY       0217_6Afs_2000035_20210219141328__1_
##  2   0.376 0.876 bad   FrBo       orig_Co můžete dělat, pokud obec postupuje př~
##  3   0.366 0.134 good  KUKY       11_vizum_pred
##  4   0.315 0.185 good  KUKY       Odvolani
##  5   0.285 0.785 bad   FrBo       orig_Jak probíhá správní řízení
##  6   0.231 0.731 bad   FrBo       orig_Jak se bránit neposkytnutí projektové do~
##  7   0.206 0.294 good  KUKY       Mestsky_urad_usneseni_-_slouceni_pred
##  8   0.19  0.69  bad   KUKY       043_Plisen-a-zavady-v-byte
##  9   0.181 0.681 bad   OmbuFlyers Soudni-poplatky
## 10   0.18  0.32  good  KUKY       1A_dokument_puvodni_ustanoven_zastupce_vyzva_~
## 11   0.179 0.321 good  KUKY       2A_dokument_puvodni_vyzva_k_zaplaceni_SOP_a_k~
## 12   0.175 0.325 good  KUKY       Mestsky_urad_Souhlas_s_prestupkovym_rizenim
## 13   0.172 0.328 good  KUKY       Odvolani_proti_rozhodnuti_o_nepovoleni_kaceni
## 14   0.153 0.653 bad   FrBo       orig_Kompletní průvodce občana obtěžovaného h~
## 15   0.139 0.361 good  KUKY       Zaloba_na_zruseni_spoluvlastnictvi
## 16   0.109 0.609 bad   OmbuFlyers Detsky-domov
## 17   0.096 0.596 bad   KUKY       PR_Masinova
## 18   0.092 0.408 good  KUKY       Mestsky_urad_kontrola_po
## 19   0.088 0.588 bad   FrBo       68
```

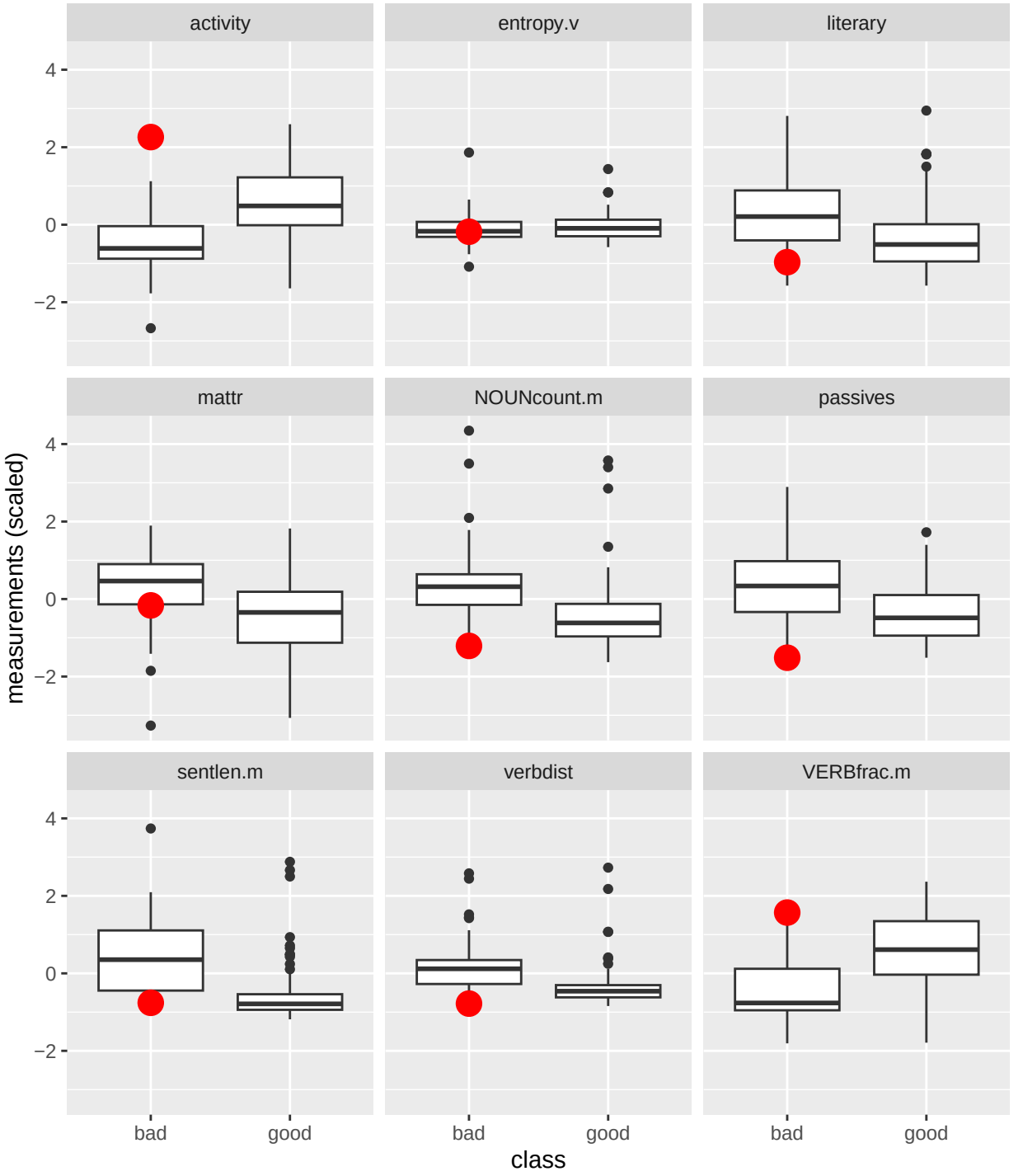
```
## 20   0.087 0.587 bad   KUKY       sluzebni_hodnoceni_puvodni
## 21   0.087 0.587 bad   KUKY       024_Opatrovnictvi
## 22   0.081 0.581 bad   FrBo       red_Smlouvy obcí s investory
## 23   0.073 0.573 bad   KUKY       41_A_32-2022_rozsudek_Martina_Kopy_Anna_Rybar~
## 24   0.07  0.43  good  KUKY       Mestsky_urad_Usneseni_narizeni_podrobit_se_pr~
## 25   0.06  0.56  bad   FrBo       189
## 26   0.059 0.441 good  FrBo       red_provokace_korupcniho_jednani
## 27   0.054 0.554 bad   FrBo       170
## 28   0.044 0.544 bad   FrBo       153
## 29   0.033 0.533 bad   FrBo       orig_Jaké trestné činy mohou souviset s korup~
## 30   0.032 0.532 bad   OmbuFlyers Stavebni-cinnost
## 31   0.027 0.527 bad   FrBo       orig_Jak probíhá trestní řízení
## # i 4 more rows
## Names of highest-deviating documents:
## [1] "0217_6Afs_2000035_20210219141328__1_"
## [2] "orig_Co můžete dělat, pokud obec postupuje při prodeji nebo pronájmu pozemků nezákonně_final"
## [3] "11_vizum_pred"
## [4] "Odvolani"
## [5] "orig_Jak probíhá správní řízení"
```

```
for (dev in mismatches_sel$highest_deviations) {
  print(plot_outlier(dev, cimportances_sel, testing_set_sel) +
    labs(title = "Top 9 most important feature values", subtitle = dev))
}
```

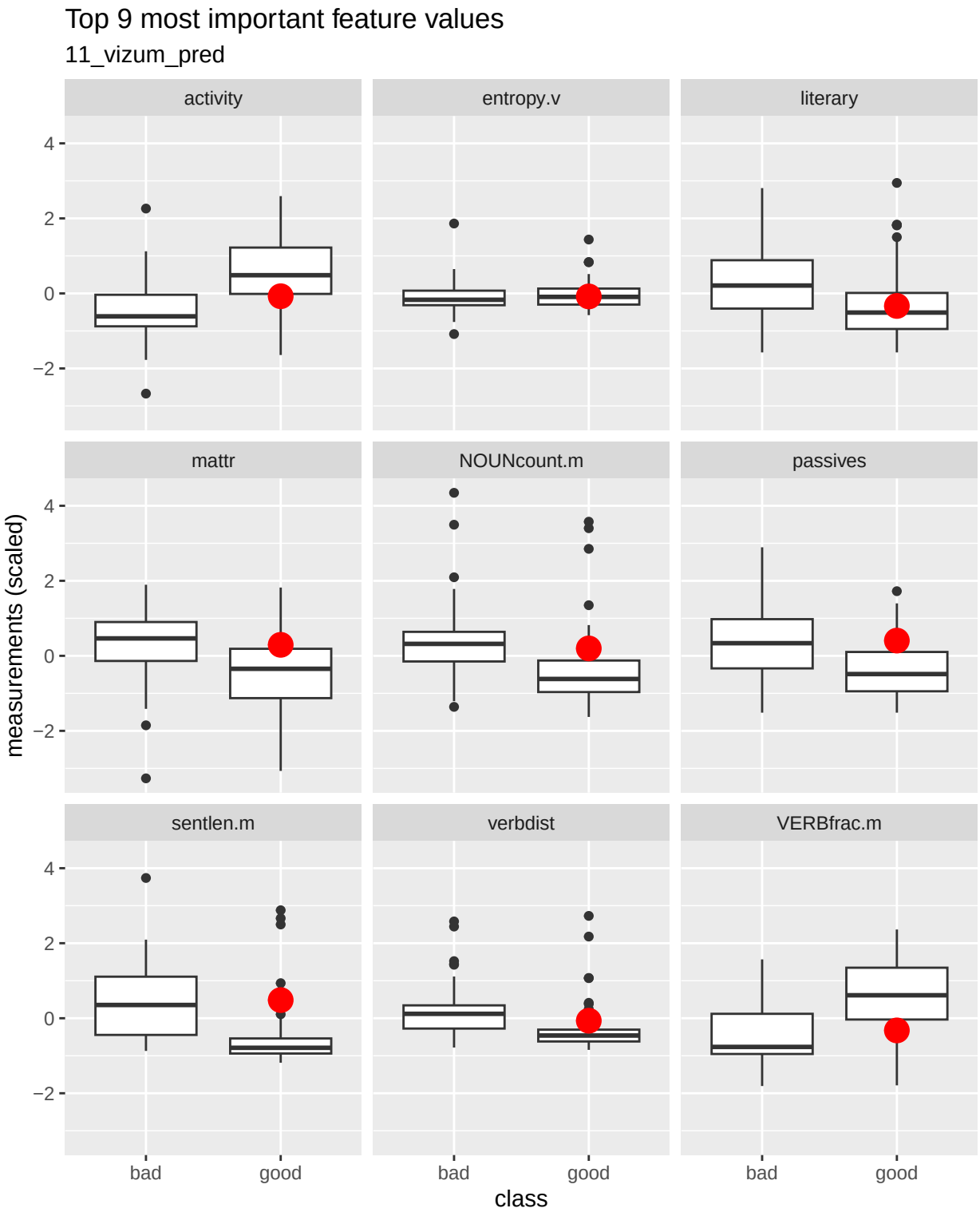
```
## 2 observation(s) removed from the plot
```



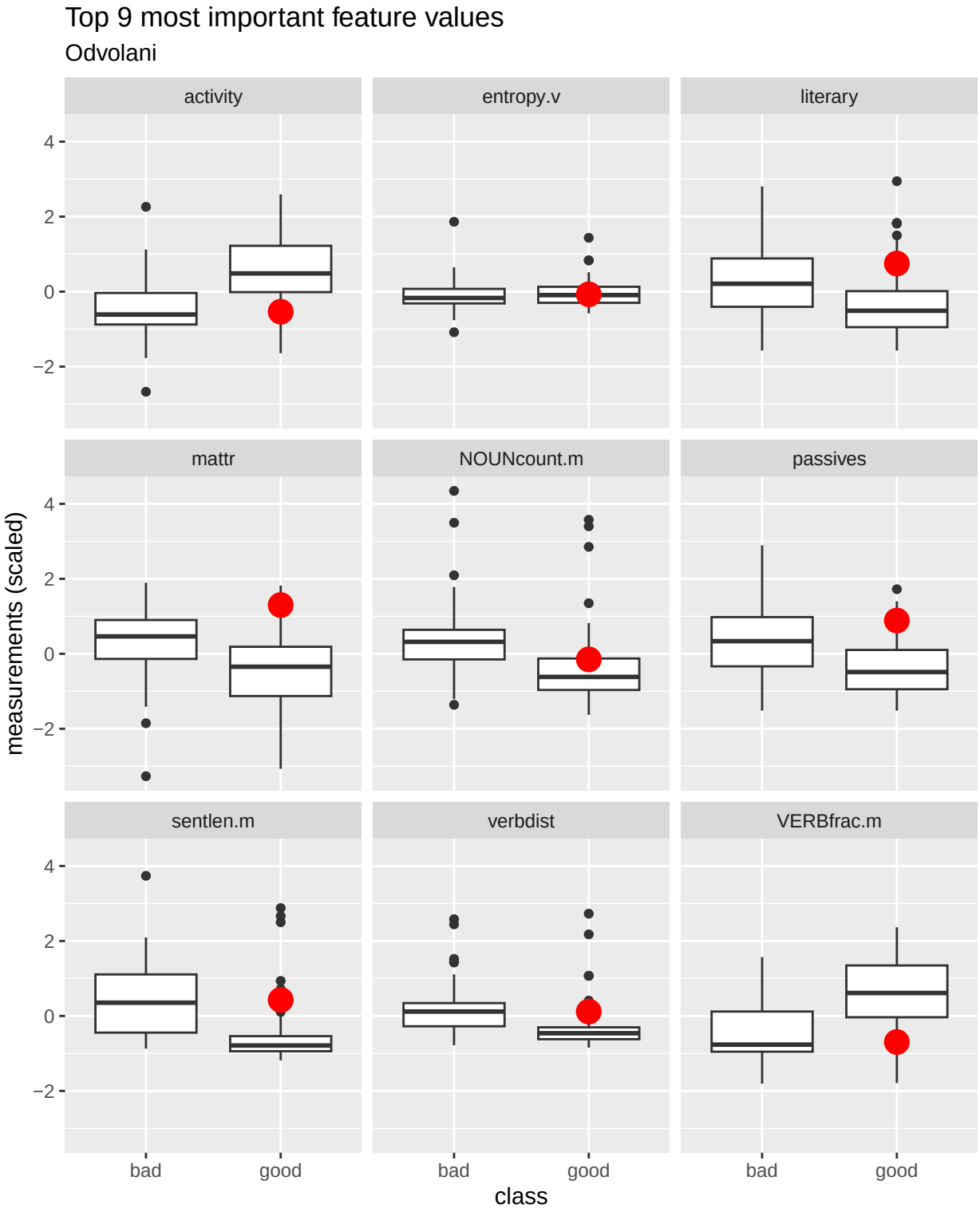

```
## 2 observation(s) removed from the plot
```



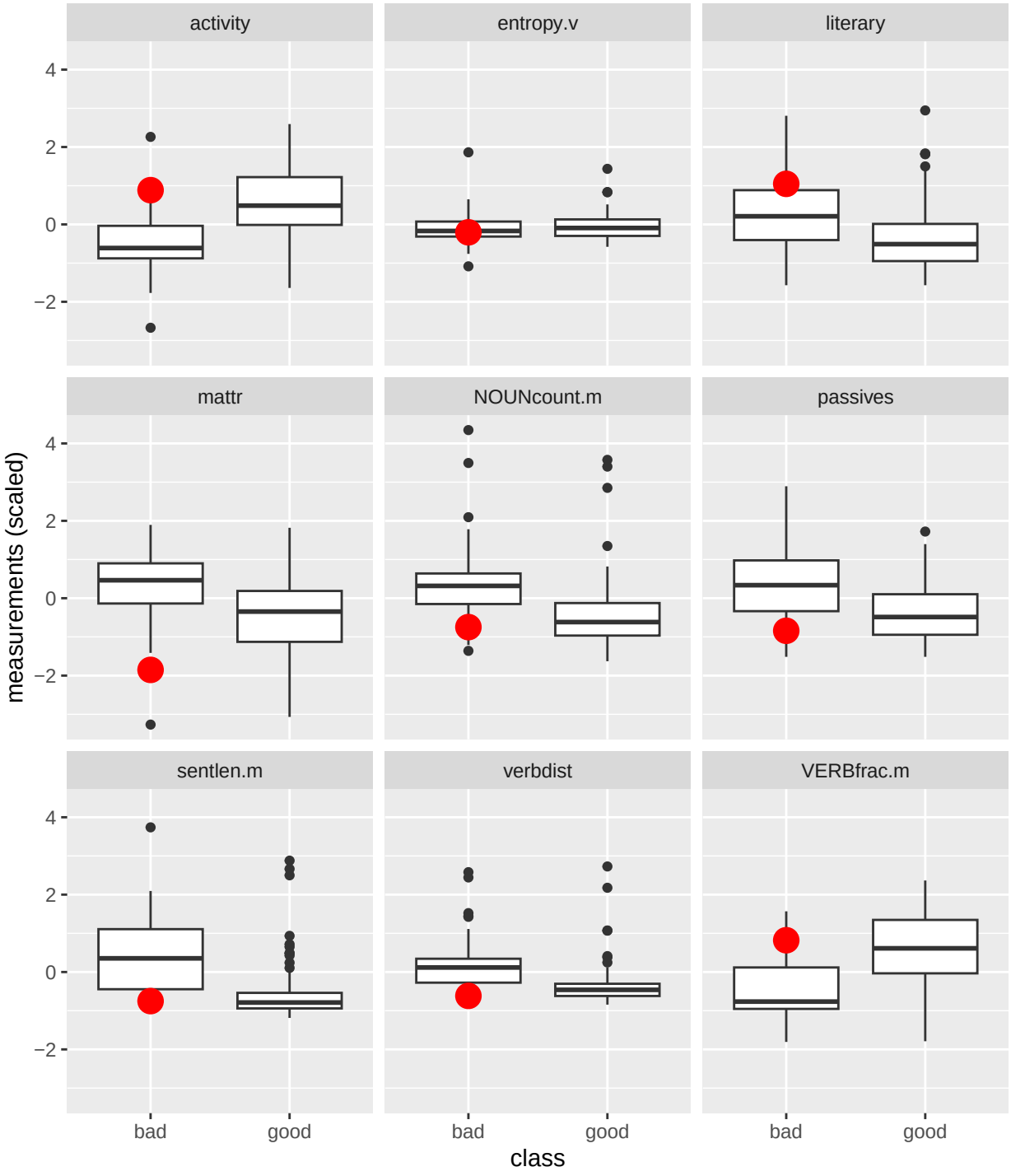

```
## 2 observation(s) removed from the plot
```




```
## 2 observation(s) removed from the plot
```




```
## 2 observation(s) removed from the plot
```


Top 9 most important feature values
orig_Jak probíhá správní .ízení
activity
entropy.v
literary
mattr
NOUNcount.m
passives
sentlen.m
verbdist
VERBfrac.m
measurements (scaled)
4
2
0
−2
bad
good
class

All-features model

Tune

```
tune_df_all <- tune_crf(formula_all, folds, ntree_tune_levels)
```

```
## ntree_  500
## ntree_  750
## ntree_  1000
## ntree_  1250
## ntree_  1500
## ntree_  1750
## ntree_  2000
## ntree_  2250
## ntree_  2500
```

```
tune_df_all %>%
  group_by(ntree) %>%
  summarize(mean_acc = mean(accuracy), sd_acc = sd(accuracy))
```

```
## # A tibble: 9 x 3
##   ntree mean_acc sd_acc
##   <dbl>    <dbl>  <dbl>
## 1   500    0.762 0.0371
## 2   750    0.759 0.0345
## 3  1000    0.759 0.0293
## 4  1250    0.764 0.0335
## 5  1500    0.757 0.0406
## 6  1750    0.759 0.0348
## 7  2000    0.762 0.0401
## 8  2250    0.762 0.0343
## 9  2500    0.752 0.0430
```

```
tune_df_all %>%
  group_by(fold) %>%
  summarize(mean_acc = mean(accuracy), sd_acc = sd(accuracy))
```

```
## # A tibble: 10 x 3
##     fold mean_acc  sd_acc
##    <dbl>    <dbl>   <dbl>
##  1     1    0.795 0.00529
##  2     2    0.731 0.0114
##  3     3    0.754 0
##  4     4    0.748 0.0100
##  5     5    0.804 0.0162
##  6     6    0.698 0.0155
##  7     7    0.741 0.0121
##  8     8    0.736 0.00893
##  9     9    0.795 0.00575
## 10    10    0.793 0
```

```r
best_ntree_all <- tune_df_all %>%
  group_by(ntree) %>%
  summarize(mean_acc = mean(accuracy)) %>%
  arrange(-mean_acc) %>%
  head(n = 1) %>%
  pull(ntree)
```

Fit

```r
model_crf_all <- cforest(
  formula_all, training_set,
  controls = cforest_control(ntree = best_ntree_all)
)

predictions_all_prob <- predict(
  model_crf_all,
  newdata = testing_set, type = "prob"
) %>%
  map(function(x) x[1, 2]) %>%
  unlist() %>%
  as.vector()
predictions_all <- if_else(predictions_all_prob > 0.5, "good", "bad") %>%
  as.factor()

confusionMatrix(
  predictions_all, testing_set$class,
  positive = "good", mode = "everything"
)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction bad good
##       bad   63   14
##       good  21   55
##
##                Accuracy : 0.7712
##                  95% CI : (0.6965, 0.8352)
##     No Information Rate : 0.549
##     P-Value [Acc > NIR] : 9.382e-09
##
##                   Kappa : 0.5422
##
##  Mcnemar's Test P-Value : 0.3105
##
##             Sensitivity : 0.7971
##             Specificity : 0.7500
##          Pos Pred Value : 0.7237
##          Neg Pred Value : 0.8182
##               Precision : 0.7237
##                  Recall : 0.7971
##                      F1 : 0.7586
##              Prevalence : 0.4510
```

```
##          Detection Rate : 0.3595
##    Detection Prevalence : 0.4967
##       Balanced Accuracy : 0.7736
##
##        'Positive' Class : good
##
```

```
cimportances_all <- varimp(model_crf_all, conditional = TRUE, nperm = 20)
cimportances_all %>%
  sort() %>%
  as.data.frame() %>%
  print(digits = 3)
```

```
##                              .
## doubleADPs           -3.45e-05
## relativisticexprs    -3.10e-05
## NEGfrac.v            -2.49e-05
## GPcoordovs           -1.70e-05
## caserepcount.m       -1.68e-05
## xcompdist.v          -1.63e-05
## xcomp                -1.51e-05
## predsubjdist.v       -1.34e-05
## extrcaseexprs        -1.28e-05
## NEGcount.v           -1.27e-05
## GPpatinstr           -1.01e-05
## predobjdist.m        -9.32e-06
## NOUNfrac.v           -7.05e-06
## doubleADPdist.m      -6.40e-06
## hpoint               -4.17e-06
## predobjdist.v        -3.87e-06
## fre                  -3.87e-06
## doubleADPdist.v      -3.68e-06
## analyticVERBsdist.v  -2.16e-06
## GPpatbenperson       -1.78e-06
## NEGcount.m           -1.20e-06
## predsubjdist.m       -1.02e-06
## GPadjective           0.00e+00
## redundexprs           2.83e-07
## GPwordorder           3.07e-06
## hapaxes               7.47e-06
## GPdeverbaddr          8.73e-06
## entropy               1.13e-05
## predorder.v           1.41e-05
## NEGfrac.m             1.53e-05
## abstractNOUNs         1.77e-05
## weakmeaning           1.92e-05
## VERBfrac.v            2.06e-05
## rfpass_animsubj       2.71e-05
## verbalNOUNs           2.79e-05
## maentropy             2.79e-05
## caserepcount.v        2.83e-05
## compoundVERBs         3.02e-05
## analyticVERBsdist.m   3.45e-05
## xcompdist.m           3.84e-05
## obj                   4.53e-05
```

```
## subj                 4.63e-05
## mattr                4.67e-05
## fkgl                 4.91e-05
## GPdeverbsubj         5.99e-05
## ttr.v                6.31e-05
## ttr                  6.44e-05
## smog                 6.96e-05
## NOUNcount.v          7.47e-05
## NOUNfrac.m           7.52e-05
## mamr                 8.46e-05
## cli                  8.52e-05
## longexprs            9.39e-05
## sentlen.v            9.78e-05
## NOUNcount.m          1.00e-04
## entropy.v            1.01e-04
## predorder.m          1.25e-04
## sentlen.m            1.43e-04
## gf                   1.46e-04
## ari                  1.52e-04
## anaphoricrefs        1.74e-04
## atl                  2.07e-04
## passives             2.34e-04
## verbdist             2.82e-04
## literary             3.20e-04
## activity             4.07e-04
## VERBfrac.m           5.78e-04
```

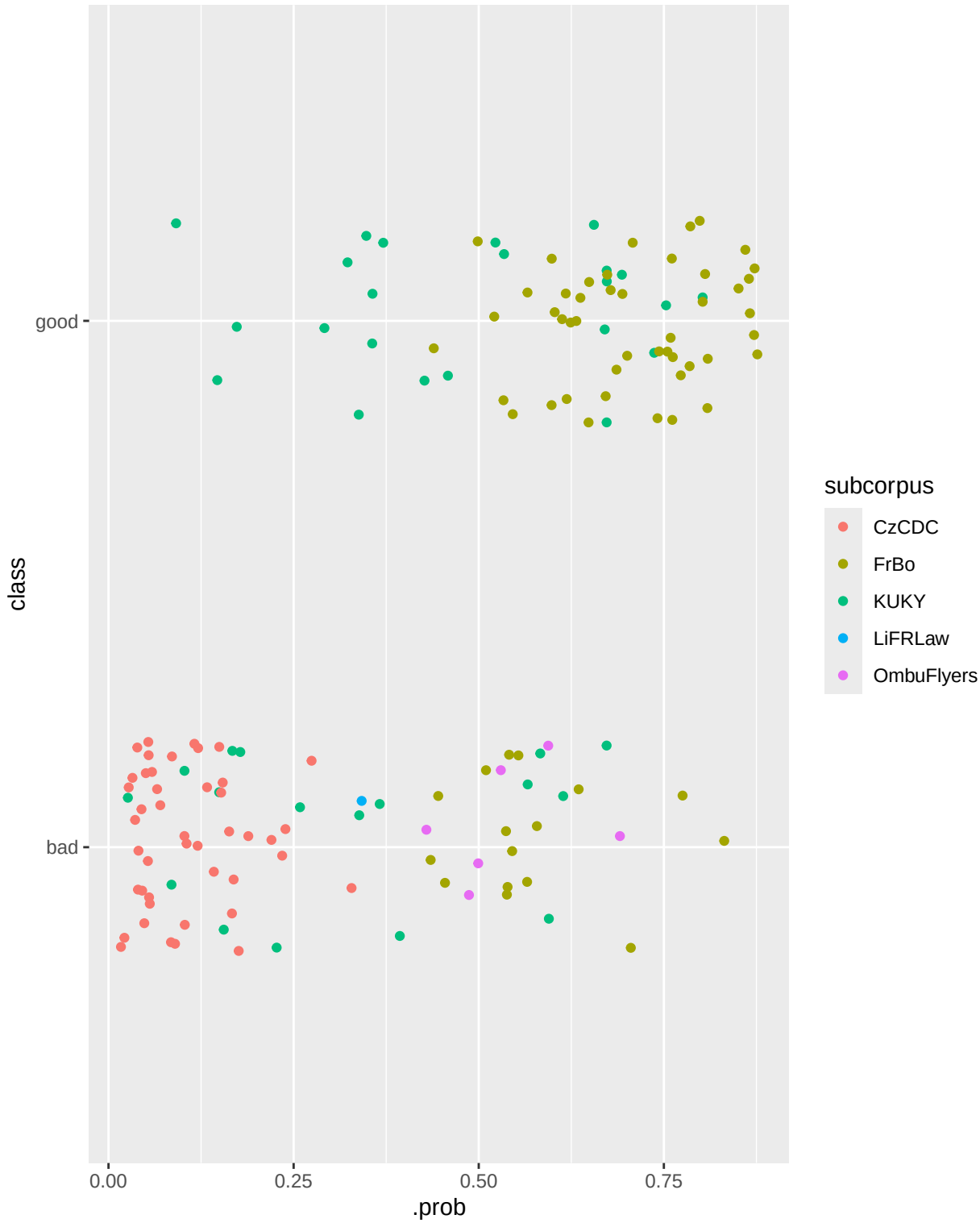
```
cimportances_all %>%
  abs() %>%
  sort(decreasing = TRUE) %>%
  as.data.frame() %>%
  print(digits = 3)
```

```
##                          .
## VERBfrac.m          5.78e-04
## activity            4.07e-04
## literary            3.20e-04
## verbdist            2.82e-04
## passives            2.34e-04
## atl                 2.07e-04
## anaphoricrefs       1.74e-04
## ari                 1.52e-04
## gf                  1.46e-04
## sentlen.m           1.43e-04
## predorder.m         1.25e-04
## entropy.v           1.01e-04
## NOUNcount.m         1.00e-04
## sentlen.v           9.78e-05
## longexprs           9.39e-05
## cli                 8.52e-05
## mamr                8.46e-05
## NOUNfrac.m          7.52e-05
## NOUNcount.v         7.47e-05
## smog                6.96e-05
## ttr                 6.44e-05
```

```
## ttr.v               6.31e-05
## GPdeverbsubj        5.99e-05
## fkgl                4.91e-05
## mattr               4.67e-05
## subj                4.63e-05
## obj                 4.53e-05
## xcompdist.m         3.84e-05
## doubleADPs          3.45e-05
## analyticVERBsdist.m 3.45e-05
## relativisticexprs   3.10e-05
## compoundVERBs       3.02e-05
## caserepcount.v      2.83e-05
## maentropy           2.79e-05
## verbalNOUNs         2.79e-05
## rfpass_animsubj     2.71e-05
## NEGfrac.v           2.49e-05
## VERBfrac.v          2.06e-05
## weakmeaning         1.92e-05
## abstractNOUNs       1.77e-05
## GPcoordovs          1.70e-05
## caserepcount.m      1.68e-05
## xcompdist.v         1.63e-05
## NEGfrac.m           1.53e-05
## xcomp               1.51e-05
## predorder.v         1.41e-05
## predsubjdist.v      1.34e-05
## extrcaseexprs       1.28e-05
## NEGcount.v          1.27e-05
## entropy             1.13e-05
## GPpatinstr          1.01e-05
## predobjdist.m       9.32e-06
## GPdeverbaddr        8.73e-06
## hapaxes             7.47e-06
## NOUNfrac.v          7.05e-06
## doubleADPdist.m     6.40e-06
## hpoint              4.17e-06
## predobjdist.v       3.87e-06
## fre                 3.87e-06
## doubleADPdist.v     3.68e-06
## GPwordorder         3.07e-06
## analyticVERBsdist.v 2.16e-06
## GPpatbenperson      1.78e-06
## NEGcount.m          1.20e-06
## predsubjdist.m      1.02e-06
## redundexprs         2.83e-07
## GPadjective         0.00e+00
```

```
testing_set_all <- testing_set %>%
  mutate(.prob = predictions_all_prob, .pred = predictions_all)
```

```
mismatches_all <- get_mismatch_details(testing_set_all)
```

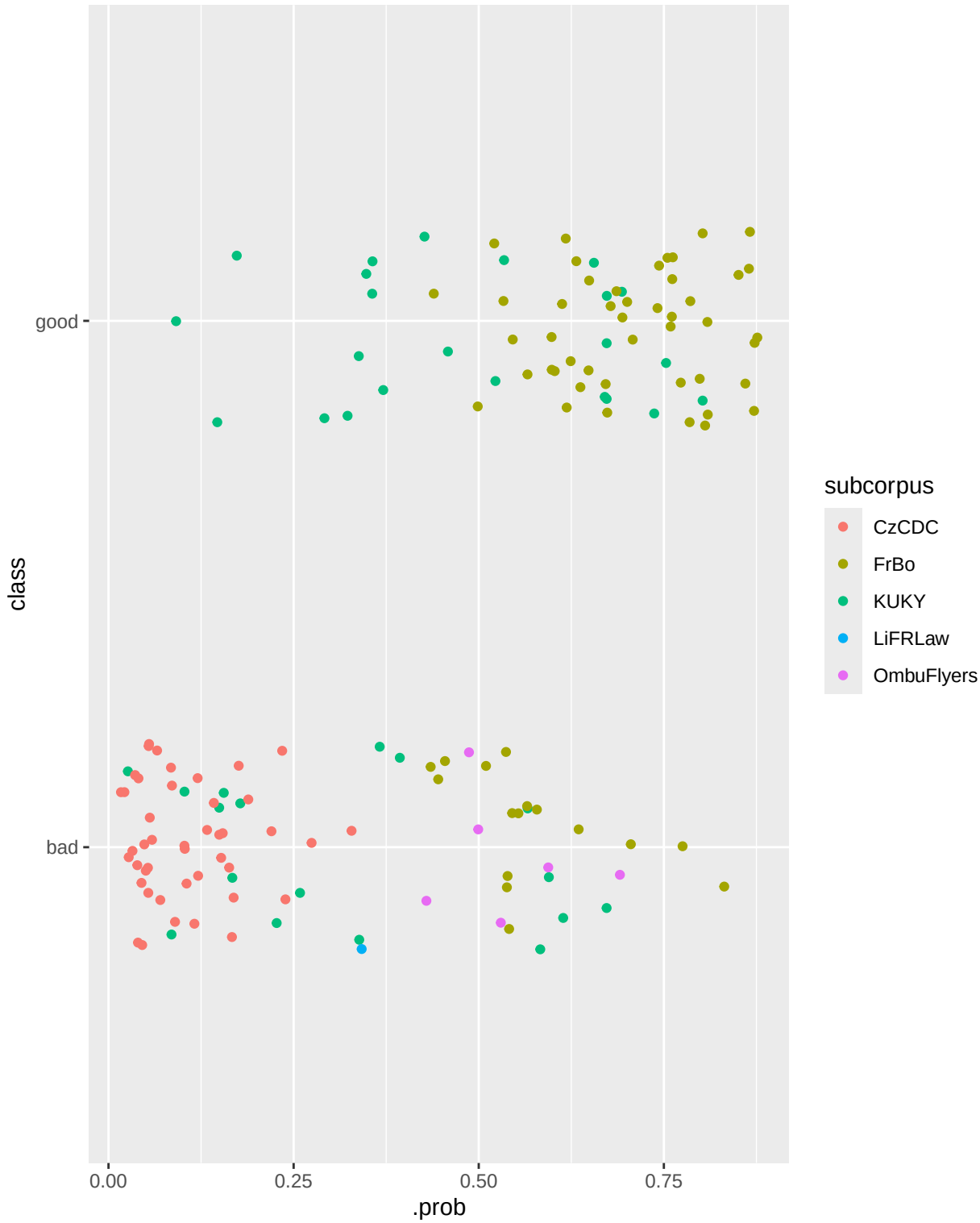



```
## Confusion matrices by subcorpora:
## , , subcorpus = CzCDC
##
##       class
## .pred  bad good
##   bad   44    0
```

```
##   good   0    0
##
## , , subcorpus = FrBo
##
##       class
## .pred  bad good
##   bad    3    2
##   good  13   44
##
## , , subcorpus = KUKY
##
##       class
## .pred  bad good
##   bad   12   12
##   good   5   11
##
## , , subcorpus = LiFRLaw
##
##       class
## .pred  bad good
##   bad    1    0
##   good   0    0
##
## , , subcorpus = OmbuFlyers
##
##       class
## .pred  bad good
##   bad    3    0
##   good   3    0
##
##
## Greatest deviations:
## # A tibble: 35 x 5
##    abs_dev .prob class subcorpus  FileName
##      <dbl> <dbl> <fct> <chr>      <chr>
##  1   0.409 0.091 good  KUKY       0217_6Afs_2000035_20210219141328__1_
##  2   0.353 0.147 good  KUKY       11_vizum_pred
##  3   0.332 0.832 bad   FrBo       orig_Co můžete dělat, pokud obec postupuje př~
##  4   0.327 0.173 good  KUKY       Odvolani
##  5   0.275 0.775 bad   FrBo       orig_Jak probíhá správní řízení
##  6   0.208 0.292 good  KUKY       Mestsky_urad_usneseni_-_slouceni_pred
##  7   0.205 0.705 bad   FrBo       orig_Jak se bránit neposkytnutí projektové do~
##  8   0.191 0.691 bad   OmbuFlyers Soudni-poplatky
##  9   0.177 0.323 good  KUKY       Mestsky_urad_Souhlas_s_prestupkovym_rizenim
## 10   0.173 0.673 bad   KUKY       043_Plisen-a-zavady-v-byte
## 11   0.162 0.338 good  KUKY       Odvolani_proti_rozhodnuti_o_nepovoleni_kaceni
## 12   0.152 0.348 good  KUKY       1A_dokument_puvodni_ustanoven_zastupce_vyzva_~
## 13   0.144 0.356 good  KUKY       Mestsky_urad_Usneseni_narizeni_podrobit_se_pr~
## 14   0.143 0.357 good  KUKY       2A_dokument_puvodni_vyzva_k_zaplaceni_SOP_a_k~
## 15   0.135 0.635 bad   FrBo       orig_Kompletní průvodce občana obtěžovaného h~
## 16   0.129 0.371 good  KUKY       Mestsky_urad_kontrola_po
## 17   0.114 0.614 bad   KUKY       PR_Masinova
## 18   0.095 0.595 bad   KUKY       024_Opatrovnictvi
## 19   0.094 0.594 bad   OmbuFlyers Detsky-domov
```

```
## 20   0.083 0.583 bad   KUKY       sluzebni_hodnoceni_puvodni
## 21   0.079 0.579 bad   FrBo       red_Smlouvy obcí s investory
## 22   0.073 0.427 good  KUKY       Zaloba_na_zruseni_spoluvlastnictvi
## 23   0.066 0.566 bad   KUKY       41_A_32-2022_rozsudek_Martina_Kopy_Anna_Rybar~
## 24   0.065 0.565 bad   FrBo       170
## 25   0.061 0.439 good  FrBo       red_provokace_korupcniho_jednani
## 26   0.054 0.554 bad   FrBo       68
## 27   0.045 0.545 bad   FrBo       orig_Jaké trestné činy mohou souviset s korup~
## 28   0.042 0.458 good  KUKY       857_2024_VOP
## 29   0.041 0.541 bad   FrBo       27
## 30   0.039 0.539 bad   FrBo       orig_Jak probíhá trestní řízení
## 31   0.038 0.538 bad   FrBo       153
## # i 4 more rows
## Names of highest-deviating documents:
## [1] "0217_6Afs_2000035_20210219141328__1_"
## [2] "11_vizum_pred"
## [3] "orig_Co můžete dělat, pokud obec postupuje při prodeji nebo pronájmu pozemků nezákonně_final"
## [4] "Odvolani"
## [5] "orig_Jak probíhá správní řízení"
```

```
mismatches_all <- get_mismatch_details(testing_set_all)
```



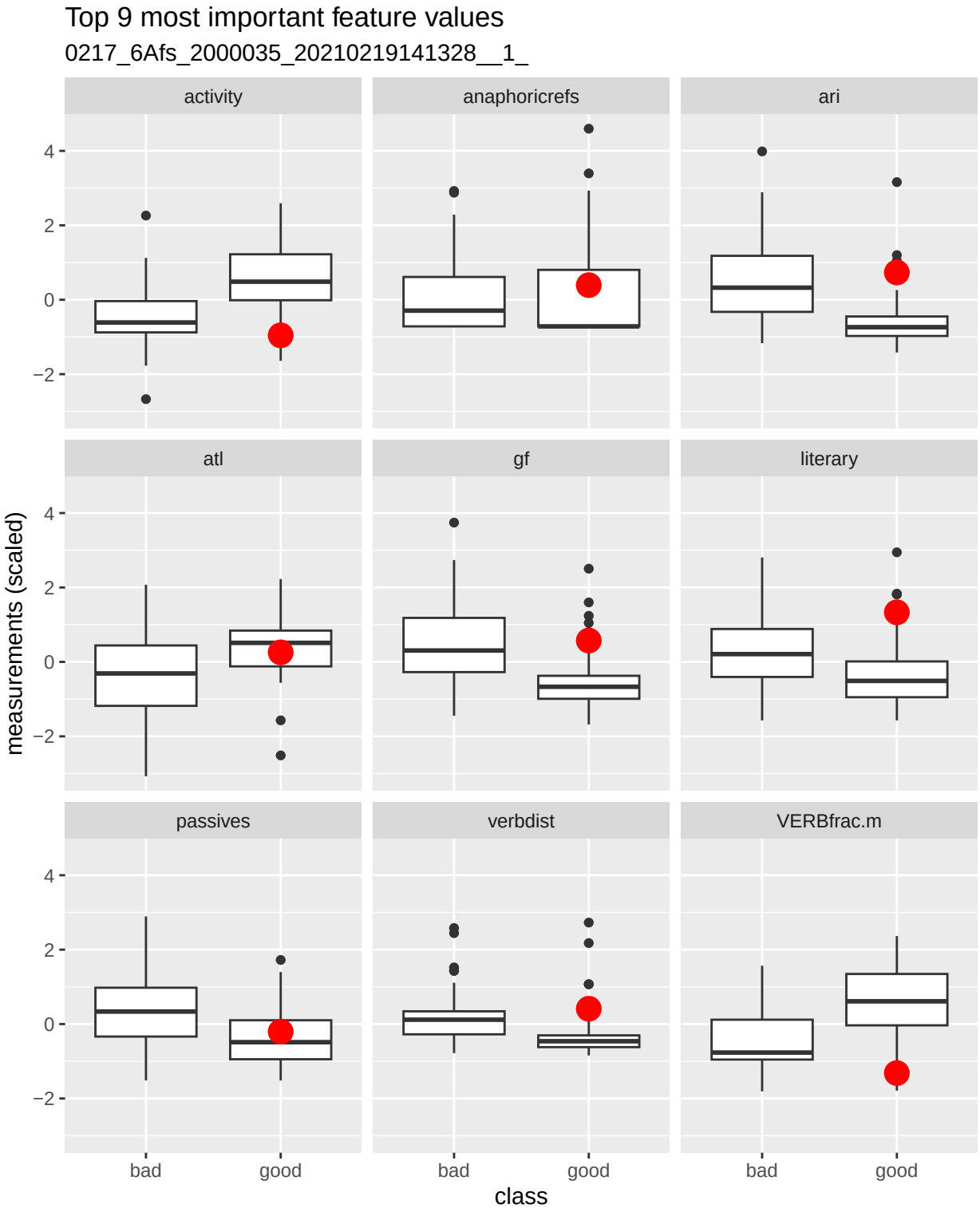

```
## Confusion matrices by subcorpora:
## , , subcorpus = CzCDC
##
##       class
## .pred  bad good
##   bad   44    0
```

```
##   good   0    0
##
## , , subcorpus = FrBo
##
##       class
## .pred  bad good
##   bad    3    2
##   good  13   44
##
## , , subcorpus = KUKY
##
##       class
## .pred  bad good
##   bad   12   12
##   good   5   11
##
## , , subcorpus = LiFRLaw
##
##       class
## .pred  bad good
##   bad    1    0
##   good   0    0
##
## , , subcorpus = OmbuFlyers
##
##       class
## .pred  bad good
##   bad    3    0
##   good   3    0
##
##
## Greatest deviations:
## # A tibble: 35 x 5
##    abs_dev .prob class subcorpus  FileName
##      <dbl> <dbl> <fct> <chr>      <chr>
##  1   0.409 0.091 good  KUKY       0217_6Afs_2000035_20210219141328__1_
##  2   0.353 0.147 good  KUKY       11_vizum_pred
##  3   0.332 0.832 bad   FrBo       orig_Co můžete dělat, pokud obec postupuje př~
##  4   0.327 0.173 good  KUKY       Odvolani
##  5   0.275 0.775 bad   FrBo       orig_Jak probíhá správní řízení
##  6   0.208 0.292 good  KUKY       Mestsky_urad_usneseni_-_slouceni_pred
##  7   0.205 0.705 bad   FrBo       orig_Jak se bránit neposkytnutí projektové do~
##  8   0.191 0.691 bad   OmbuFlyers Soudni-poplatky
##  9   0.177 0.323 good  KUKY       Mestsky_urad_Souhlas_s_prestupkovym_rizenim
## 10   0.173 0.673 bad   KUKY       043_Plisen-a-zavady-v-byte
## 11   0.162 0.338 good  KUKY       Odvolani_proti_rozhodnuti_o_nepovoleni_kaceni
## 12   0.152 0.348 good  KUKY       1A_dokument_puvodni_ustanoven_zastupce_vyzva_~
## 13   0.144 0.356 good  KUKY       Mestsky_urad_Usneseni_narizeni_podrobit_se_pr~
## 14   0.143 0.357 good  KUKY       2A_dokument_puvodni_vyzva_k_zaplaceni_SOP_a_k~
## 15   0.135 0.635 bad   FrBo       orig_Kompletní průvodce občana obtěžovaného h~
## 16   0.129 0.371 good  KUKY       Mestsky_urad_kontrola_po
## 17   0.114 0.614 bad   KUKY       PR_Masinova
## 18   0.095 0.595 bad   KUKY       024_Opatrovnictvi
## 19   0.094 0.594 bad   OmbuFlyers Detsky-domov
```

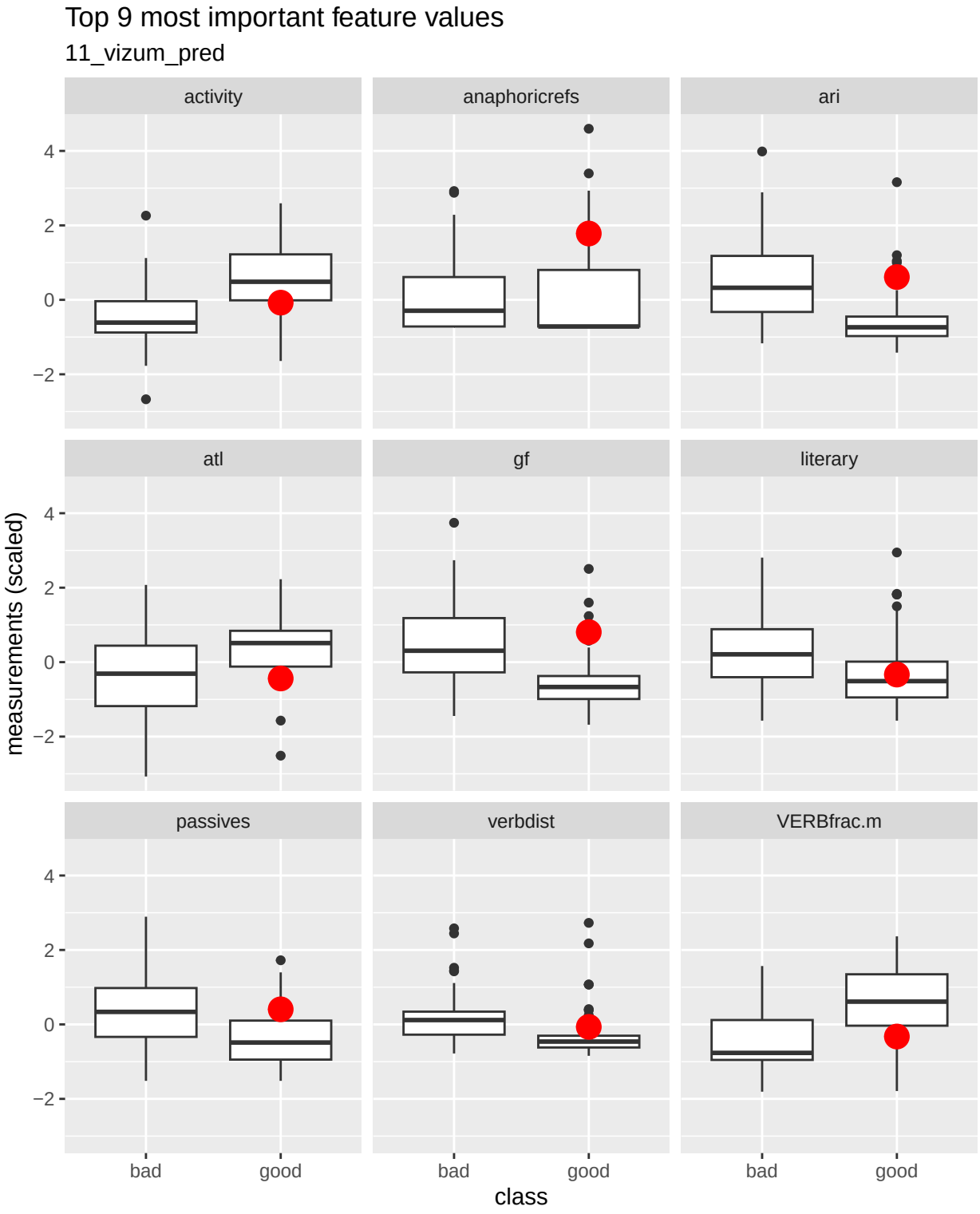
```
## 20   0.083 0.583 bad   KUKY       sluzebni_hodnoceni_puvodni
## 21   0.079 0.579 bad   FrBo       red_Smlouvy obcí s investory
## 22   0.073 0.427 good  KUKY       Zaloba_na_zruseni_spoluvlastnictvi
## 23   0.066 0.566 bad   KUKY       41_A_32-2022_rozsudek_Martina_Kopy_Anna_Rybar~
## 24   0.065 0.565 bad   FrBo       170
## 25   0.061 0.439 good  FrBo       red_provokace_korupcniho_jednani
## 26   0.054 0.554 bad   FrBo       68
## 27   0.045 0.545 bad   FrBo       orig_Jaké trestné činy mohou souviset s korup~
## 28   0.042 0.458 good  KUKY       857_2024_VOP
## 29   0.041 0.541 bad   FrBo       27
## 30   0.039 0.539 bad   FrBo       orig_Jak probíhá trestní řízení
## 31   0.038 0.538 bad   FrBo       153
## # i 4 more rows
## Names of highest-deviating documents:
## [1] "0217_6Afs_2000035_20210219141328__1_"
## [2] "11_vizum_pred"
## [3] "orig_Co můžete dělat, pokud obec postupuje při prodeji nebo pronájmu pozemků nezákonně_final"
## [4] "Odvolani"
## [5] "orig_Jak probíhá správní řízení"
```

```
for (dev in mismatches_all$highest_deviations) {
  print(plot_outlier(dev, cimportances_all, testing_set_all) +
    labs(title = "Top 9 most important feature values", subtitle = dev))
}
```

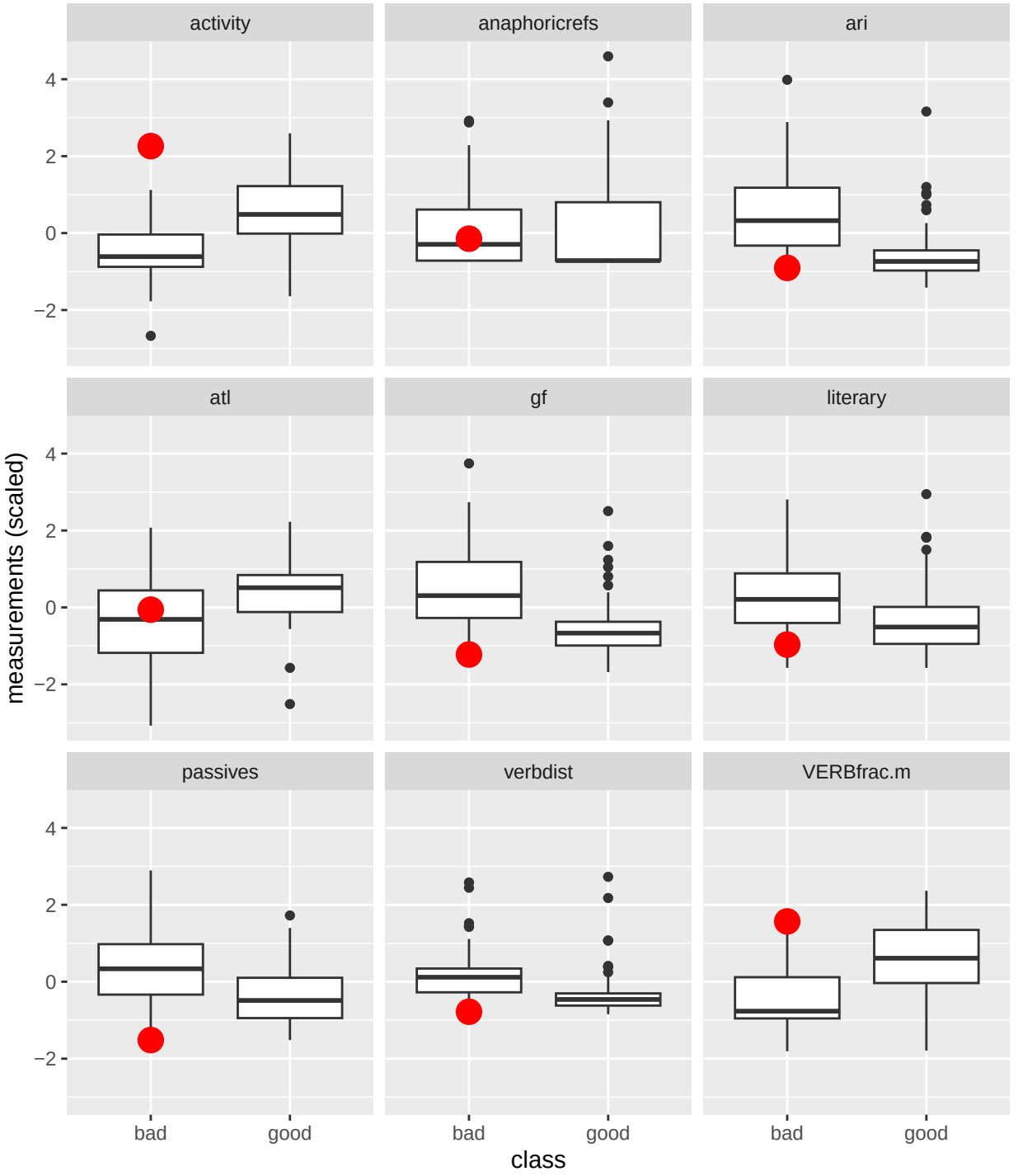
```
## 1 observation(s) removed from the plot
```



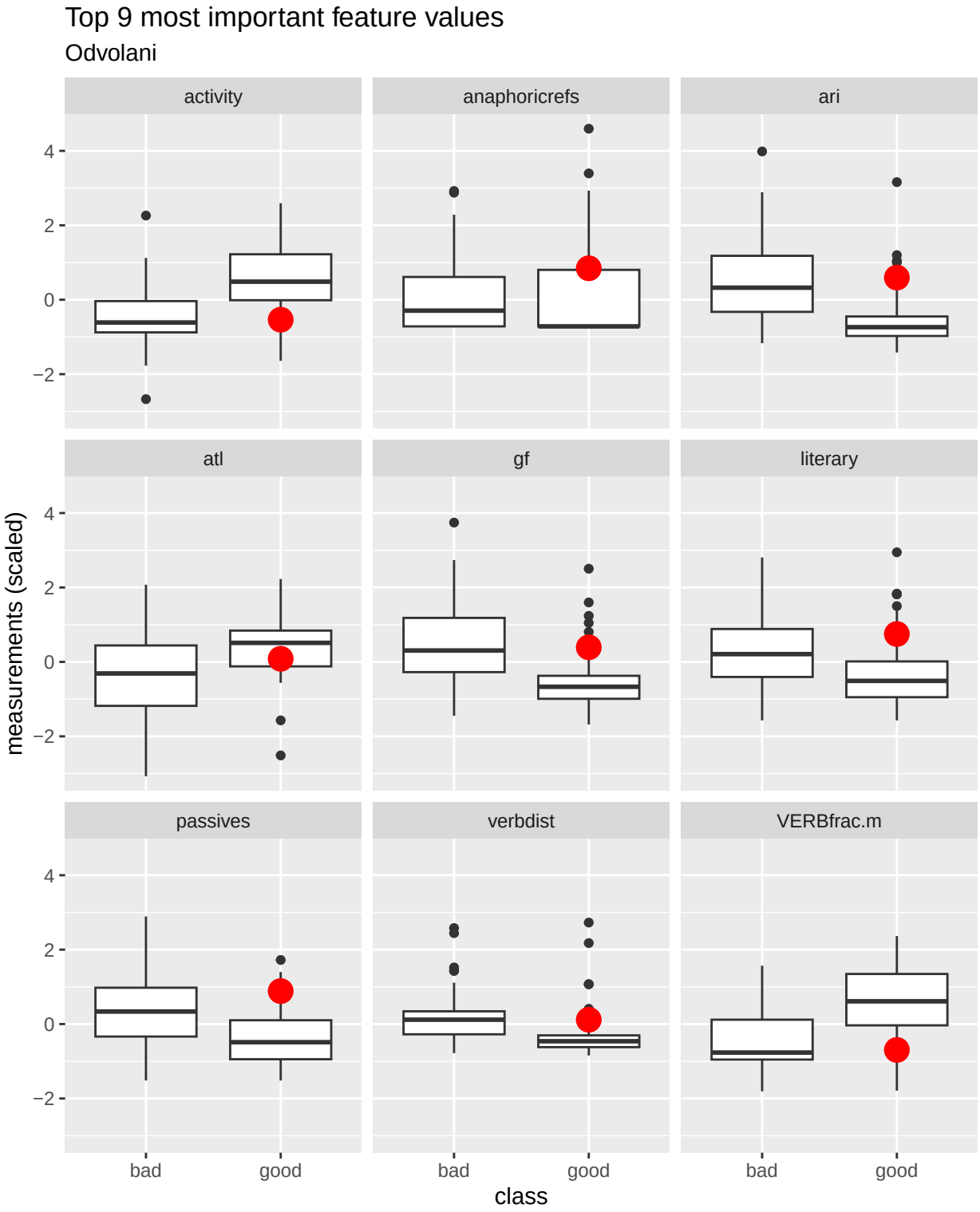

```
## 1 observation(s) removed from the plot
```



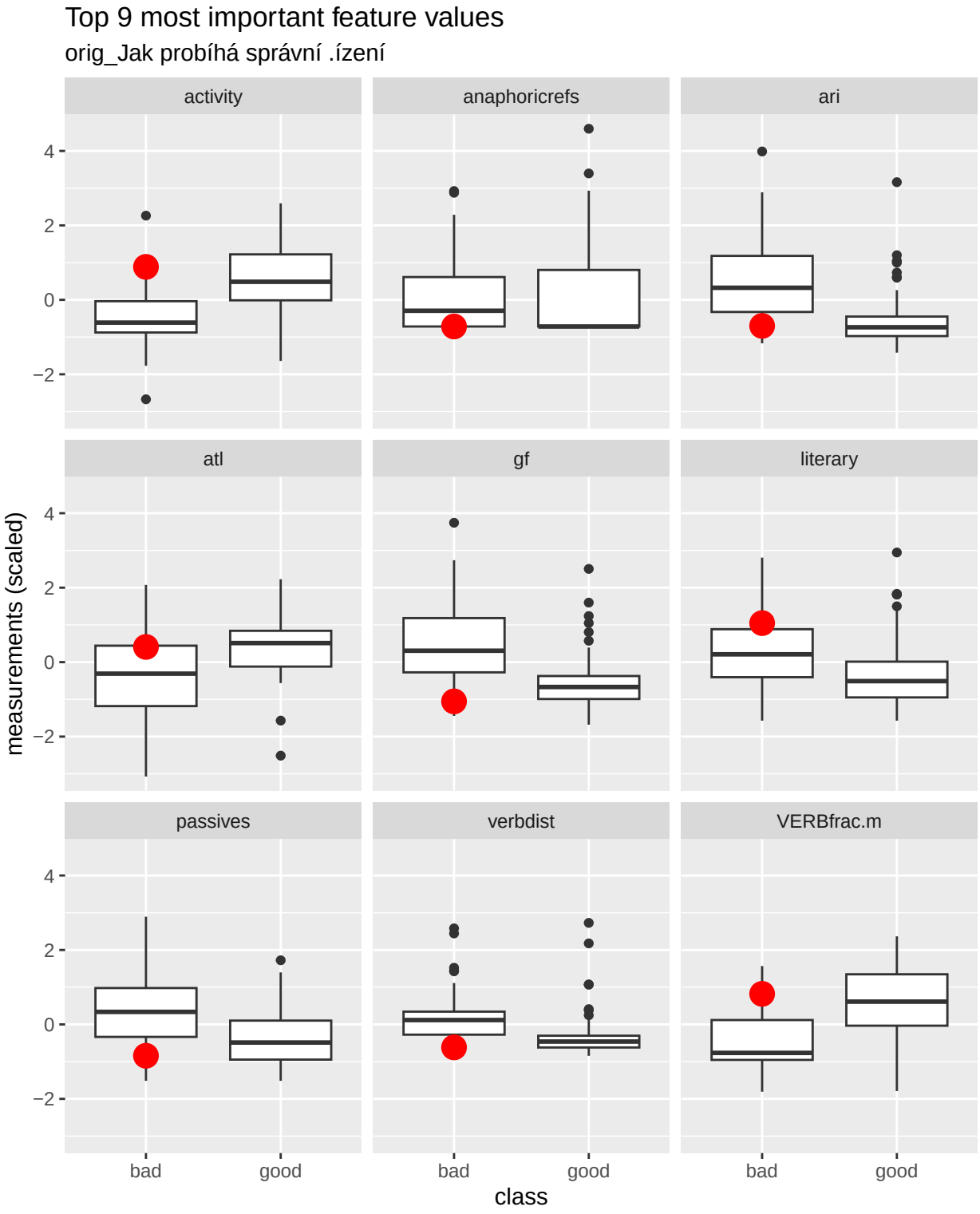

```
## 1 observation(s) removed from the plot
```




```
## 1 observation(s) removed from the plot
```




```
## 1 observation(s) removed from the plot
```



Variable importance comparison

```
glm_feature_importances <- tibble(
  feat_name = character(), p_value_glm = numeric()
```

```
)

for (i in .firstnonmetacolumn:ncol(data_clean)) {
  fname <- names(data_clean)[i]

  formula_single <- reformulate(fname, "class")

  glm_model <- glm(formula_single, data_clean, family = "binomial")
  glm_coefficients <- summary(glm_model)$coefficients
  row_index <- which(rownames(glm_coefficients) == fname)
  p_value <- glm_coefficients[row_index, 4]

  glm_feature_importances <- glm_feature_importances %>%
    add_row(feat_name = fname, p_value_glm = p_value)
}
glm_feature_importances
```

```
## # A tibble: 67 x 2
##    feat_name       p_value_glm
##    <chr>                 <dbl>
##  1 abstractNOUNs    0.00187
##  2 anaphoricrefs    0.660
##  3 caserepcount.m   0.0722
##  4 caserepcount.v   0.00479
##  5 extrcaseexprs    0.0985
##  6 doubleADPs       0.312
##  7 doubleADPdist.m  0.000154
##  8 doubleADPdist.v  0.00000356
##  9 GPadjective      0.380
## 10 GPcoordovs       0.828
## # i 57 more rows
```

```
feature_importances <- glm_feature_importances %>%
  full_join(
    data.frame(
      feat_name = names(cimportances_all) %>% prettify_feat_name_vector(),
      imp_crf_all = as.vector(cimportances_all)
    ),
    by = "feat_name"
  ) %>%
  full_join(
    data.frame(
      feat_name = names(cimportances_sel) %>% prettify_feat_name_vector(),
      imp_crf_sel = as.vector(cimportances_sel)
    ),
    by = "feat_name"
  ) %>%
  mutate(imp_crf_all_abs = abs(imp_crf_all), imp_crf_sel_abs = abs(imp_crf_sel))

feature_importances %>%
  select(feat_name, p_value_glm, imp_crf_all_abs, imp_crf_sel_abs) %>%
  arrange(p_value_glm) %>%
  as.data.frame() %>%
  print(digits = 3)
```

```
##              feat_name p_value_glm imp_crf_all_abs imp_crf_sel_abs
## 1             activity    2.48e-34        4.07e-04        7.85e-04
## 2           VERBfrac.m    2.49e-34        5.78e-04        2.19e-04
## 3                 smog    3.33e-33        6.96e-05        5.02e-05
## 4                   gf    8.87e-33        1.46e-04        7.45e-05
## 5                  ari    5.43e-32        1.52e-04        1.41e-04
## 6            sentlen.m    7.15e-31        1.43e-04        3.49e-04
## 7                 fkgl    1.75e-28        4.91e-05        6.01e-05
## 8                 mamr    1.68e-27        8.46e-05        9.23e-05
## 9          NOUNcount.m    5.43e-25        1.00e-04        1.90e-04
## 10            verbdist    6.49e-24        2.82e-04        4.27e-04
## 11                 atl    5.62e-22        2.07e-04        1.24e-04
## 12            literary    6.10e-21        3.20e-04        4.81e-04
## 13            passives    6.86e-20        2.34e-04        2.65e-04
## 14         predorder.m    7.63e-19        1.25e-04        1.05e-04
## 15       compoundVERBs    1.34e-18        3.02e-05        1.00e-05
## 16               xcomp    3.55e-15        1.51e-05        5.17e-05
## 17         NOUNcount.v    5.07e-15        7.47e-05        9.04e-05
## 18                subj    1.22e-14        4.63e-05        7.19e-05
## 19           maentropy    1.32e-14        2.79e-05        1.94e-05
## 20      predsubjdist.v    1.73e-14        1.34e-05        1.33e-05
## 21               mattr    1.92e-13        4.67e-05        1.51e-04
## 22          NEGcount.m    4.70e-13        1.20e-06        6.19e-05
## 23         predorder.v    3.14e-12        1.41e-05        9.07e-06
## 24                 cli    1.92e-11        8.52e-05        4.80e-05
## 25                 fre    4.45e-11        3.87e-06        1.34e-05
## 26             entropy    7.61e-09        1.13e-05        7.67e-05
## 27 analyticVERBsdist.v    5.59e-08        2.16e-06        8.82e-05
## 28       predobjdist.v    4.44e-07        3.87e-06        3.98e-05
## 29           NEGfrac.m    6.28e-07        1.53e-05        1.79e-05
## 30          NEGcount.v    1.26e-06        1.27e-05        5.64e-05
## 31         GPwordorder    2.46e-06        3.07e-06        1.32e-05
## 32          NOUNfrac.v    3.11e-06        7.05e-06        5.17e-05
## 33     doubleADPdist.v    3.56e-06        3.68e-06        7.00e-05
## 34          VERBfrac.v    6.24e-06        2.06e-05        3.34e-05
## 35      predsubjdist.m    6.00e-05        1.02e-06        3.66e-05
## 36         verbalNOUNs    7.48e-05        2.79e-05        1.47e-04
## 37       predobjdist.m    9.25e-05        9.32e-06        1.14e-05
## 38              hpoint    1.28e-04        4.17e-06        3.09e-06
## 39     doubleADPdist.m    1.54e-04        6.40e-06        1.48e-05
## 40                 obj    5.14e-04        4.53e-05        1.86e-05
## 41       abstractNOUNs    1.87e-03        1.77e-05        6.84e-05
## 42   relativisticexprs    2.05e-03        3.10e-05        3.75e-05
## 43 analyticVERBsdist.m    3.20e-03        3.45e-05        2.41e-05
## 44      caserepcount.v    4.79e-03        2.83e-05        2.67e-05
## 45         redundexprs    1.04e-02        2.83e-07        0.00e+00
## 46        GPdeverbaddr    1.12e-02        8.73e-06        9.06e-05
## 47        GPdeverbsubj    1.33e-02        5.99e-05        3.50e-05
## 48             hapaxes    1.39e-02        7.47e-06        1.01e-06
## 49           NEGfrac.v    3.65e-02        2.49e-05        4.33e-05
## 50         weakmeaning    3.86e-02        1.92e-05        1.97e-05
## 51           entropy.v    4.83e-02        1.01e-04        1.93e-04
## 52                 ttr    6.11e-02        6.44e-05              NA
## 53     rfpass_animsubj    6.83e-02        2.71e-05              NA
```

```
## 54      caserepcount.m    7.22e-02        1.68e-05              NA
## 55         xcompdist.v    7.55e-02        1.63e-05              NA
## 56       extrcaseexprs    9.85e-02        1.28e-05              NA
## 57         xcompdist.m    1.03e-01        3.84e-05              NA
## 58           longexprs    3.11e-01        9.39e-05              NA
## 59          doubleADPs    3.12e-01        3.45e-05              NA
## 60          GPpatinstr    3.72e-01        1.01e-05              NA
## 61         GPadjective    3.80e-01        0.00e+00              NA
## 62      GPpatbenperson    4.14e-01        1.78e-06              NA
## 63       anaphoricrefs    6.60e-01        1.74e-04              NA
## 64          NOUNfrac.m    7.33e-01        7.52e-05              NA
## 65           sentlen.v    7.92e-01        9.78e-05              NA
## 66               ttr.v    7.95e-01        6.31e-05              NA
## 67          GPcoordovs    8.28e-01        1.70e-05              NA
```

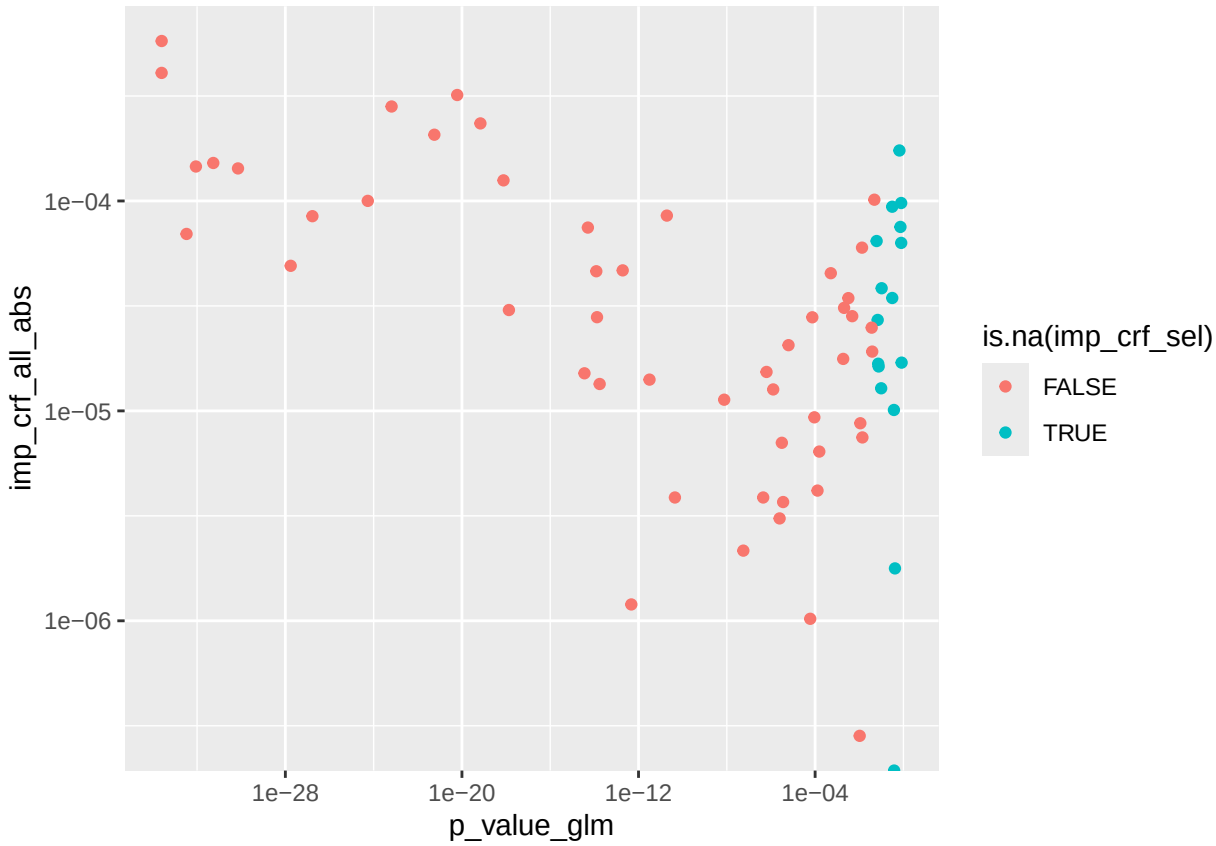
```
feature_importances %>%
  select(feat_name, imp_crf_all, imp_crf_sel) %>%
  arrange(imp_crf_all) %>%
  as.data.frame() %>%
  print(digits = 3)
```

```
##              feat_name imp_crf_all imp_crf_sel
## 1           doubleADPs   -3.45e-05          NA
## 2    relativisticexprs   -3.10e-05    3.75e-05
## 3            NEGfrac.v   -2.49e-05   -4.33e-05
## 4           GPcoordovs   -1.70e-05          NA
## 5       caserepcount.m   -1.68e-05          NA
## 6          xcompdist.v   -1.63e-05          NA
## 7                xcomp   -1.51e-05   -5.17e-05
## 8       predsubjdist.v   -1.34e-05   -1.33e-05
## 9        extrcaseexprs   -1.28e-05          NA
## 10          NEGcount.v   -1.27e-05   -5.64e-05
## 11          GPpatinstr   -1.01e-05          NA
## 12       predobjdist.m   -9.32e-06   -1.14e-05
## 13          NOUNfrac.v   -7.05e-06   -5.17e-05
## 14     doubleADPdist.m   -6.40e-06    1.48e-05
## 15              hpoint   -4.17e-06    3.09e-06
## 16       predobjdist.v   -3.87e-06   -3.98e-05
## 17                 fre   -3.87e-06    1.34e-05
## 18     doubleADPdist.v   -3.68e-06    7.00e-05
## 19 analyticVERBsdist.v   -2.16e-06    8.82e-05
## 20      GPpatbenperson   -1.78e-06          NA
## 21          NEGcount.m   -1.20e-06    6.19e-05
## 22      predsubjdist.m   -1.02e-06   -3.66e-05
## 23         GPadjective    0.00e+00          NA
## 24         redundexprs    2.83e-07    0.00e+00
## 25         GPwordorder    3.07e-06   -1.32e-05
## 26             hapaxes    7.47e-06   -1.01e-06
## 27        GPdeverbaddr    8.73e-06    9.06e-05
## 28             entropy    1.13e-05    7.67e-05
## 29         predorder.v    1.41e-05   -9.07e-06
## 30           NEGfrac.m    1.53e-05    1.79e-05
## 31       abstractNOUNs    1.77e-05    6.84e-05
## 32         weakmeaning    1.92e-05    1.97e-05
## 33          VERBfrac.v    2.06e-05    3.34e-05
```

```
## 34     rfpass_animsubj    2.71e-05          NA
## 35         verbalNOUNs    2.79e-05    1.47e-04
## 36           maentropy    2.79e-05    1.94e-05
## 37      caserepcount.v    2.83e-05   -2.67e-05
## 38       compoundVERBs    3.02e-05    1.00e-05
## 39 analyticVERBsdist.m    3.45e-05    2.41e-05
## 40         xcompdist.m    3.84e-05          NA
## 41                 obj    4.53e-05   -1.86e-05
## 42                subj    4.63e-05    7.19e-05
## 43               mattr    4.67e-05    1.51e-04
## 44                fkgl    4.91e-05    6.01e-05
## 45        GPdeverbsubj    5.99e-05    3.50e-05
## 46               ttr.v    6.31e-05          NA
## 47                 ttr    6.44e-05          NA
## 48                smog    6.96e-05    5.02e-05
## 49         NOUNcount.v    7.47e-05    9.04e-05
## 50          NOUNfrac.m    7.52e-05          NA
## 51                mamr    8.46e-05    9.23e-05
## 52                 cli    8.52e-05    4.80e-05
## 53           longexprs    9.39e-05          NA
## 54           sentlen.v    9.78e-05          NA
## 55         NOUNcount.m    1.00e-04    1.90e-04
## 56           entropy.v    1.01e-04    1.93e-04
## 57         predorder.m    1.25e-04    1.05e-04
## 58           sentlen.m    1.43e-04    3.49e-04
## 59                  gf    1.46e-04    7.45e-05
## 60                ari    1.52e-04    1.41e-04
## 61       anaphoricrefs    1.74e-04          NA
## 62                atl    2.07e-04    1.24e-04
## 63            passives    2.34e-04    2.65e-04
## 64            verbdist    2.82e-04    4.27e-04
## 65            literary    3.20e-04    4.81e-04
## 66            activity    4.07e-04    7.85e-04
## 67          VERBfrac.m    5.78e-04    2.19e-04
```

```
feature_importances %>%
  ggplot(aes(
    x = p_value_glm, y = imp_crf_all_abs, color = is.na(imp_crf_sel)
  )) +
  geom_point() +
  scale_x_log10() +
  scale_y_log10()
```

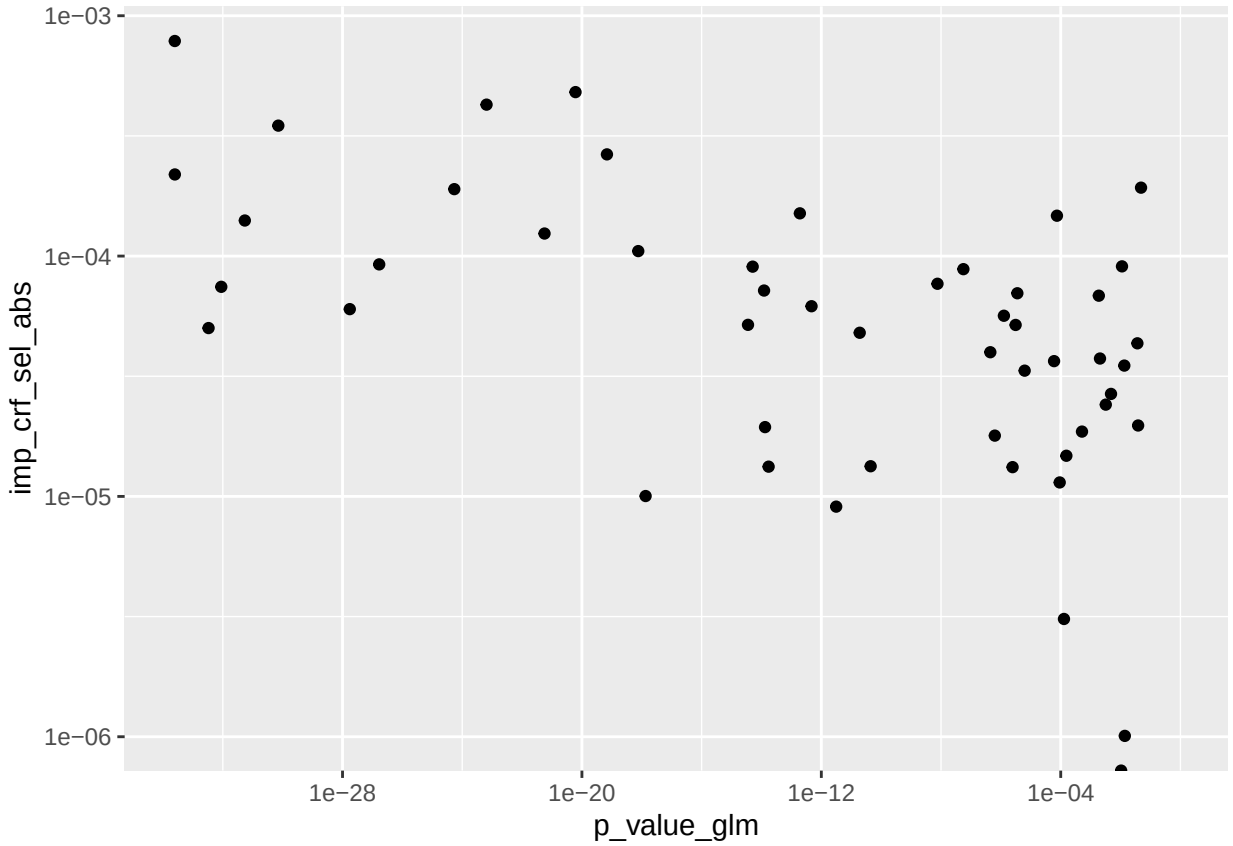
```
## Warning in scale_y_log10(): log-10 transformation introduced infinite values.
```




```
feature_importances %>%
  ggplot(aes(x = p_value_glm, y = imp_crf_sel_abs)) +
  geom_point() +
  scale_x_log10() +
  scale_y_log10()
```

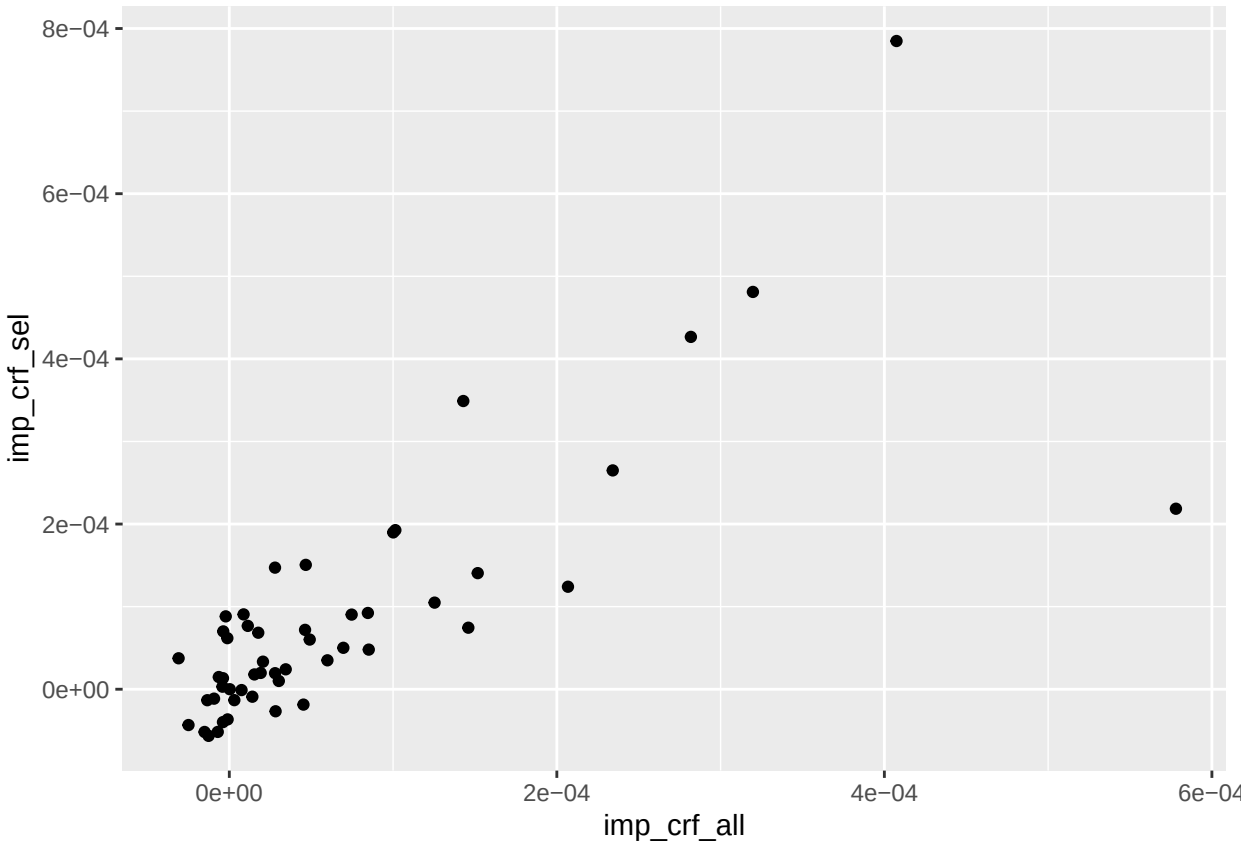
```
## Warning in scale_y_log10(): log-10 transformation introduced infinite values.
```

```
## Warning: Removed 16 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```
feature_importances %>%
  ggplot(aes(x = imp_crf_all, y = imp_crf_sel)) +
  geom_point()
```

```
## Warning: Removed 16 rows containing missing values or values outside the scale range
## (`geom_point()`).
```


8e−04
6e−04
4e−04
2e−04
0e+00
imp_crf_sel
0e+00
2e−04
4e−04
6e−04
imp_crf_all