XGBModel_YL

Yujing Lu

2022-04-21

```
library(dplyr)
library(MachineShop)
library(recipes)
library(kableExtra)
library(arsenal)
```

```
## Loading required package: foreach
```

```
## Loading required package: iterators
```

```
## Loading required package: snow
```

Define XGBoost model and its tuning grid to be tuned simultaneously with recipe input later

```
xgbtree_model <- TunedModel(XGBTreeModel,
                            grid = expand_params(
                              nrounds = as.integer(c(50, 100, 150)),
                              # number of boosting iterations
                              eta = seq(0.1, 0.5, length = 5),
                              # shrinkage of variable weights at each iteration to prevent overfitting
                              max_depth = as.integer(c(4:8))
                              # maximum tree depth
                              )
                            )
```

knn, nzv, dummy, center, scale

```
fnames <- c("./knn_none_xgboost_fit.RDS", "./knn_none_xgboost_res.RDS")

# remove predictor variables that have too many missing values by using step_nzv
# use knn to impute
# use bag to impute
rec_knn_none <- rec_base %>%
  step_nzv(all_predictors()) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_center(all_predictors()) %>%
  step_scale(all_predictors()) %>%
  step_impute_knn(all_predictors(), id = "knn")
  # step_impute_mode(all_nominal_predictors()) %>%
  # step_impute_mean(all_numeric_predictors())

rec_grid_knn_none <- expand_steps(
  knn = list(neighbors = 1:5)
)

rec_tun_knn_none <- TunedInput(rec_knn_none, grid = rec_grid_knn_none)

mspec_tun_knn_none <- ModelSpecification(
  rec_tun_knn_none,
  model = xgbtree_model,
  control = ctrl
) %>% set_optim_bayes()
# use bayesian optimization

# get the optimal model selected
mlfit_knn_none <- fit(mspec_tun_knn_none)
```

```
## ModelSpecification(15)
```

```
## Warning: There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
```

```
saveRDS(mlfit_knn_none, fnames[1])

# get resampled predictive performance
mlres_knn_none <- resample(mspec_tun_knn_none, control = ctrl)
saveRDS(mlres_knn_none, fnames[2])
summary(mlres_knn_none)
```

```
##              Statistic
## Metric             Mean    Median         SD         Min       Max NA
##   Brier       0.2232724 0.2182319 0.07953322  0.09827789 0.3614104  0
##   Accuracy    0.6917892 0.6672794 0.10808224  0.55555556 0.8750000  0
##   Kappa       0.3406824 0.3244782 0.23905518 -0.03703704 0.7333333  0
##   ROC AUC     0.7577056 0.7750000 0.14639842  0.51948052 0.9480519  0
##   Sensitivity 0.7636364 0.8000000 0.13886593  0.50000000 1.0000000  0
##   Specificity 0.5785714 0.5833333 0.22449237  0.16666667 0.8333333  0
```

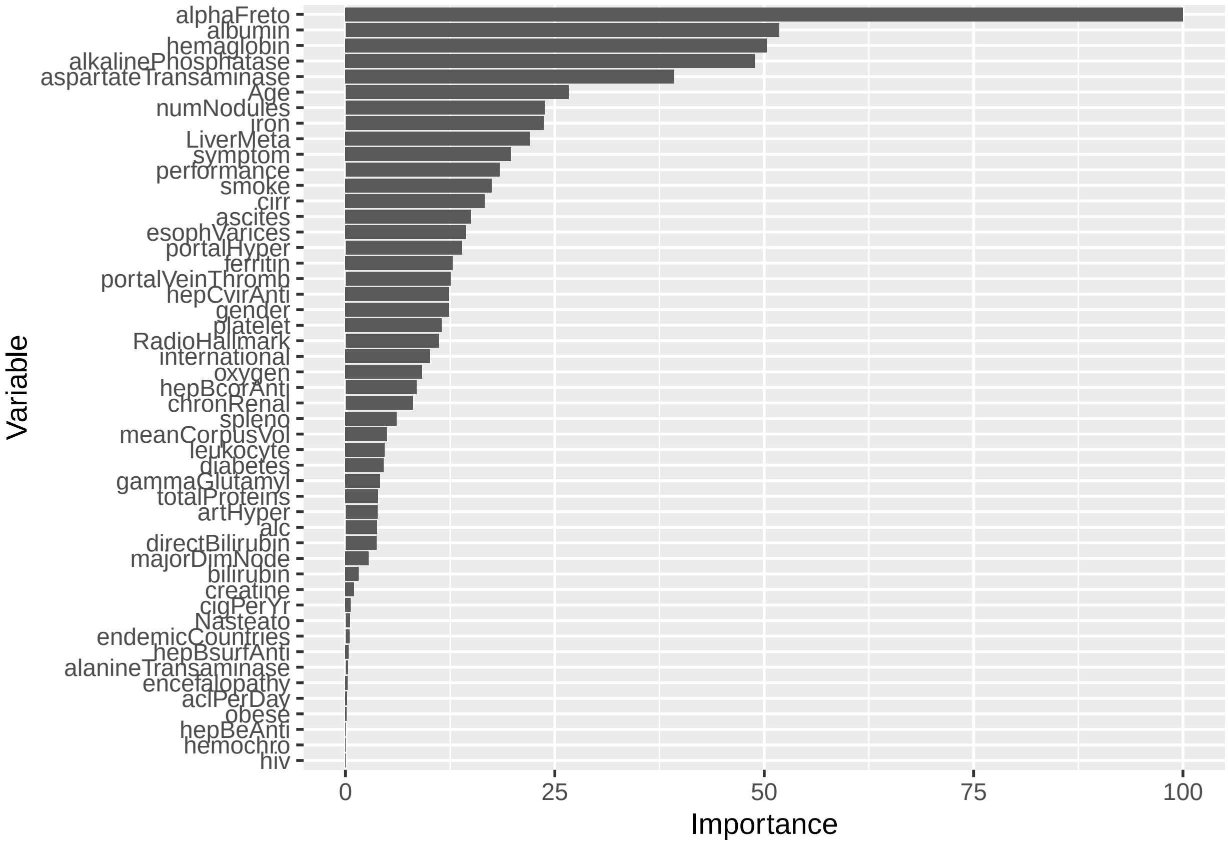
```
(tuned_model_knn_none <- as.MLModel(mlfit_knn_none))
```

```
## --- MLModel object -------------------------------------------------------------
##
## Model name: XGBTreeModel
## Label: Trained Extreme Gradient Boosting (Tree)
## Package: xgboost (>= 1.3.0)
## Response types: factor, numeric, PoissonVariate, Surv
## Case weights support: TRUE
## Tuning grid: TRUE
## Variable importance: TRUE
##
## Parameters:
## List of 27
##  $ nrounds                    : num 50
##  $ eta                        : num 0.1
##  $ gamma                      : num 0
##  $ max_depth                  : num 5
##  $ min_child_weight           : num 1
##  $ max_delta_step             : language 0.7 * is(y, "PoissonVariate")
##  $ subsample                  : num 1
##  $ colsample_bytree           : num 1
##  $ colsample_bylevel          : num 1
##  $ colsample_bynode           : num 1
##   [list output truncated]
##
## === $TrainingStep1 =============================================================
## === TrainingStep object ===
##
## Optimization method: Bayesian Optimization
## ModelSpecification log:
## # A tibble: 15 x 5
##    name         epoch selected params$input.tbW1~ $model.1qSf$nro~ metrics$Brier
##    <chr>        <int> <lgl>                 <dbl>            <dbl>         <dbl>
##  1 ModelSpec.1      0 FALSE                     3               57         0.229
##  2 ModelSpec.2      0 FALSE                     5              119         0.244
##  3 ModelSpec.3      0 FALSE                     1               74         0.215
##  4 ModelSpec.4      0 FALSE                     4              132         0.235
##  5 ModelSpec.5      0 FALSE                     2               95         0.240
##  6 ModelSpec.6      1 FALSE                     5              150         0.233
##  7 ModelSpec.7      2 FALSE                     1               55         0.218
##  8 ModelSpec.8      3 FALSE                     1              101         0.215
##  9 ModelSpec.9      4 FALSE                     1               76         0.213
## 10 ModelSpec.10     5 TRUE                      1               50         0.207
## # ... with 5 more rows, and 7 more variables: params$model.1qSf$eta <dbl>,
## #   $$max_depth <dbl>, metrics$Accuracy <dbl>, $Kappa <dbl>, $`ROC AUC` <dbl>,
## #   $Sensitivity <dbl>, $Specificity <dbl>
##
## Selected row: 10
## Metric: Brier = 0.2070273
```

```
summary(tuned_model_knn_none)
```

```
## --- $TrainingStep1 -------------------------------------------------------------
## # A tibble: 15 x 5
##    name         epoch selected params$input.tbW1~ $model.1qSf$nro~ metrics$Brier
##    <chr>        <int> <lgl>                 <dbl>            <dbl>         <dbl>
##  1 ModelSpec.1      0 FALSE                     3               57         0.229
##  2 ModelSpec.2      0 FALSE                     5              119         0.244
##  3 ModelSpec.3      0 FALSE                     1               74         0.215
##  4 ModelSpec.4      0 FALSE                     4              132         0.235
##  5 ModelSpec.5      0 FALSE                     2               95         0.240
##  6 ModelSpec.6      1 FALSE                     5              150         0.233
##  7 ModelSpec.7      2 FALSE                     1               55         0.218
##  8 ModelSpec.8      3 FALSE                     1              101         0.215
##  9 ModelSpec.9      4 FALSE                     1               76         0.213
## 10 ModelSpec.10     5 TRUE                      1               50         0.207
## # ... with 5 more rows, and 7 more variables: params$model.1qSf$eta <dbl>,
## #   $$max_depth <dbl>, metrics$Accuracy <dbl>, $Kappa <dbl>, $`ROC AUC` <dbl>,
## #   $Sensitivity <dbl>, $Specificity <dbl>
```

```
# variable importance
varimp(mlfit_knn_none) %>% plot()
```



none-1.pdf

knn, corr, nzv, dummy, center, scale

```
fnames <- c("./knn_corr_xgboost_fit.RDS", "./knn_corr_xgboost_res.RDS")
```

```r
# remove predictor variables that have too many missing values by using step_nzv
# use knn to impute
rec_knn_corr <- rec_base %>%
  step_nzv(all_predictors()) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_center(all_predictors()) %>%
  step_scale(all_predictors()) %>%

  step_impute_knn(all_predictors(), id = "knn") %>%
  # step_impute_mode(all_nominal_predictors()) %>%
  # step_impute_mean(all_numeric_predictors())
  step_corr(all_numeric_predictors(), id = "corr")

rec_grid_knn_corr <- expand_steps(
  knn = list(neighbors = 1:5),
  corr = list(threshold = c(0.75, 0.8, 0.85, 0.9))
)

rec_tun_knn_corr <- TunedInput(rec_knn_corr, grid = rec_grid_knn_corr)

mspec_tun_knn_corr <- ModelSpecification(
  rec_tun_knn_corr,
  model = xgbtree_model,
  control = ctrl
) %>% set_optim_bayes()
# use bayesian optimization

# get the optimal model selected
mlfit_knn_corr <- fit(mspec_tun_knn_corr)
```

```
## ModelSpecification(16)
```

```
## Warning: There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
## There are new levels in a factor: NA
```

```r
saveRDS(mlfit_knn_corr, fnames[1])

# get resampled predictive performance
mlres_knn_corr <- resample(mspec_tun_knn_corr, control = ctrl)
saveRDS(mlres_knn_corr, fnames[2])
```

```r
summary(mlres_knn_corr)
```

```
##              Statistic
## Metric             Mean    Median         SD        Min       Max NA
##   Brier       0.2341164 0.2052805 0.08017107  0.1325841 0.3538957  0
##   Accuracy    0.6789216 0.6875000 0.13395454  0.5000000 0.8750000  0
##   Kappa       0.3072910 0.3672004 0.31548204 -0.1428571 0.7333333  0
##   ROC AUC     0.7640260 0.7845238 0.10718698  0.5454545 0.9090909  0
##   Sensitivity 0.7454545 0.7500000 0.09270945  0.6000000 0.9000000  0
##   Specificity 0.5714286 0.6190476 0.28104653  0.1666667 0.8571429  0
```

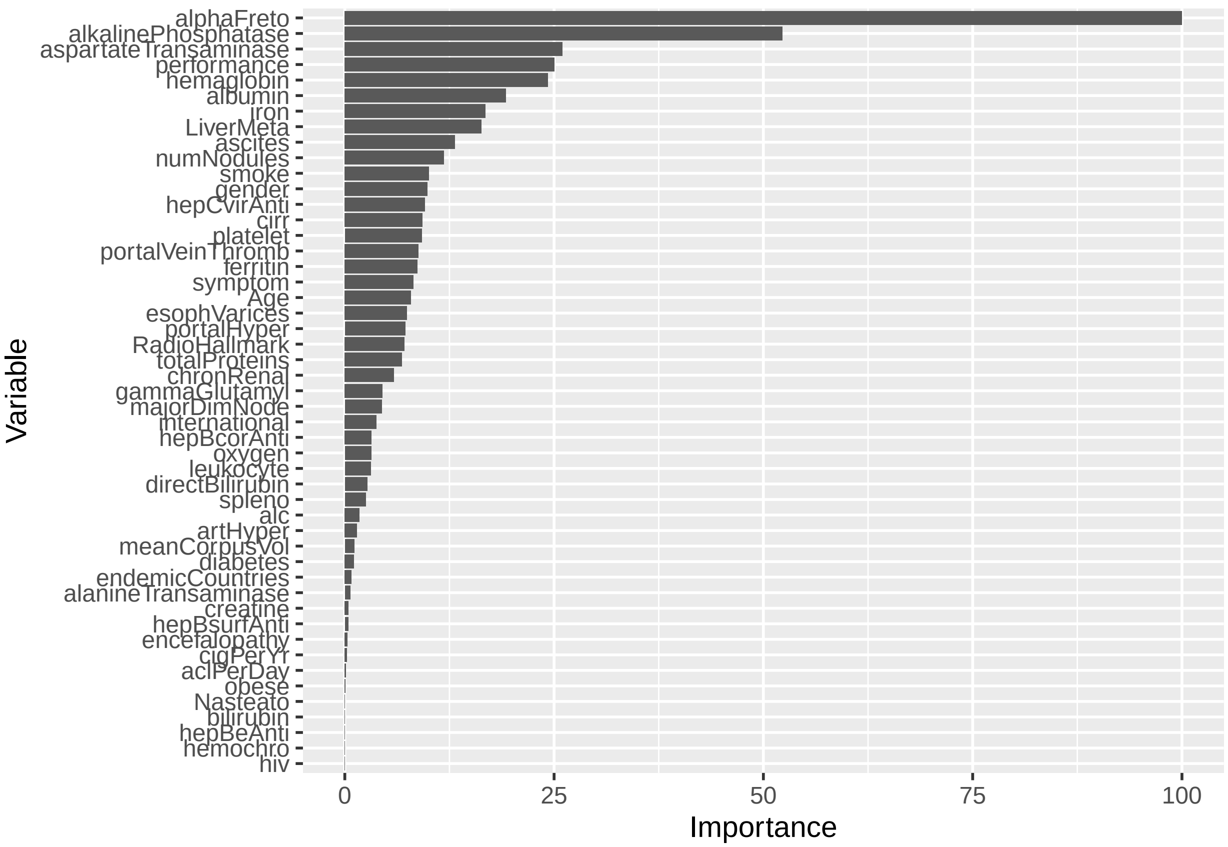
```r
(tuned_model_knn_corr <- as.MLModel(mlfit_knn_corr))
```

```
## --- MLModel object ----------------------------------------------------------
## 
## Model name: XGBTreeModel
## Label: Trained Extreme Gradient Boosting (Tree)
## Package: xgboost (>= 1.3.0)
## Response types: factor, numeric, PoissonVariate, Surv
## Case weights support: TRUE
## Tuning grid: TRUE
## Variable importance: TRUE
## 
## Parameters:
## List of 27
##  $ nrounds                   : num 50
##  $ eta                       : num 0.1
##  $ gamma                     : num 0
##  $ max_depth                 : num 8
##  $ min_child_weight          : num 1
##  $ max_delta_step            : language 0.7 * is(y, "PoissonVariate")
##  $ subsample                 : num 1
##  $ colsample_bytree          : num 1
##  $ colsample_bylevel         : num 1
##  $ colsample_bynode          : num 1
##   [list output truncated]
## 
## === $TrainingStep1 ==========================================================
## === TrainingStep object ===
## 
## Optimization method: Bayesian Optimization
## ModelSpecification log:
## # A tibble: 16 x 5
##    name         epoch selected params$input.1mQk~ $model.1qSf$nro~ metrics$Brier
##    <chr>        <int> <lgl>                 <dbl>            <dbl>         <dbl>
##  1 ModelSpec.1      0 FALSE                     4               63         0.229
##  2 ModelSpec.2      0 FALSE                     5               76         0.228
##  3 ModelSpec.3      0 FALSE                     3              132         0.244
##  4 ModelSpec.4      0 FALSE                     3               86         0.231
##  5 ModelSpec.5      0 FALSE                     2              105         0.247
##  6 ModelSpec.6      0 FALSE                     2              135         0.242
##  7 ModelSpec.7      1 FALSE                     1               67         0.208
##  8 ModelSpec.8      2 FALSE                     1               50         0.207
##  9 ModelSpec.9      3 FALSE                     1              150         0.221
## 10 ModelSpec.10     4 TRUE                         1               50         0.203
## # ... with 6 more rows, and 8 more variables:
## #   params$input.1mQk$corr <tibble[,1]>, params$model.1qSf$eta <dbl>,
## #   $$max_depth <dbl>, metrics$Accuracy <dbl>, $Kappa <dbl>, ...
##
## Selected row: 10
## Metric: Brier = 0.2033579
```

```
summary(tuned_model_knn_corr)
```

```
## --- $TrainingStep1 -------------------------------------------------------------
## # A tibble: 16 x 5
##    name         epoch selected params$input.1mQk~ $model.1qSf$nro~ metrics$Brier
##    <chr>        <int> <lgl>                 <dbl>            <dbl>         <dbl>
##  1 ModelSpec.1      0 FALSE                     4               63         0.229
##  2 ModelSpec.2      0 FALSE                     5               76         0.228
##  3 ModelSpec.3      0 FALSE                     3              132         0.244
##  4 ModelSpec.4      0 FALSE                     3               86         0.231
##  5 ModelSpec.5      0 FALSE                     2              105         0.247
##  6 ModelSpec.6      0 FALSE                     2              135         0.242
##  7 ModelSpec.7      1 FALSE                     1               67         0.208
##  8 ModelSpec.8      2 FALSE                     1               50         0.207
##  9 ModelSpec.9      3 FALSE                     1              150         0.221
## 10 ModelSpec.10     4 TRUE                      1               50         0.203
## # ... with 6 more rows, and 8 more variables:
## #   params$input.1mQk$corr <tibble[,1]>, params$model.1qSf$eta <dbl>,
## #   $$max_depth <dbl>, metrics$Accuracy <dbl>, $Kappa <dbl>, $`ROC AUC` <dbl>,
## #   $Sensitivity <dbl>, $Specificity <dbl>
```

```
# variable importance
varimp(mlfit_knn_corr) %>% plot()
```


alphaFreto
alkalinePhosphatase
aspartateTransaminase
performance
hemaglobin
albumin
iron
LiverMeta
ascites
numNodules
smoke
gender
hepCvirAnti
cirr
platelet
portalVeinThromb
ferritin
symptom
Age
esophVarices
portalHyper
RadioHallmark
totalProteins
chronRenal
gammaGlutamyl
majorDimNode
international
hepBcorAnti
oxygen
leukocyte
directBilirubin
spleno
alc
artHyper
meanCorpusVol
diabetes
endemicCountries
alanineTransaminase
creatine
hepBsurfAnti
encefalopathy
cigPerYr
aclPerDay
obese
Nasteato
bilirubin
hepBeAnti
hemochro
hiv
Variable
0
25
50
75
100
Importance

corr-1.pdf