

Experiment 1

A & S

2023-08-21

```
max_s_opts = c(5, 10, 20, 30, 40, 113)
#max_s_opts = c(5, 10)
nfolds = 5

# load RENT features
RF <- read.csv(paste0(path, "data/RENT_features_all.csv"))
RF$index = RF$index + 1 # Python-R-conversion of indices

RF_all <- data.frame(index = rep(c(1:113), 5), max_s = rep(113, 113*113*5), fold = rep(1:5, each = 113))
colnames(RF) = c("index", "max_s", "fold")
RF = rbind(RF, RF_all)

feature_df <- cbind(RF, fs = "RENT")

# train UBayFS models to obtain features
system.time(
for (max_s in max_s_opts) {
  set.seed(1)
  for (test_fold in 1:nfolds) {
    block_const <- UBayFS::buildConstraints(constraint_types = c("max_size"),
                                           constraint_vars = max_s,
                                           num_elements = ncol(data_list[[test_fold]]$train_data),
                                           rho = Inf)

    model = train_UBay_model(data = data_list[[test_fold]]$train_data,
                             target = as.numeric(data_list[[test_fold]]$train_target$category),
                             M = 100,
                             tt_split = 0.75,
                             nf = max_s,
                             weights = weights,
                             lam = 10,
                             block_constraints = list(block_const))

    features = which(model$output$feature_set[1,] == 1)
    feature_df <- rbind(feature_df, data.frame(index = features,
                                                max_s = max_s,
                                                fold = test_fold,
                                                fs = "UBayFS"))
  }
})
```

```
##          User          System verstrichen
```

```
##          32.64          1.31          87.08
```

```
save(feature_df, file = paste0(path, "data/exp1_features.Rdata"))
```

```
pred_df <- data.frame()
RED_df <- data.frame()
stability_df <- data.frame()
param_df <- data.frame()
```

```
# signs within elementary models
```

```
feature_df <- cbind(feature_df, sign = 0)
```

```
# PREDICTIONS
```

```
for(fs_ in c("RENT", "UBayFS")){
  for (max_s_ in max_s_opts) {
    set.seed(1)
    for (test_fold in 1:nfolds) {

      features = subset(feature_df, (max_s == max_s_) & (fold == test_fold) & (fs == fs_))$index
      eval <- eval_all(features,
                        train_data = data_list[[test_fold]]$train_data,
                        test_data = data_list[[test_fold]]$test_data,
                        train_target = data_list[[test_fold]]$train_target,
                        test_target = data_list[[test_fold]]$test_target#,
                        #fixed_param = 3)

      )

      pred_df <- rbind(pred_df,
                      cbind(eval$res,
                            max_s = max_s_,
                            fold = test_fold,
                            fs = fs_))

      param_df <- rbind(param_df,
                        data.frame(param = eval$param,
                                  method = names(eval$param),
                                  max_s = max_s_,
                                  fold = test_fold,
                                  fs = fs_))

      feat_inds <- (feature_df$index %in% features &
                   feature_df$max_s == max_s_ &
                   feature_df$fold == test_fold &
                   feature_df$fs == fs_)
      feature_df[feat_inds, "sign"] = feature_df[feat_inds, "sign"] + eval$sign

      # RED
      cor_mat <- cor(rbind(data_list[[test_fold]]$train_data[, features],
                           data_list[[test_fold]]$test_data[, features])) # feature-wise correlation ma
      RED_df <- rbind(RED_df,
                     data.frame(RED = mean(abs(cor_mat[upper.tri(cor_mat)])),
                                max_s = max_s_,
                                fold = test_fold,
                                fs = fs_))
    }
  }
}
```

```

}

# stability
f_mat <- matrix(0, nrow = nfeats, ncol = nfolds)
features <- subset(feature_df, (max_s == max_s_) & (fs == fs_)) %>%
  select(index, fold) %>%
  as.matrix()
f_mat[features] <- f_mat[features] + 1

stab = getStability(t(f_mat))
stability_df <- rbind(stability_df,
                      cbind(max_s = max_s_,
                            fs = fs_,
                            stab))
}
}

RED_df <- RED_df %>% group_by(max_s, fs) %>%
  summarize(value = mean(RED),
            variance = var(RED),
            lower = min(RED),
            upper = max(RED)) %>%
  as.data.frame()

summary_df <- pred_df %>% group_by(metric, model, type, max_s, fs) %>%
  summarize(mean = mean(value),
            median = median(value),
            sd = sd(value),
            min = min(value),
            max = max(value)) %>%
  as.data.frame()

save(feature_df, pred_df, RED_df, stability_df, summary_df, param_df,
      file = paste0(path, "data/exp1_predictions.Rdata"))

```

```

pred_baseline = c()
for (test_fold in 1:nfolds) {

  # all features
  features = 1:ncol(data_list[[test_fold]]$train_data)
  eval_baseline_all <- eval_all(features,
                                train_data = data_list[[test_fold]]$train_data,
                                test_data = data_list[[test_fold]]$test_data,
                                train_target = data_list[[test_fold]]$train_target,
                                test_target = data_list[[test_fold]]$test_target)

  pred_baseline <- rbind(pred_baseline,
                        cbind(eval_baseline_all$res,
                              max_s = "all",
                              fold = test_fold,
                              fs = "none"))

  features = c()
}

```

```

eval_baseline_no_features <- eval_all(features,
  train_data = data_list[[test_fold]]$train_data,
  test_data = data_list[[test_fold]]$test_data,
  train_target = data_list[[test_fold]]$train_target,
  test_target = data_list[[test_fold]]$test_target)

pred_baseline <- rbind(pred_baseline,
  cbind(eval_baseline_no_features$res,
    max_s = "none",
    fold = test_fold,
    fs = "none"))
}

```

```

## [1] "no features"
## [1] "no features"
## [1] "no features"
## [1] "no features"
## [1] "no features"

```

```

print(pred_baseline %>% subset((type=="test") & (metric == "R2") & (max_s == "none")))

```

```

##           value metric  model type max_s fold  fs
## 16 -0.000174744    R2 no_feats test  none    1 none
## 32 -0.041876692    R2 no_feats test  none    2 none
## 48 -0.054206792    R2 no_feats test  none    3 none
## 64 -0.010899006    R2 no_feats test  none    4 none
## 80 -0.001685231    R2 no_feats test  none    5 none

```

```

print(pred_baseline %>% subset((type=="test") & (metric == "RMSE") & (max_s == "all")))# %>% group_by(mod

```

```

##           value metric  model type max_s fold  fs
## 3  12.1655798   RMSE linear test  all    1 none
## 7   1.2089410   RMSE   knn test  all    1 none
## 11  1.4486256   RMSE   svm test  all    1 none
## 19  6.3092933   RMSE linear test  all    2 none
## 23  1.5932550   RMSE   knn test  all    2 none
## 27  1.6395465   RMSE   svm test  all    2 none
## 35 13.8281535   RMSE linear test  all    3 none
## 39  1.9430349   RMSE   knn test  all    3 none
## 43  1.5519600   RMSE   svm test  all    3 none
## 51  8.9323053   RMSE linear test  all    4 none
## 55  1.3047532   RMSE   knn test  all    4 none
## 59  0.8279466   RMSE   svm test  all    4 none
## 67 80.1312008   RMSE linear test  all    5 none
## 71  1.3564660   RMSE   knn test  all    5 none
## 75  1.2560683   RMSE   svm test  all    5 none

```

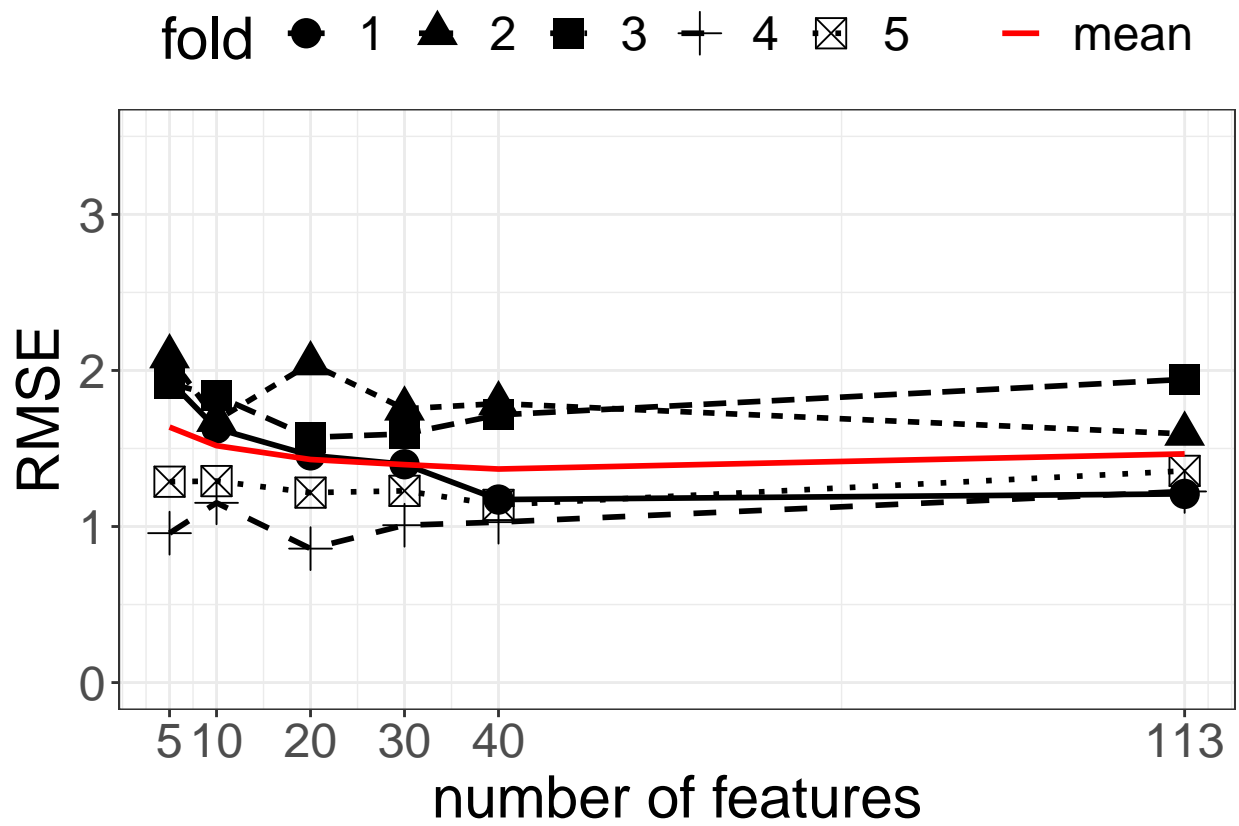
PREDICTION PLOT UBAYFS

```
load(paste0(path, "data/exp1_features.Rdata"))
load(paste0(path, "data/exp1_predictions.Rdata"))

plot_performance <- function(fs_ = "UBayFS", type_ = "test", metric_ = "RMSE", model_ = "knn"){
  p <- pred_df %>% subset(fs == fs_ & type == type_ & metric == metric_ & model == model_) %>%
    mutate(fold = as.factor(fold)) %>%
    ggplot(aes(x = max_s)) +
    geom_point(aes(y = value, pch = fold), size = 5) +
    geom_line(aes(y = value, lty = fold), linewidth = 1) +
    geom_line(data = summary_df %>% subset(fs == fs_ & type == type_ & metric == metric_ & model == model_),
              aes(x = max_s,
                  y = mean,
                  color = "mean"),
              linewidth = 1) +
    ylim(0, 3.5) +
    ylab("RMSE") +
    xlab("number of features") +
    scale_x_continuous(breaks = max_s_opts) +
    theme_bw() +
    theme(text = element_text(size = 22), legend.position = "top") +
    guides(fill="none", color = guide_legend(title = "")) +
    scale_fill_manual(values = "red") +
    scale_color_manual(values = "red")

  return(p)
}

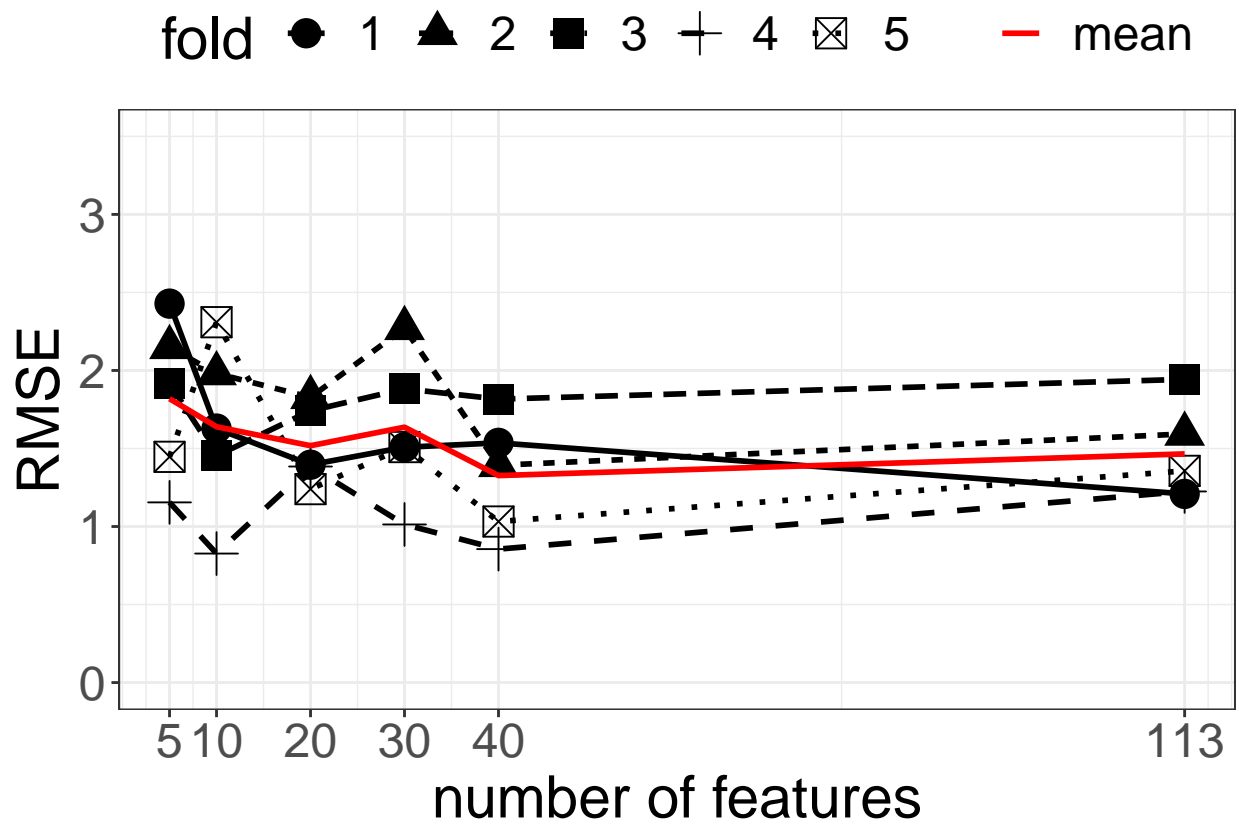
p <- plot_performance(fs_ = "UBayFS", type_ = "test", metric_ = "RMSE", model_ = "knn")
p
```



```
ggsave(p, filename = paste0(path, "plots/experiment_1_performance_max_s_kNN_UBayFS.png"), width = 8, height = 6)
```

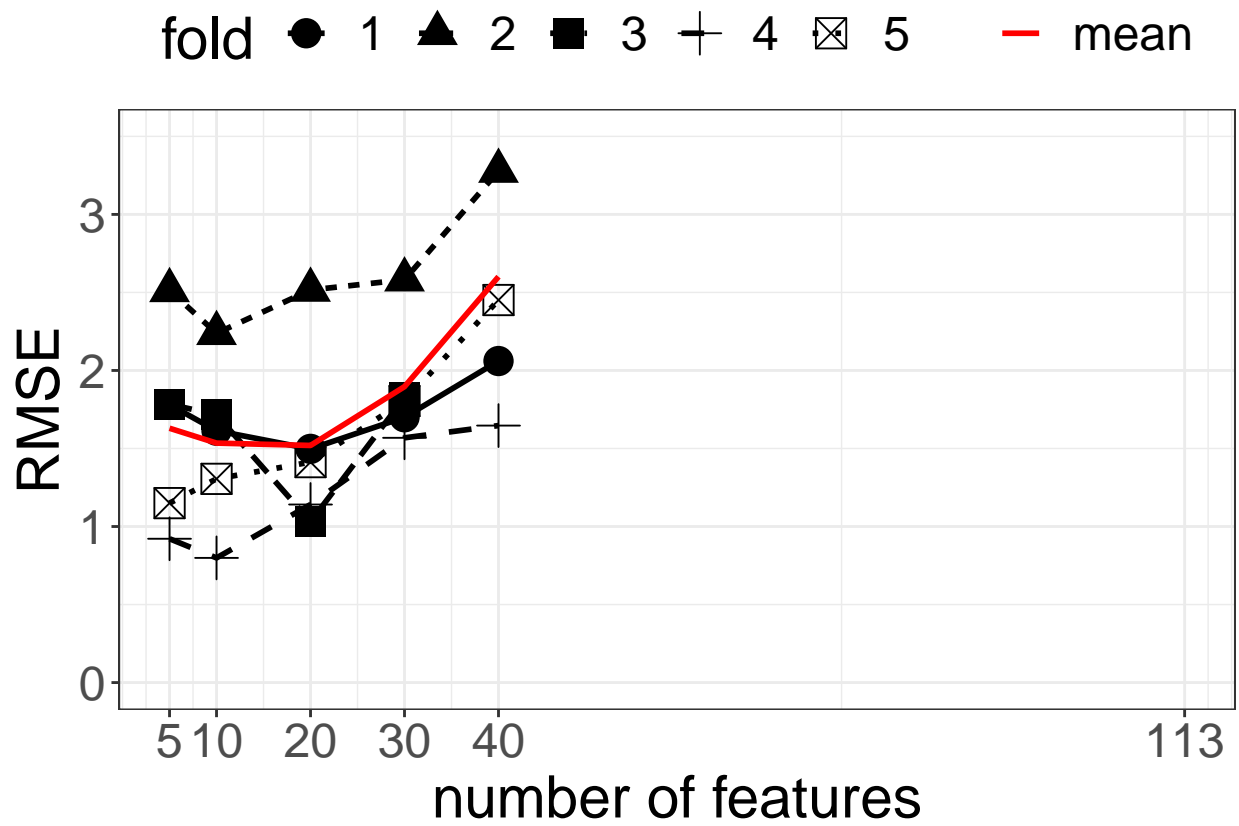
```
p <- plot_performance(fs_ = "RENT", type_ = "test", metric_ = "RMSE", model_ = "knn")
```

```
p
```

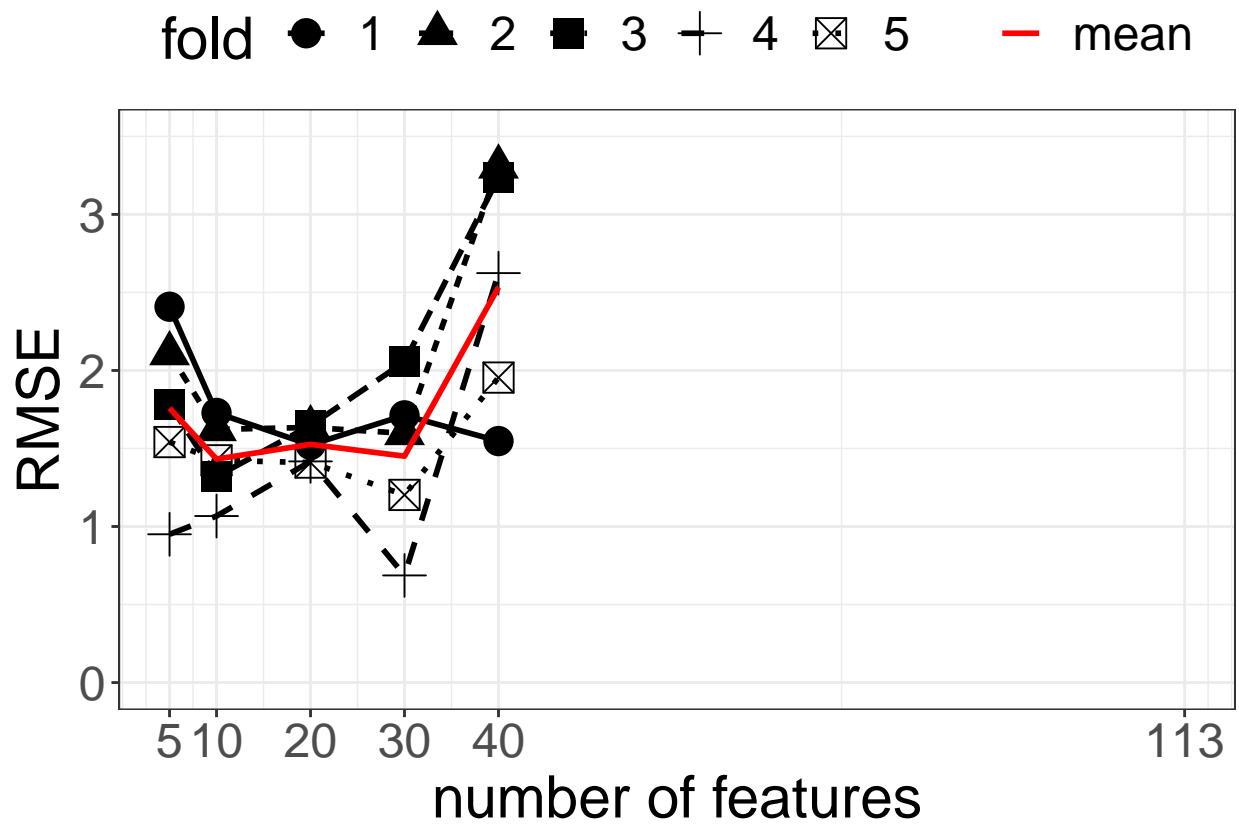


```
ggsave(p, filename = paste0(path, "plots/experiment_1_performance_max_s_kNN_RENT.png"), width = 8, height = 6)
```

```
p <- plot_performance(fs_ = "UBayFS", type_ = "test", metric_ = "RMSE", model_ = "linear")
p
```

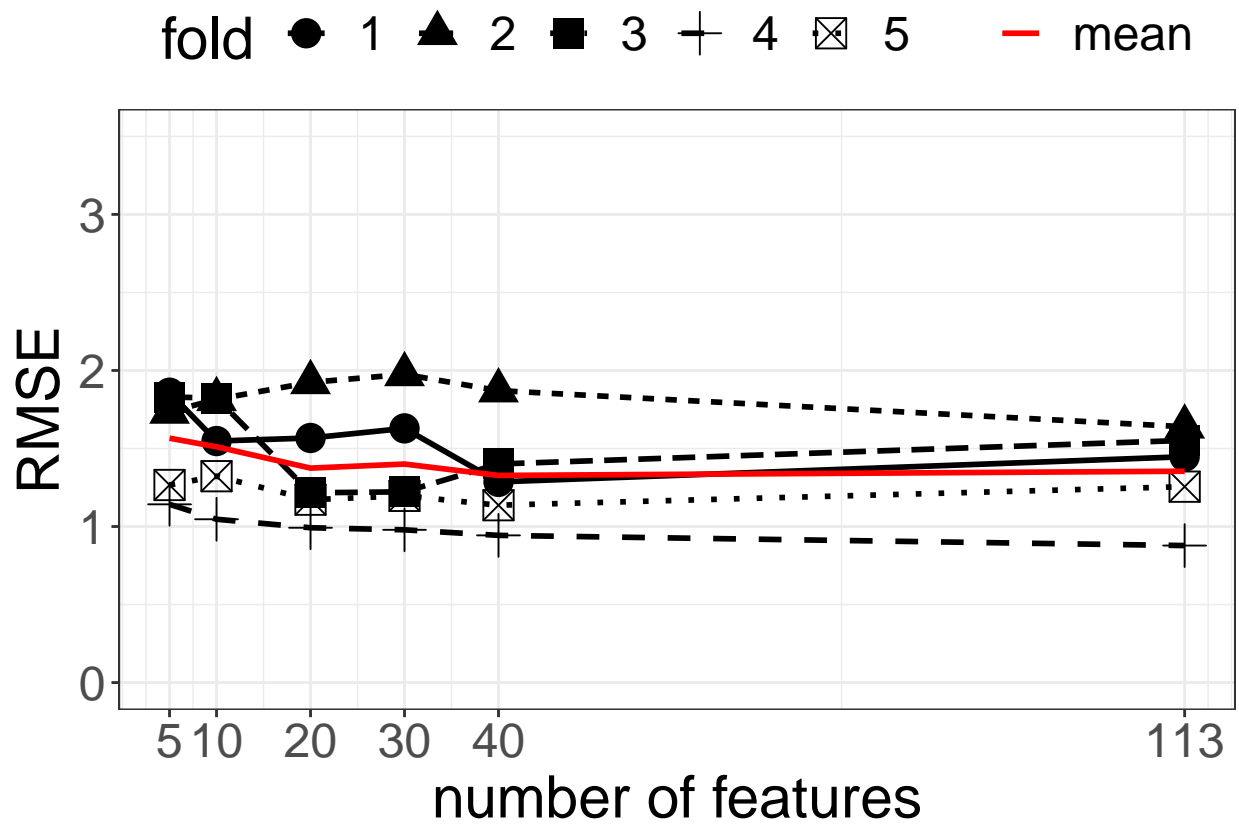


```
ggsave(p, filename = paste0(path, "plots/experiment_1_performance_max_s_linear_UBayFS.png"), width = 8,
p <- plot_performance(fs_ = "RENT", type_ = "test", metric_ = "RMSE", model_ = "linear")
p
```

```
ggsave(p, filename = paste0(path, "plots/experiment_1_performance_max_s_linear_RENT.png"), width = 8, height = 6)
```

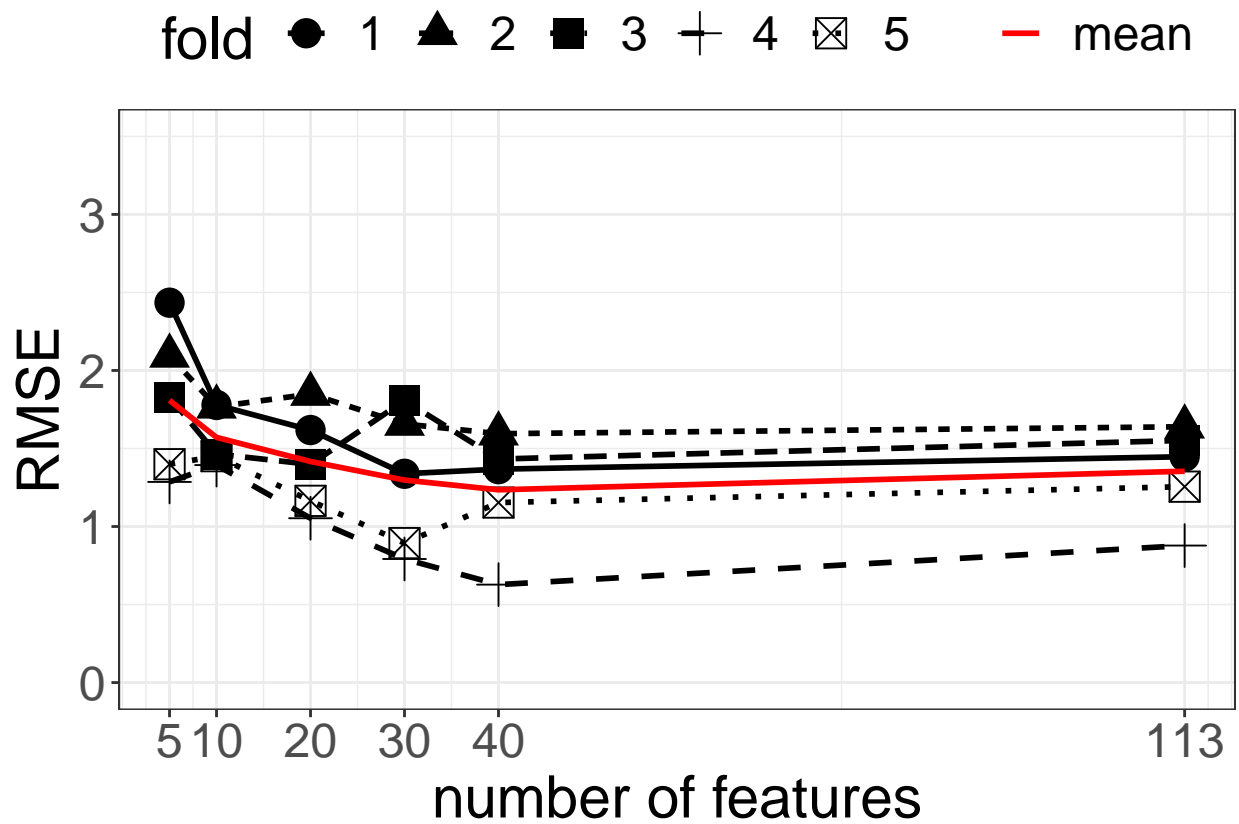
```
p <- plot_performance(fs_ = "UBayFS", type_ = "test", metric_ = "RMSE", model_ = "svm")
p
```



```
ggsave(p, filename = paste0(path, "plots/experiment_1_performance_max_s_svm_UBayFS.png"), width = 8, height = 6)
```

```
p <- plot_performance(fs_ = "RENT", type_ = "test", metric_ = "RMSE", model_ = "svm")
```

```
p
```



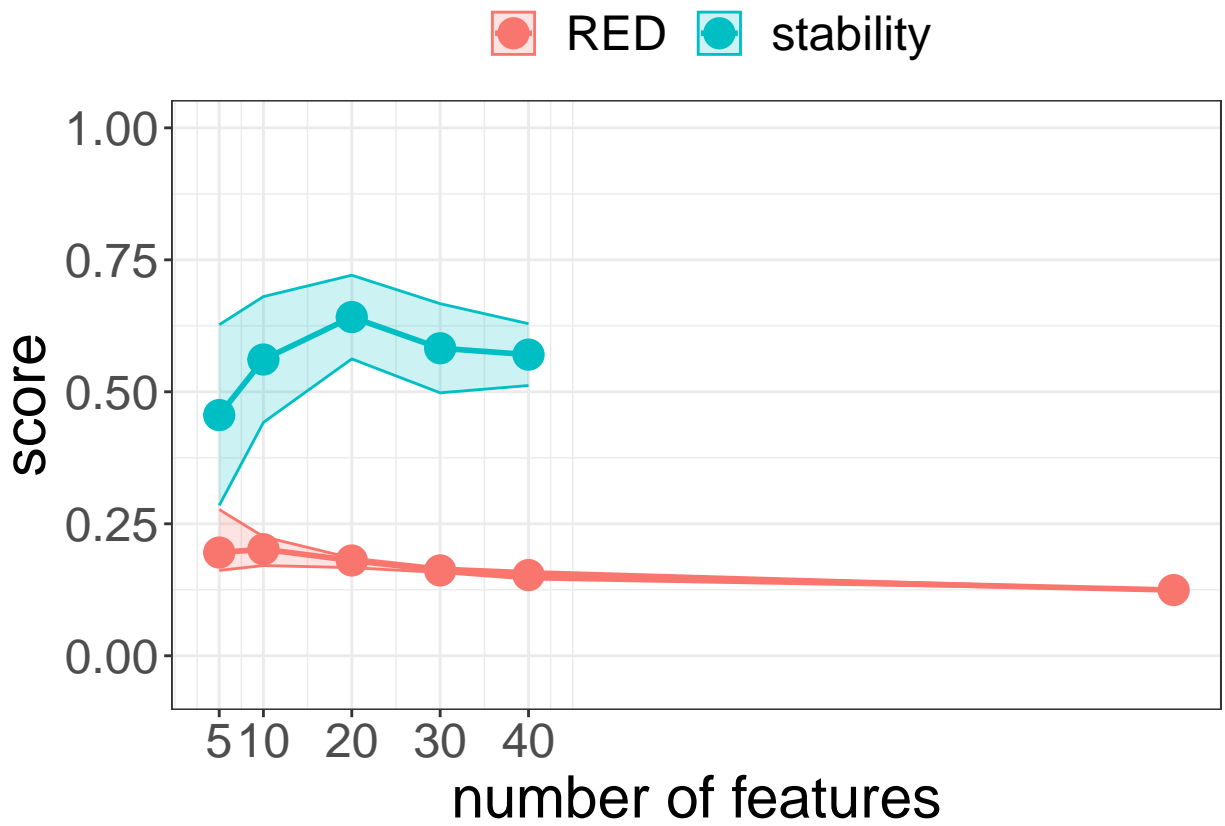
```
ggsave(p, filename = paste0(path, "plots/experiment_1_performance_max_s_svm_RENT.png"), width = 8, height = 6)
```

STABILITY PLOT UBAYFS

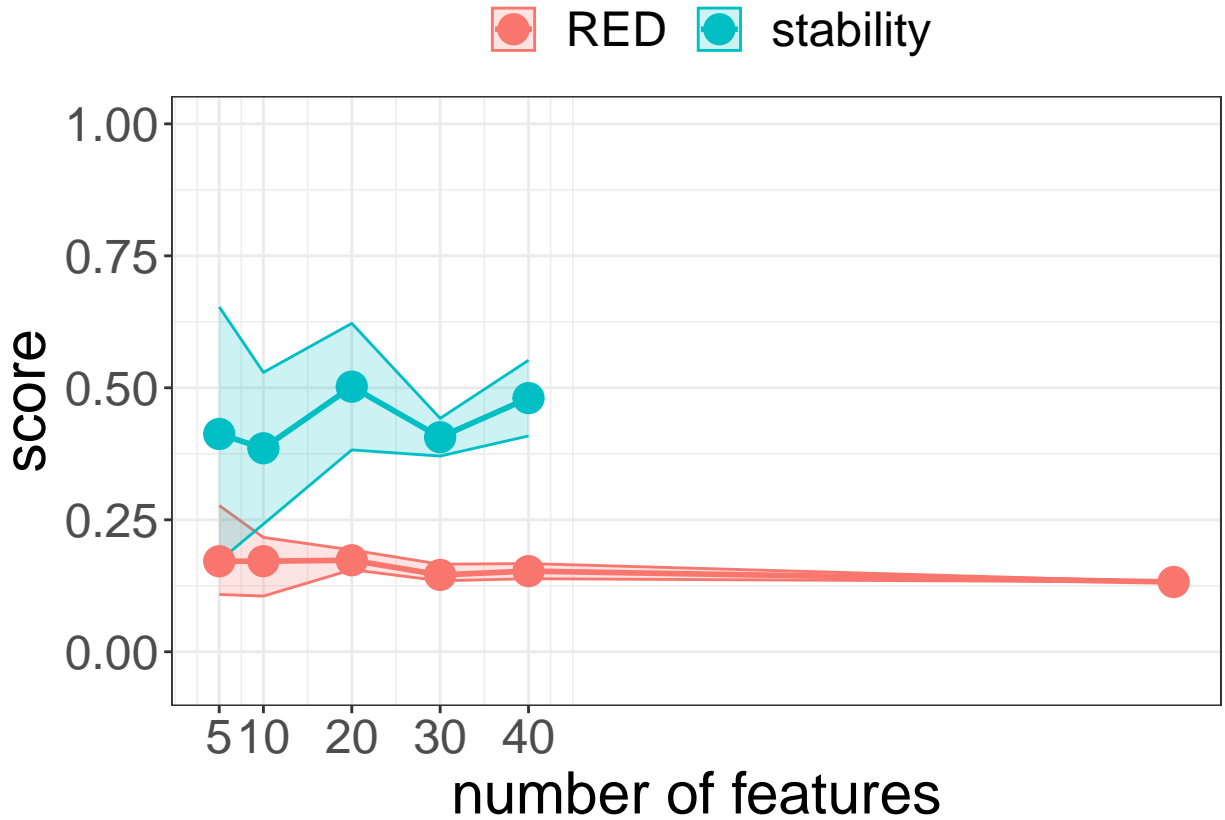
```
plot_stability <- function(fs_ = "UBayFS"){
  p <- rbind(cbind(stability_df, score = "stability"),
            cbind(RED_df, score = "RED")) %>% subset(fs == fs_) %>%
  ggplot(aes(x= max_s, y = value, color = score, fill = score)) +
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.2)+
  geom_point(size = 5) +
  geom_line(linewidth = 1) +
  ylim(-0.05,1)+
  ylab("score") +
  xlab("number of features") +
  scale_x_continuous(breaks=c(5,10,20,30,40)) +
  theme_bw() +
  theme(text = element_text(size = 22), legend.position = "top", legend.title = element_blank())

  return(p)
}

p <- plot_stability(fs_ = "UBayFS")
p
```



```
ggsave(p, filename = paste0(path, "plots/experiment_1_stability_red_UBayFS.png"), width = 8, height = 5)
p <- plot_stability(fs_ = "RENT")
p
```



```
ggsave(p, filename = paste0(path, "plots/experiment_1_stability_red_RENT.png"), width = 8, height = 5)
```

IN-DEPTH ANALYSIS FOLD 2 & 4

```
max_s_ = 20 # fix max_s

folds = c(2,4)
title = c("worst (fold 2)", "best (fold 4)")

fold_res <- data.frame()

for(i in 1:length(folds)){
  fold_ = folds[i]

  k_U = subset(param_df, (fs == "UBayFS") & (max_s == max_s_) & (fold == fold_) & (method == "knn"))$para
  k_R = subset(param_df, (fs == "RENT") & (max_s == max_s_) & (fold == fold_) & (method == "knn"))$para

  fold_res_U <- fold_analysis(fold_,
                             subset(feature_df, (max_s == max_s_) & (fs == "UBayFS") & (fold == fold_))$
                             k = k_U)
  fold_res_R <- fold_analysis(fold_,
                             subset(feature_df, (max_s == max_s_) & (fs == "RENT") & (fold == fold_))$in
                             k = k_R)
```

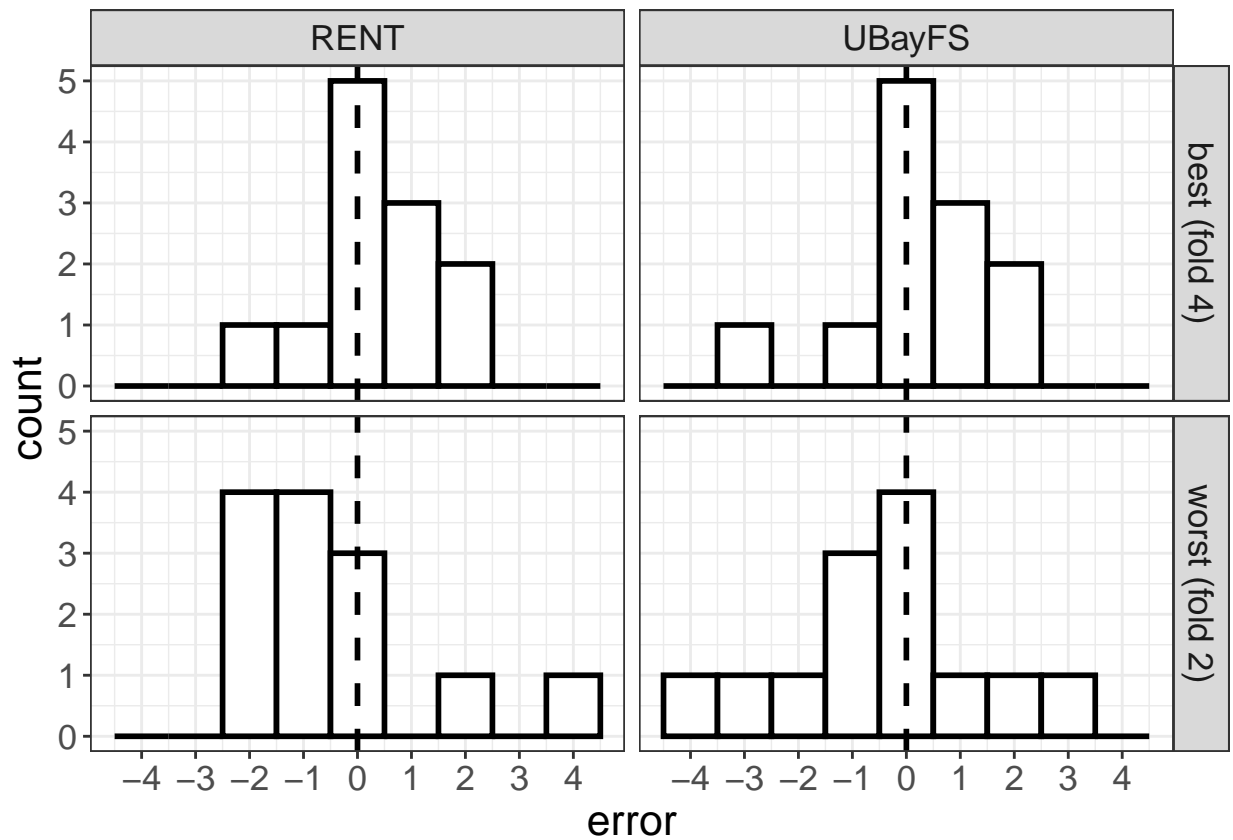
```

fold_res <- rbind(fold_res,
  cbind(fold_res_U, fs = "UBayFS", fold = title[i]),
  cbind(fold_res_R, fs = "RENT", fold = title[i])
)
}

fold_res = fold_res %>% mutate(error = true - pred)

p <- fold_res %>%
  ggplot(aes(x = error)) +
    geom_histogram(col = "black",
      fill = "white",
      binwidth = 1,
      linewidth = 1) +
    scale_x_continuous(breaks = seq(-6, 6, 1)) +
    geom_vline(aes(xintercept = 0),
      lty = "dashed",
      color = "black",
      linewidth = 1) +
    facet_grid(fold~fs) +
    theme_bw() +
    theme(text = element_text(size = 16))
p

```



```
ggsave(p, filename = paste0(path, "plots/experiment_1_histograms.png"), width = 8, height = 5)
```

OUTLIER PATIENTS

```
outliers <- fold_res %>%  
  subset(fold == "worst (fold 2)" & abs(error) > 2.5)  
outliers
```

```
##      id true    pred    fs      fold    error  
## 1  8139    6 2.666667 UBayFS worst (fold 2) 3.333333  
## 8  9086    1 5.333333 UBayFS worst (fold 2) -4.333333  
## 9  9090    1 4.333333 UBayFS worst (fold 2) -3.333333  
## 14 8139    6 2.000000  RENT worst (fold 2) 4.000000
```

FEATURE SET DIFFERENCES ON WORST FOLD

```
features_UBayFS <- feature_df %>% subset(max_s == 20 & fold == 2 & fs == "UBayFS") %>%  
  select(index, fs)  
features_RENT <- feature_df %>% subset(max_s == 20 & fold == 2 & fs == "RENT") %>%  
  select(index, fs)  
  
full_join(features_UBayFS, features_RENT, by = "index")
```

```
##      index  fs.x fs.y  
## 1        2 UBayFS RENT  
## 2        3 UBayFS RENT  
## 3        6 UBayFS RENT  
## 4        7 UBayFS <NA>  
## 5       15 UBayFS RENT  
## 6       20 UBayFS RENT  
## 7       22 UBayFS RENT  
## 8       23 UBayFS RENT  
## 9       28 UBayFS RENT  
## 10      40 UBayFS RENT  
## 11      43 UBayFS RENT  
## 12      46 UBayFS <NA>  
## 13      54 UBayFS RENT  
## 14      59 UBayFS <NA>  
## 15      66 UBayFS <NA>  
## 16      71 UBayFS RENT  
## 17      94 UBayFS RENT  
## 18     101 UBayFS <NA>  
## 19     108 UBayFS RENT  
## 20     113 UBayFS RENT  
## 21        8 <NA> RENT  
## 22       17 <NA> RENT  
## 23       27 <NA> RENT
```

```
## 24    53    <NA> RENT
## 25    67    <NA> RENT
## 26   112    <NA> RENT
```

FEATURE SET DIFFERENCES ON BEST/WORST FOLD

```
features_UBayFS_2 <- feature_df %>% subset(max_s == 20 & fold == 2 & fs == "UBayFS") %>%
  select(index, fs)
features_UBayFS_4 <- feature_df %>% subset(max_s == 20 & fold == 4 & fs == "UBayFS") %>%
  select(index, fs)

features_RENT_2 <- feature_df %>% subset(max_s == 20 & fold == 2 & fs == "RENT") %>%
  select(index, fs)
features_RENT_4 <- feature_df %>% subset(max_s == 20 & fold == 4 & fs == "RENT") %>%
  select(index, fs)

full_join(
  full_join(features_UBayFS_2, features_RENT_2, by = "index", suffix = c("UBayFS", "RENT")),
  full_join(features_UBayFS_4, features_RENT_4, by = "index", suffix = c("UBayFS", "RENT")),
  by = "index", suffix = c("fold 2", "fold 4"))
```

##	index	fsUBayFSfold 2	fsRENTfold 2	fsUBayFSfold 4	fsRENTfold 4
## 1	2	UBayFS	RENT	UBayFS	<NA>
## 2	3	UBayFS	RENT	UBayFS	RENT
## 3	6	UBayFS	RENT	<NA>	<NA>
## 4	7	UBayFS	<NA>	UBayFS	RENT
## 5	15	UBayFS	RENT	UBayFS	<NA>
## 6	20	UBayFS	RENT	UBayFS	<NA>
## 7	22	UBayFS	RENT	UBayFS	RENT
## 8	23	UBayFS	RENT	<NA>	RENT
## 9	28	UBayFS	RENT	UBayFS	RENT
## 10	40	UBayFS	RENT	<NA>	RENT
## 11	43	UBayFS	RENT	<NA>	<NA>
## 12	46	UBayFS	<NA>	<NA>	<NA>
## 13	54	UBayFS	RENT	UBayFS	RENT
## 14	59	UBayFS	<NA>	<NA>	<NA>
## 15	66	UBayFS	<NA>	UBayFS	RENT
## 16	71	UBayFS	RENT	UBayFS	RENT
## 17	94	UBayFS	RENT	UBayFS	RENT
## 18	101	UBayFS	<NA>	UBayFS	RENT
## 19	108	UBayFS	RENT	UBayFS	RENT
## 20	113	UBayFS	RENT	<NA>	<NA>
## 21	8	<NA>	RENT	<NA>	<NA>
## 22	17	<NA>	RENT	<NA>	RENT
## 23	27	<NA>	RENT	<NA>	<NA>
## 24	53	<NA>	RENT	UBayFS	RENT
## 25	67	<NA>	RENT	<NA>	<NA>
## 26	112	<NA>	RENT	UBayFS	RENT
## 27	16	<NA>	<NA>	UBayFS	RENT
## 28	24	<NA>	<NA>	UBayFS	RENT
## 29	58	<NA>	<NA>	UBayFS	RENT

## 30	85	<NA>	<NA>	UBayFS	<NA>
## 31	90	<NA>	<NA>	UBayFS	RENT
## 32	18	<NA>	<NA>	<NA>	RENT
## 33	65	<NA>	<NA>	<NA>	RENT
## 34	82	<NA>	<NA>	<NA>	RENT
## 35	98	<NA>	<NA>	<NA>	RENT

Check how often the prior features were selected.

```
feature_df %>% subset(index %in% prior_features_all_ind & max_s == max_s_) %>%
  group_by(fold, fs) %>%
  summarize(num_abs = n()) %>%
  group_by(fs) %>%
  summarize(avg_num_abs = mean(num_abs),
            avg_num_rel = mean(num_abs / max_s_))
```

```
## # A tibble: 2 x 3
##   fs      avg_num_abs avg_num_rel
##   <chr>      <dbl>      <dbl>
## 1 RENT         5.8         0.29
## 2 UBayFS       5.8         0.29
```

FEATURE TABLE

```
f_mat <- matrix(0, ncol = 2, nrow = nfeats)
sign_mat <- matrix(0, ncol = 2, nrow = nfeats)
colnames(f_mat) <- colnames(sign_mat) <- c("RENT", "UBayFS")

for(fold_ in 1:nfolds){
  for(fs_ in c("RENT", "UBayFS")){
    features <- feature_df %>% subset(max_s == 20 & fs == fs_ & fold == fold_)
    f_mat[features$index, fs_] <- f_mat[features$index, fs_] + 1
    sign_mat[features$index, fs_] <- sign_mat[features$index, fs_] + features$sign
  }
}

feature_counts <- data.frame(block = "",
  prior = ifelse(1:nfeats %in% prior_features_all_ind, "*", ""),
  name = feat_names,
  RENT = paste0(f_mat[, "RENT"],
    ifelse(sign_mat[, "RENT"] != 0, paste0("(",
      ifelse(sign_mat[, "RENT"] > 0,
        ifelse(sign_mat[, "RENT"] == f_mat[, "RENT"], "++", "+"),
        ifelse(sign_mat[, "RENT"] == -f_mat[, "RENT"], "--", "-"),
        "")), "")),
  UBayFS = paste0(f_mat[, "UBayFS"],
    ifelse(sign_mat[, "UBayFS"] != 0, paste0("(",
      ifelse(sign_mat[, "UBayFS"] > 0,
```

```

        ifelse(sign_mat[, "UBayFS"] == f_mat[, "UBayFS"], "+", "
        ifelse(sign_mat[, "UBayFS"] == -f_mat[, "UBayFS"], "--",
        ")", ""))
    )

save(feature_counts, file = paste0(path, "data/feature_counts_exp1.Rdata"))
print(xtable(feature_counts, display = c("s", "s", "s", "s", "s", "s")), include.rownames = FALSE)

```

```

## % latex table generated in R 4.2.2 by xtable 1.8-4 package
## % Mon Aug 21 09:47:43 2023
## \begin{table}[ht]
## \centering
## \begin{tabular}{l11111}
## \hline
## block & prior & name & RENT & UBayFS \\
## \hline
## & & Abs. Neutrophil Count & 0 & 0 \\
## & * & Albumin & 2 & 4(--) \\
## & & CRP & 5(-) & 5(--) \\
## & & Creatinine & 0 & 0 \\
## & & Haemoglobin & 0 & 0 \\
## & & WBC & 1(--) & 1(--) \\
## & & ALP > $ Normal < $= 3UNL & 4(--) & 5(--) \\
## & & ALP > $ 3UNL & 1(++) & 0 \\
## & & Chromogranin\_A > $ Normal < $= 2UNL & 0 & 0 \\
## & & Chromogranin\_A > $ 2UNL & 0 & 0 \\
## & * & LDH > $ Normal < $= 2UNL & 0 & 0 \\
## & * & LDH > $ 2UNL & 0 & 0 \\
## & & NSE > $ Normal < $= 2UNL & 0 & 0 \\
## & & NSE > $ 2UNL & 0 & 0 \\
## & * & Platelets & 2(--) & 5(--) \\
## & & Injection to Scan [min] & 2 & 2(++) \\
## & & Weight [kg] & 2(--) & 0 \\
## & * & Total MTV [cm^3] & 3(+) & 1(--) \\
## & & SUVmean & 0 & 0 \\
## & * & SUVmax & 2 & 4 \\
## & & SUVmean (total) & 1(++) & 0 \\
## & & SUVmax (total) & 5(--) & 5(--) \\
## & * & Total TLG [g] & 4 & 1(++) \\
## & & Institution Rikshospitalet & 4(++) & 4(++) \\
## & & Institution Ullevaall & 0 & 0 \\
## & & Height [cm] & 0 & 0 \\
## & & Glucose [mmol/L] & 2(--) & 0 \\
## & * & Ki-67 & 5(--) & 5(--) \\
## & & Hist Exam Metastasis & 0 & 0 \\
## & * & Primary Tumour Esophagus & 0 & 0 \\
## & * & Primary Tumour Gallbladder/duct & 0 & 0 \\
## & * & Primary Tumour Gastric & 0 & 0 \\
## & * & Primary Tumour Other abdominal & 0 & 0 \\
## & * & Primary Tumour Pancreas & 1(++) & 0 \\
## & * & Primary Tumour Rectum & 0 & 0 \\
## & * & Unknown Pr. With Dominance of GI met. & 0 & 0 \\
## & & Co-existing Neoplasm Adenoma & 0 & 0

```

```

## & & Co-existing Neoplasm Dysplasia & 0 & 0 \\
## & & No Co-existing Neoplasm & 0 & 0 \\
## & * & Tumour Morphology WD & 4(+) & 3(--) \\
## & & Chromogranin A Staining & 0 & 0 \\
## & & Architecture Infiltrative & 1(++) & 0 \\
## & & Architecture Organoid & 1(++) & 1(++) \\
## & & Architecture Solid & 0 & 0 \\
## & & Architecture Trabecular & 1(--)& 1(++) \\
## & & Vessel Pattern Distant & 1(++) & 2(--) \\
## & & Biopsy Location Gastric & 0 & 0 \\
## & & Biopsy Location Liver Metastasis & 0 & 0 \\
## & & Biopsy Location Lymph Node & 0 & 0 \\
## & & Biopsy Location Oesophagus & 0 & 0 \\
## & & Biopsy Location Pancreas & 0 & 0 \\
## & & Biopsy Location Peritoneum & 2 & 1(--) \\
## & & No Stroma & 4(++) & 1(++) \\
## & & Stroma & 3(++) & 3(-) \\
## & & Geographic Necrosis & 0 & 1(--) \\
## & & Synaptophysin Staining 2+ & 0 & 0 \\
## & & Synaptophysin Staining 3+ & 0 & 0 \\
## & * & Age at Diagnosis & 2 & 1(--) \\
## & & Time from PET to Metastasis (days) & 0 & 1(++) \\
## & & Time from PET to Diagnosis (days) & 0 & 0 \\
## & & Time from diag to mets (months) & 0 & 0 \\
## & & Sex & 0 & 0 \\
## & & Loc. Adv. Resectable Disease & 0 & 0 \\
## & & Loc. Recurrence & 0 & 0 \\
## & & Metastatic Disease at Time of Diagnosis & 3(+) & 1(++) \\
## & & Treatment Intention Palliative & 4(-) & 5(--) \\
## & & Prior Other Cancer & 2(++) & 1(--) \\
## & & Living Alone & 0 & 0 \\
## & * & TNM staging Pathological & 0 & 0 \\
## & & Stage grouped Stage IV & 0 & 0 \\
## & & Mets Bone & 5(--)& 5(--) \\
## & & Mets LN Distant & 0 & 0 \\
## & & Mets LN Regional & 0 & 0 \\
## & & Mets LN Retro & 0 & 0 \\
## & & Mets LN & 0 & 0 \\
## & & Mets Liver & 0 & 0 \\
## & & Mets Lung & 0 & 0 \\
## & & Mets Other & 0 & 0 \\
## & & Mets Skin & 0 & 0 \\
## & & Primary Tumour Resected & 0 & 0 \\
## & & M-stage M1 & 0 & 0 \\
## & & BMI & 1(--)& 0 \\
## & & Non Smoker & 0 & 0 \\
## & & Smoker & 0 & 0 \\
## & & Radical Surgery & 3(++) & 4 \\
## & & Co-morbidity Severity 1 & 0 & 0 \\
## & & Co-morbidity Severity $>$ 1 & 0 & 0 \\
## & & T-stage T2 & 0 & 0 \\
## & & T-stage T3 & 0 & 0 \\
## & & T-stage T4 & 2(--)& 2(--) \\
## & & N-stage N1 & 0 & 0

```

```

## & & N-stage $>$ N1 & 0 & 0 \\
## & * & WHO Perf Stat 1 & 0 & 0 \\
## & * & WHO Perf Stat 2 & 4(--) & 5(--) \\
## & * & WHO Perf Stat 3 & 0 & 0 \\
## & * & WHO Perf Stat 4 & 0 & 0 \\
## & & Time from PET to first treatment (days) & 0 & 0 \\
## & & Chemotherapy Type Cisplatin/Etoposide & 4(+) & 3(+) \\
## & & Chemotherapy Type Other & 0 & 0 \\
## & & Chemotherapy Type Temozolomide/Capecitabine & 1(++) & 0 \\
## & & Chemotherapy Type Temozolomide/Everolimus & 4(++) & 5(++) \\
## & & Best Response (RECIST) Not Assessed & 0 & 0 \\
## & & Best Response (RECIST) Only Clinical PD & 0 & 0 \\
## & & Best Response (RECIST) Partial Response & 2(--) & 0 \\
## & & Best Response (RECIST) Progressive Disease & 0 & 0 \\
## & & Best Response (RECIST) Stable Disease & 0 & 0 \\
## & & Reintroduction with Cisplatin Etoposide & 0 & 0 \\
## & & Number of Courses & 4(++) & 4(++) \\
## & & Treatment Stopped Other & 1(++) & 1(++) \\
## & & Treatment Stopped Progression of Disease & 0 & 0 \\
## & & Treatment Stopped Toxicity & 0 & 0 \\
## & & No Progression & 5(++) & 4(++) \\
## & & Progression & 3 & 3(+) \\
## \hline
## \end{tabular}
## \end{table}

```