Featurte importance / Feature selection

In machine learning and statistics, feature selection, also known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of relevant features (variables, predictors) for use in model construction. Feature selection techniques are used for several reasons:

simplification of models to make them easier to interpret by researchers/users shorter training times *to avoid the curse of dimensionality* improve data's compatibility with a learning model class *encode inherent symmetries present in the input space.

we know from the prediction that the rf and lda and glm models have the best performance compared to the other models, so we will only apply the Feaure selection algorithm for these 3 models.

Load Data

```
workPath ="D:\\DataMining\\"
cardio <- readRDS(paste (workPath , "cardio.rds", sep =""))
```

Train test split

```
set.seed(1001)

cardio_complete <- cardio[(complete.cases(cardio)),]

nrow(cardio_complete)

## [1] 2927

n_test <- 1500

idx_test <- sample(1:nrow(cardio_complete), n_test)

cardio_test <- cardio_complete[ idx_test,]
cardio_train <- cardio_complete[ -idx_test,]


control <- trainControl(method="cv", number=5)
metric <- "Accuracy"
```

Greedy Forward Selection

Forward stepwise selection is a variable selection method which:

Begins with a model that contains no variables (called the Null Model) Then starts adding the most significant variables one after the other Until a pre-specified stopping rule is reached or until all the variables under consideration are included in the mode.

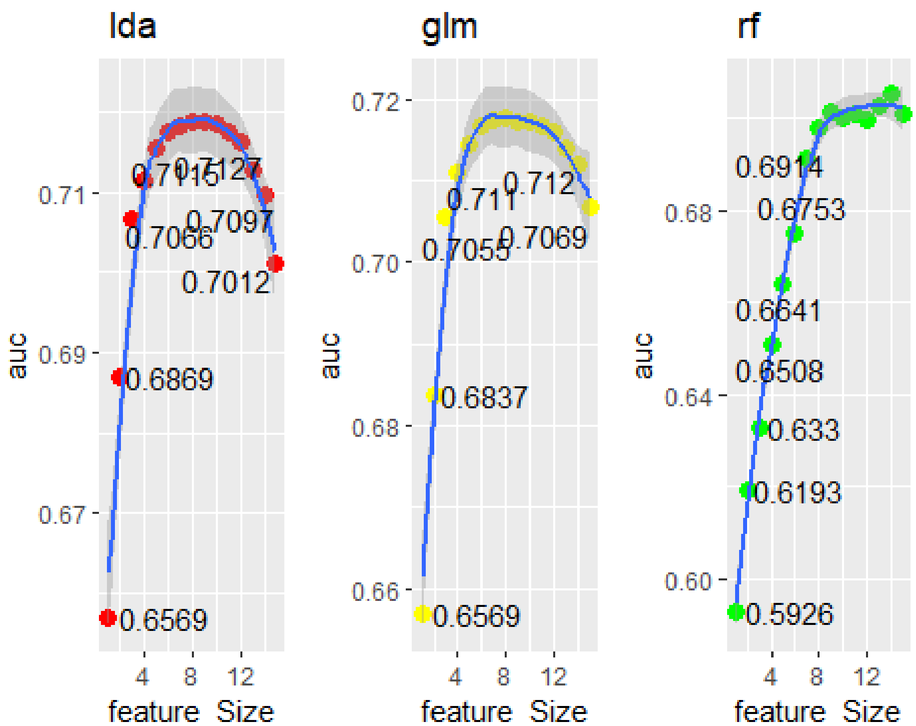
```
computeData <- F
workPath = "D:\\DataMining\\Seafile\\01_cardio\\Projekt\\"

modell <- c("glm", "lda", "rf")

if (computeData) {

sff_performanceDf_Forward = expand.grid(feature_Size = seq(1, 15), model =
modell)


sff_performanceDf_Forward$auc <- NA

sff_performanceDf_Forward$festure <- ""

sff_performanceDf_Forward$Bestfesture <- NA

cardio_train_rf <- cardio_train

nbr_features <- 15
aucperfor <- NA



for (model in  c("glm", "lda", "rf"))
{
  sff_featureset <- c()

  while (length(sff_featureset) < nbr_features) {
    featutes_to_test <- setdiff(rel_features, sff_featureset)

    aucperfor <- rep(NA, length(featutes_to_test))

    for (i_featutes_to_test  in 1:length(featutes_to_test)) {
      tmp_feat_set <-
        c(sff_featureset, featutes_to_test[i_featutes_to_test])

      mod <-
        train(
          target ~ .,
          data = cardio_train_rf[, c(tmp_feat_set , "target")],
          method = model ,
          metric = metric,
          trControl = control
```

```r
    )

    y_pred_prob <- predict(mod ,  cardio_test, type = "prob")
    auc <-   roc(cardio_test$target , y_pred_prob[, 1])
    aucperfor[i_featutes_to_test] <- auc$auc

  }

  best_idx <-   which.max(aucperfor)

  print (paste(
    "new Feature:" ,
    featutes_to_test[best_idx],
    " with AUC:",
    max(aucperfor)
  ))
  sff_featureset <- c(sff_featureset, featutes_to_test[best_idx])

  print (paste("FeatureSet Size:" , length(sff_featureset)))

  print (paste("FeatureSet :" , paste(sff_featureset, collapse = ",")))

  print(paste("model :", model))


  if (model == "glm") {
    sff_performanceDf_Forward$Bestfesture[length(sff_featureset)] <-
      featutes_to_test[best_idx]
    sff_performanceDf_Forward$auc[length(sff_featureset)] <-
      max(aucperfor)
    sff_performanceDf_Forward$festure[length(sff_featureset)] <-
      paste(sff_featureset, sep = ",", collapse = ",")
  }


  if (model == "lda") {
    sff_performanceDf_Forward$Bestfesture[15 + length(sff_featureset)] <-
      featutes_to_test[best_idx]
    sff_performanceDf_Forward$auc[15 + length(sff_featureset)] <-
      max(aucperfor)
    sff_performanceDf_Forward$festure[15 + length(sff_featureset)] <-
      paste(sff_featureset, sep = ",", collapse = ",")
  }


  if (model == "rf") {
    sff_performanceDf_Forward$Bestfesture[30 + length(sff_featureset)] <-
      featutes_to_test[best_idx]
    sff_performanceDf_Forward$auc[30 + length(sff_featureset)] <-
```

```
        max(aucperfor)
      sff_performanceDf_Forward$festure[30 + length(sff_featureset)] <-
        paste(sff_featureset, sep = ",", collapse = ",")
    }

  }
}
saveRDS(sff_performanceDf_Forward, file  = paste (workPath ,
"sff_performanceDf_Forward.rds", sep = ""))

}
sff_performanceDf_Forward <-
  readRDS(paste (workPath , "sff_performanceDf_Forward.rds", sep = ""))


#Diagramm lda
lda_forward <-
  sff_performanceDf_Forward[sff_performanceDf_Forward$model   == "lda",]
gg_lda_forward <-
  ggplot(lda_forward , aes(x = feature_Size , y = auc)) +
  geom_point(colour = "red", size = 3) +
  geom_smooth() + ggtitle(" lda") +
  geom_text_repel(label = round(lda_forward$auc, 4))

#Diagramm glm
glm_forward <-
  sff_performanceDf_Forward[sff_performanceDf_Forward$model   == "glm",]
gg_glm_forward <-
  ggplot(glm_forward, aes(x = feature_Size , y = auc)) +
  geom_point(colour = "yellow", size = 3) +
  geom_smooth() + ggtitle(" glm") +
  geom_text_repel(label = round(glm_forward$auc, 4))


#Diagramm rf
rf_forward <-
  sff_performanceDf_Forward[sff_performanceDf_Forward$model   == "rf",]
gg_rf_forward <-
  ggplot(rf_forward, aes(x = feature_Size , y = auc)) +
  geom_point(colour = "green", size = 3) +
  geom_smooth() + ggtitle(" rf") +
  geom_text_repel(label = round(rf_forward$auc, 4))

#Diagramm
gg_lda_forward + gg_glm_forward + gg_rf_forward
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
## Warning: ggrepel: 8 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps

## Warning: ggrepel: 9 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps

## Warning: ggrepel: 8 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```




```
#performanceModell

glm
```

	feature_Size	model	auc	festure	Bestfesture
1	1	glm	0.6568708	age	age
2	2	glm	0.6837093	age,sysBP	sysBP
3	3	glm	0.7055242	age,sysBP,cigsPerDay	cigsPerDay
4	4	glm	0.7110169	age,sysBP,cigsPerDay,sex	sex
5	5	glm	0.7143704	age,sysBP,cigsPerDay,sex,diabetes	diabetes
6	6	glm	0.7166704	age,sysBP,cigsPerDay,sex,diabetes,totChol	totChol
7	7	glm	0.7174970	age,sysBP,cigsPerDay,sex,diabetes,totChol,smoking	smoking
8	8	glm	0.7176460	age,sysBP,cigsPerDay,sex,diabetes,totChol,smoking,BMI	BMI
9	9	glm	0.7173344	age,sysBP,cigsPerDay,sex,diabetes,totChol,smoking,BMI,BloodPresMed	BloodPresMed
10	10	glm	0.7171176	age,sysBP,cigsPerDay,sex,diabetes,totChol,smoking,BMI,BloodPresMed,stroke	stroke
11	11	glm	0.7167721	age,sysBP,cigsPerDay,sex,diabetes,totChol,smoking,BMI,BloodPresMed,stroke,heartRate	heartRate
12	12	glm	0.7159659	age,sysBP,cigsPerDay,sex,diabetes,totChol,smoking,BMI,BloodPresMed,stroke,heartRate,education	education
13	13	glm	0.7139063	age,sysBP,cigsPerDay,sex,diabetes,totChol,smoking,BMI,BloodPresMed,stroke,heartRate,education,diaBP	diaBP
14	14	glm	0.7120128	age,sysBP,cigsPerDay,sex,diabetes,totChol,smoking,BMI,BloodPresMed,stroke,heartRate,education,diaBP,hypertensive	hypertensive
15	15	glm	0.7068537	age,sysBP,cigsPerDay,sex,diabetes,totChol,smoking,BMI,BloodPresMed,stroke,heartRate,education,diaBP,hypertensive,gluc...	glucose

##Interpretation:

Das Modell glm hat die beste leistung mit 8 feature, ,also

```
"age,sysBP,cigsPerDay,sex,diabetes,totChol,smoking,BMI"
```

und die Auc beträgt 0,7176460.

Das Modell lda hat die beste leistung mit 8 feature, ,also

age,sysBP,cigsPerDay,diabetes,sex,totChol,smoking,BMI

und die Auc beträgt 0.7187706.

**Der Unterschied besteht darin, dass im Modell glm das sex 4-stellig ist und im Modell lda 5-stellig und das Model lda hat bessere Leistung.

Das Modell rf hat die beste leistung mit 12 feature, ,also

```
"sysBP,sex,diabetes,cigsPerDay,stroke,age,BMI,BloodPresMed,smoking,totChol,di
aBP,hypertensive"
```

und die Auc beträgt 0.7013221.