Cervical cancer - A multilabel classification

mlr(Machine Learning in R) 패키지 설명

- https://www.rdocumentation.org/packages/mlr/versions/2.12.1
- machine-readable parameter descriptions 을 포함해서 다수의 분류와 회귀 방법들과 인터페이스를 제공
- 생존분석, 군집 및 example-specific cost-sensitive learning(예제 관련 비용에 민감한 학습)을 위한 실험적인 확장됨.
- cross-validation, bootstrapping and subsampling 을 포함한 일반적인 resampling 을 지원.
- 단일 및 다중 목표 문제에 대한 최신 최적화 기법을 사용한 하이퍼 매개 변수 튜닝 지원.
- 기능 선택을위한 필터 및 래퍼 메소드 지원.
- 기계 학습에서 공통적인 추가 작업으로 기본 학습자를 확장하고 쉽게 중첩된 재샘플링을 허용.
- 대부분의 작업을 병렬 처리 => 한방 패키지.

데이터 설명

- UCI repository 에서 가지고 옴
- 858 개의 Case 와 36 개의 변수
- 타켓변수 : Biopsy(생검), Citology(??), Schiller(??), Hinselmann(??)

```
# R 3.3.x 에서는 오류 R 3.5.0 에서 실행시킴

#install.packages('gbm')  #  Generalized Boosted Regression Models
#install.packages('randomForestSRC') # Random Forests for Survival, Regressio
n, and Classification
#install.packages('DataExplorer')
#install.packages('tidyverse')
#install.packages('mlr')
Sys.setlocale('LC_ALL','C')
```

```
## [1] "C"
```

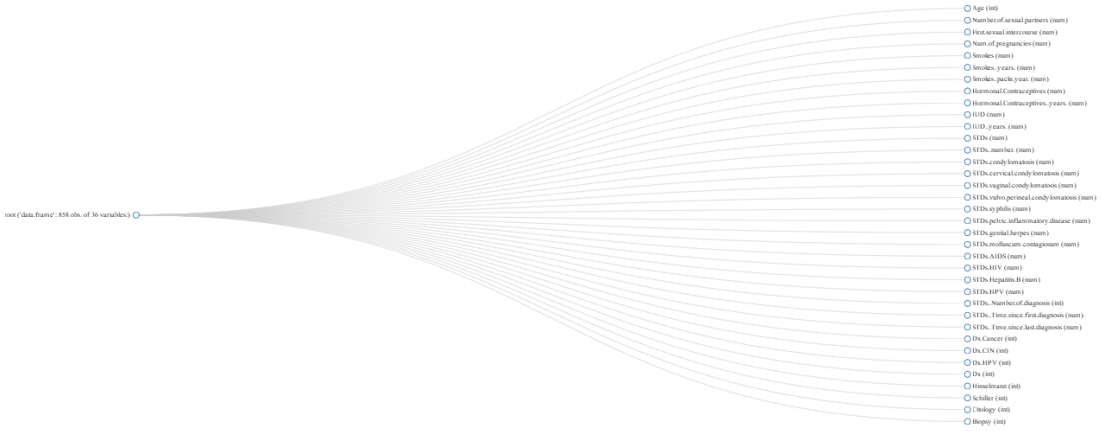
```
Sys.getlocale()
```

```
## [1] "C"
```

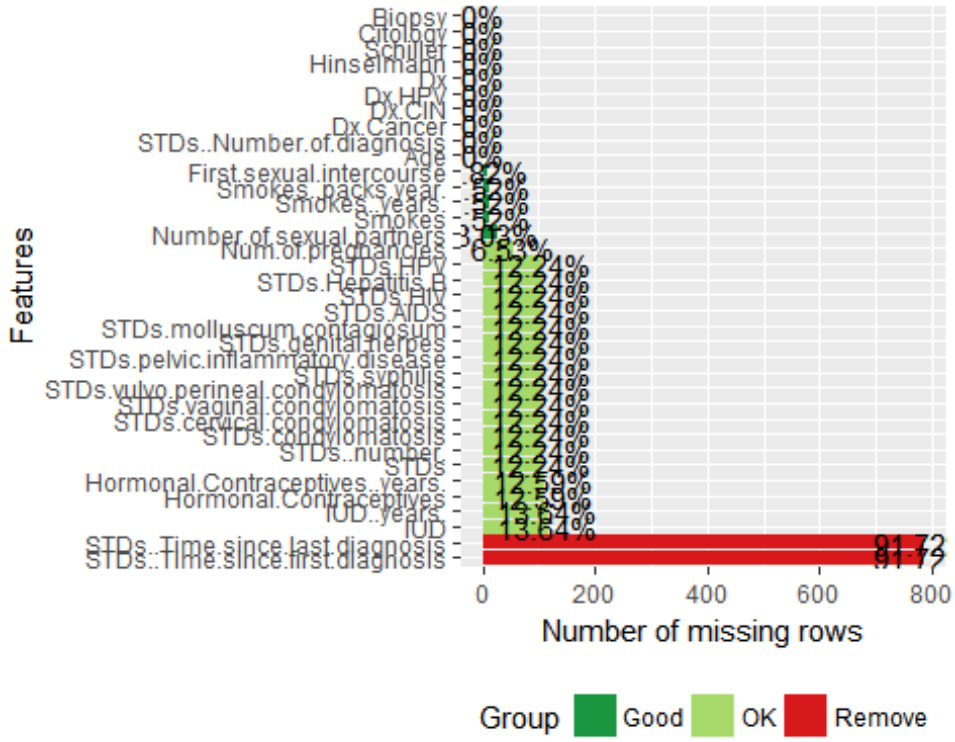
```
setwd('D:/Work_Git/DeepMenia/part02/week1_180608')

library(DataExplorer)
library(tidyverse)
library(mlr)
df <- read.csv("kag_risk_factors_cervical_cancer.csv", header = T, sep =',',
na.strings = '?')
```

데이터 탐색

```
plot_str(df)
```




```
plot_missing(df)
```




```
summary(df)
```

```
##       Age        Number.of.sexual.partners First.sexual.intercourse
##  Min.   :13.00   Min.   : 1.000            Min.   :10
##  1st Qu.:20.00   1st Qu.: 2.000            1st Qu.:15
##  Median :25.00   Median : 2.000            Median :17
##  Mean   :26.82   Mean   : 2.528            Mean   :17
##  3rd Qu.:32.00   3rd Qu.: 3.000            3rd Qu.:18
##  Max.   :84.00   Max.   :28.000            Max.   :32
##                  NA's   :26                NA's   :7
##  Num.of.pregnancies     Smokes       Smokes..years.  Smokes..packs.year.
##  Min.   : 0.000     Min.   :0.0000   Min.   : 0.00   Min.   : 0.0000
##  1st Qu.: 1.000     1st Qu.:0.0000   1st Qu.: 0.00   1st Qu.: 0.0000
##  Median : 2.000     Median :0.0000   Median : 0.00   Median : 0.0000
##  Mean   : 2.276     Mean   :0.1456   Mean   : 1.22   Mean   : 0.4531
##  3rd Qu.: 3.000     3rd Qu.:0.0000   3rd Qu.: 0.00   3rd Qu.: 0.0000
##  Max.   :11.000     Max.   :1.0000   Max.   :37.00   Max.   :37.0000
##  NA's   :56         NA's   :13       NA's   :13      NA's   :13
##  Hormonal.Contraceptives Hormonal.Contraceptives..years.      IUD
##  Min.   :0.0000          Min.   : 0.000                  Min.   :0.000
##  1st Qu.:0.0000          1st Qu.: 0.000                  1st Qu.:0.000
##  Median :1.0000          Median : 0.500                  Median :0.000
##  Mean   :0.6413          Mean   : 2.256                  Mean   :0.112
##  3rd Qu.:1.0000          3rd Qu.: 3.000                  3rd Qu.:0.000
##  Max.   :1.0000          Max.   :30.000                  Max.   :1.000
##  NA's   :108             NA's   :108                     NA's   :117
```

```
##   IUD..years.           STDs        STDs..number.    STDs.condylomatosis
##  Min.   : 0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.: 0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000
##  Median : 0.0000   Median :0.0000   Median :0.0000   Median :0.00000
##  Mean   : 0.5148   Mean   :0.1049   Mean   :0.1766   Mean   :0.05843
##  3rd Qu.: 0.0000   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.00000
##  Max.   :19.0000   Max.   :1.0000   Max.   :4.0000   Max.   :1.00000
##  NA's   :117       NA's   :105      NA's   :105      NA's   :105
##  STDs.cervical.condylomatosis STDs.vaginal.condylomatosis
##  Min.   :0                    Min.   :0.00000
##  1st Qu.:0                    1st Qu.:0.00000
##  Median :0                    Median :0.00000
##  Mean   :0                    Mean   :0.00531
##  3rd Qu.:0                    3rd Qu.:0.00000
##  Max.   :0                    Max.   :1.00000
##  NA's   :105                  NA's   :105
##  STDs.vulvo.perineal.condylomatosis STDs.syphilis
##  Min.   :0.0000                     Min.   :0.0000
##  1st Qu.:0.0000                     1st Qu.:0.0000
##  Median :0.0000                     Median :0.0000
##  Mean   :0.0571                     Mean   :0.0239
##  3rd Qu.:0.0000                     3rd Qu.:0.0000
##  Max.   :1.0000                     Max.   :1.0000
##  NA's   :105                        NA's   :105
##  STDs.pelvic.inflammatory.disease STDs.genital.herpes
##  Min.   :0.00000                  Min.   :0.00000
##  1st Qu.:0.00000                  1st Qu.:0.00000
##  Median :0.00000                  Median :0.00000
##  Mean   :0.00133                  Mean   :0.00133
##  3rd Qu.:0.00000                  3rd Qu.:0.00000
##  Max.   :1.00000                  Max.   :1.00000
##  NA's   :105                      NA's   :105
##  STDs.molluscum.contagiosum   STDs.AIDS      STDs.HIV
##  Min.   :0.00000            Min.   :0     Min.   :0.0000
##  1st Qu.:0.00000            1st Qu.:0     1st Qu.:0.0000
##  Median :0.00000            Median :0     Median :0.0000
##  Mean   :0.00133            Mean   :0     Mean   :0.0239
##  3rd Qu.:0.00000            3rd Qu.:0     3rd Qu.:0.0000
##  Max.   :1.00000            Max.   :0     Max.   :1.0000
##  NA's   :105                NA's   :105   NA's   :105
##  STDs.Hepatitis.B     STDs.HPV       STDs..Number.of.diagnosis
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.00133   Mean   :0.00266   Mean   :0.08741
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :3.00000
##  NA's   :105       NA's   :105
##  STDs..Time.since.first.diagnosis STDs..Time.since.last.diagnosis
```

```
##  Min.   : 1.000                  Min.   : 1.000
##  1st Qu.: 2.000                  1st Qu.: 2.000
##  Median : 4.000                  Median : 3.000
##  Mean   : 6.141                  Mean   : 5.817
##  3rd Qu.: 8.000                  3rd Qu.: 7.500
##  Max.   :22.000                  Max.   :22.000
##  NA's   :787                     NA's   :787
##    Dx.Cancer           Dx.CIN            Dx.HPV              Dx
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.02098   Mean   :0.01049   Mean   :0.02098   Mean   :0.02797
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##
##    Hinselmann         Schiller          Citology           Biopsy
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.0000
##  Mean   :0.04079   Mean   :0.08625   Mean   :0.05128   Mean   :0.0641
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.0000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.0000
##
```

데이터 변환 및 준비

```
# changing variables into factor datatype
col <- c(5,8,10,12,14:25,29:36)
df1 <- df
df1[col] <- lapply(df1[col], factor)



# creating id variable for the instances
Id <- c(1:858)
df1 <- cbind(df1,Id)
df1 <- df1 %>%
  select(Id, everything())


head( df1$Biopsy )
```

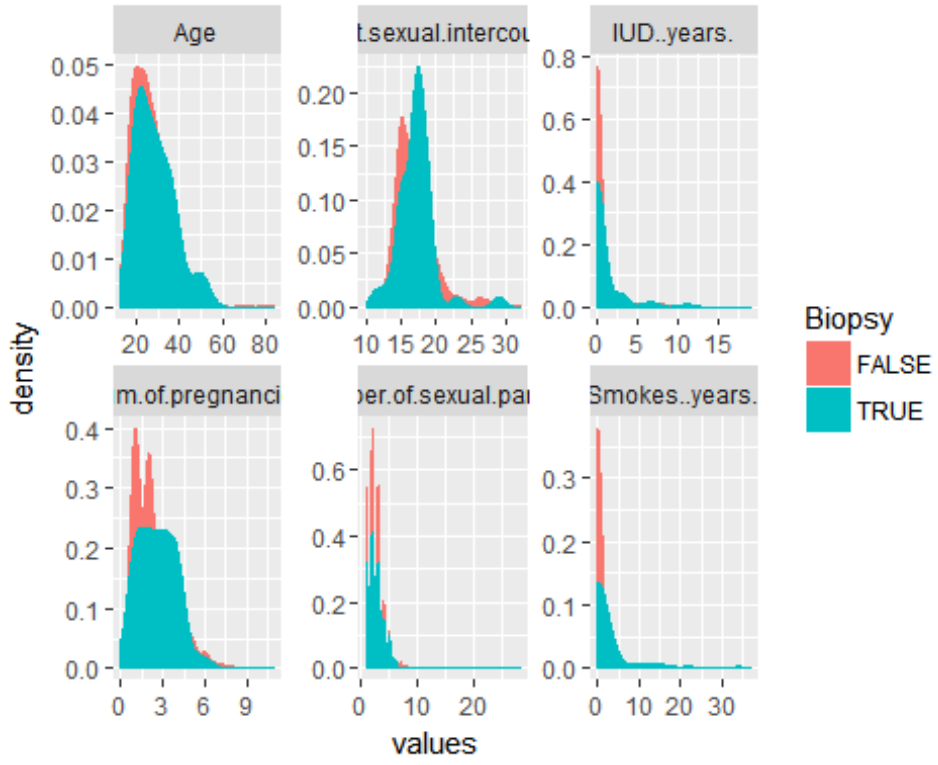
```
## [1] 0 0 0 0 0 0
## Levels: 0 1
```

```
# changing target variables into logical datatype(To work with multilabel cla
ssification)
df1$Biopsy <- as.logical(as.integer(as.character(df1$Biopsy)))
df1$Hinselmann <- as.logical(as.integer(as.character(df1$Hinselmann)))
```

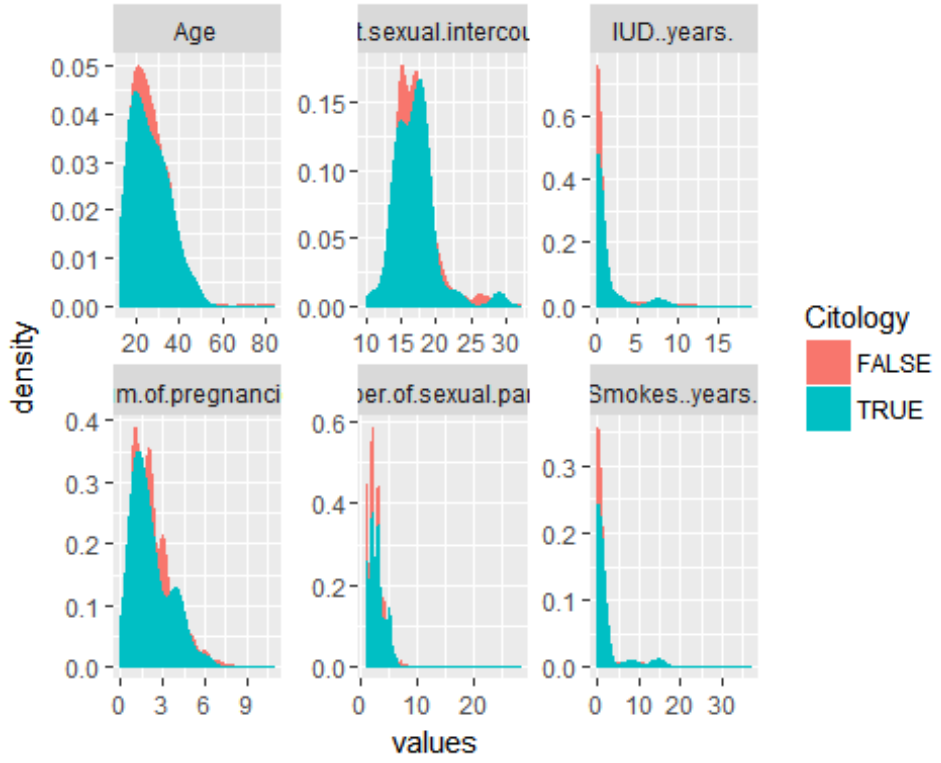
```
df1$Schiller <- as.logical(as.integer(as.character(df1$Schiller)))
df1$Citology <- as.logical(as.integer(as.character(df1$Citology)))

head( df1$Biopsy )

## [1] FALSE FALSE FALSE FALSE FALSE FALSE

# removing the variables with more than 90% of missing values
df1$STDs..Time.since.first.diagnosis <- NULL
df1$STDs..Time.since.last.diagnosis <- NULL
```

데이터 시각화

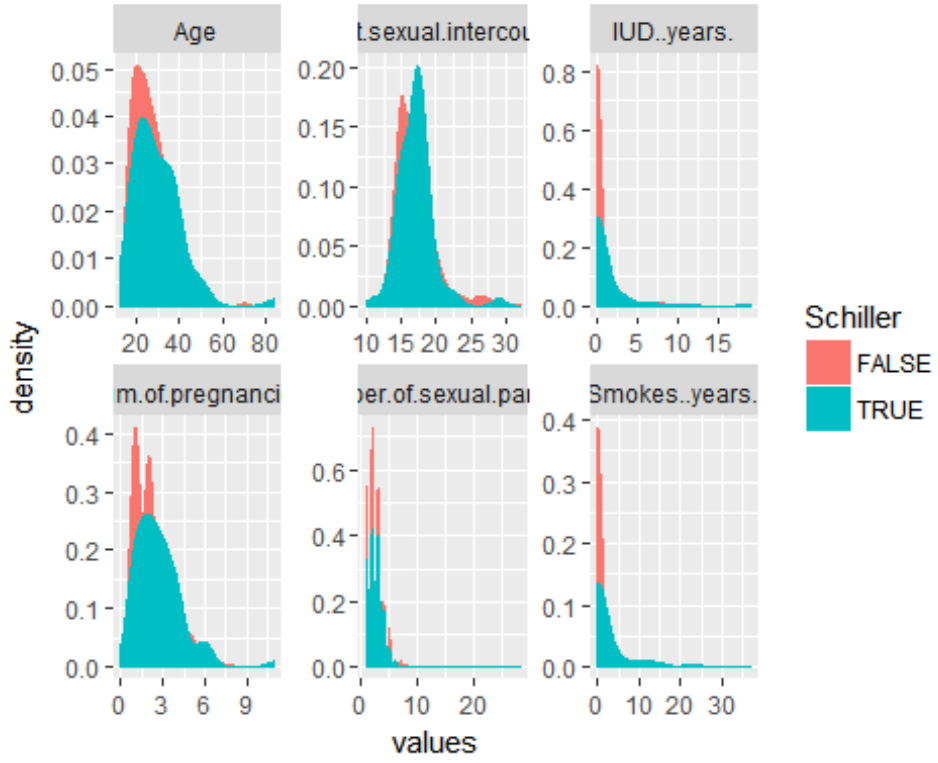
```
# ploting density distributtion of continous variables with target variables

df1mod <- df1 %>% gather(c(Age:Num.of.pregnancies,Smokes..years.,IUD..years.),
 key='Variables', value='values')

df1mod %>% ggplot(aes(x= values, fill= Biopsy,color = Biopsy)) + geom_density
() +
  facet_wrap(~Variables,ncol=3,scales="free")
```



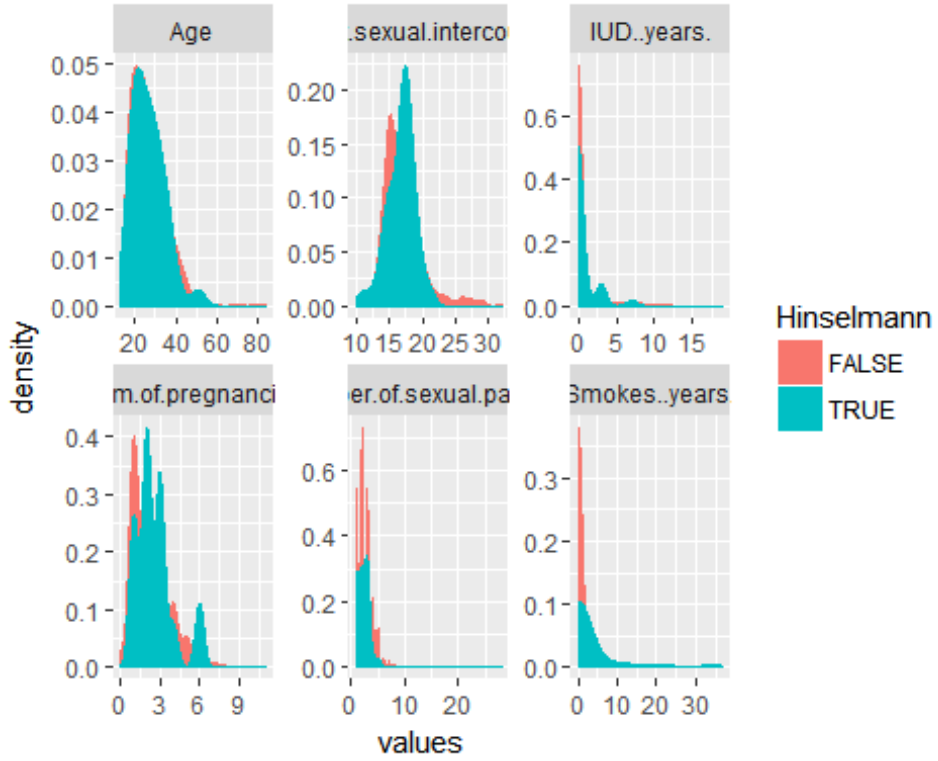

```
df1mod %>% ggplot(aes(x= values, fill= Citology,color = Citology)) + geom_den
sity() +
  facet_wrap(~Variables,ncol=3,scales="free")
```



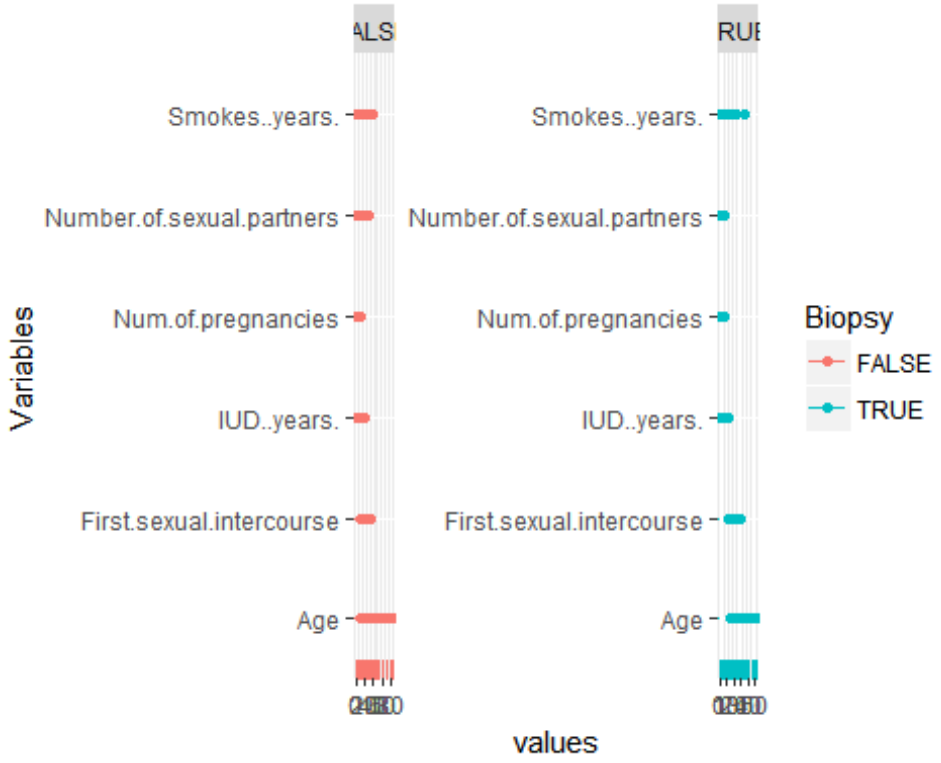

```
df1mod %>% ggplot(aes(x= values, fill= Schiller,color = Schiller)) + geom_den
sity() +
  facet_wrap(~Variables,ncol=3,scales="free")
```



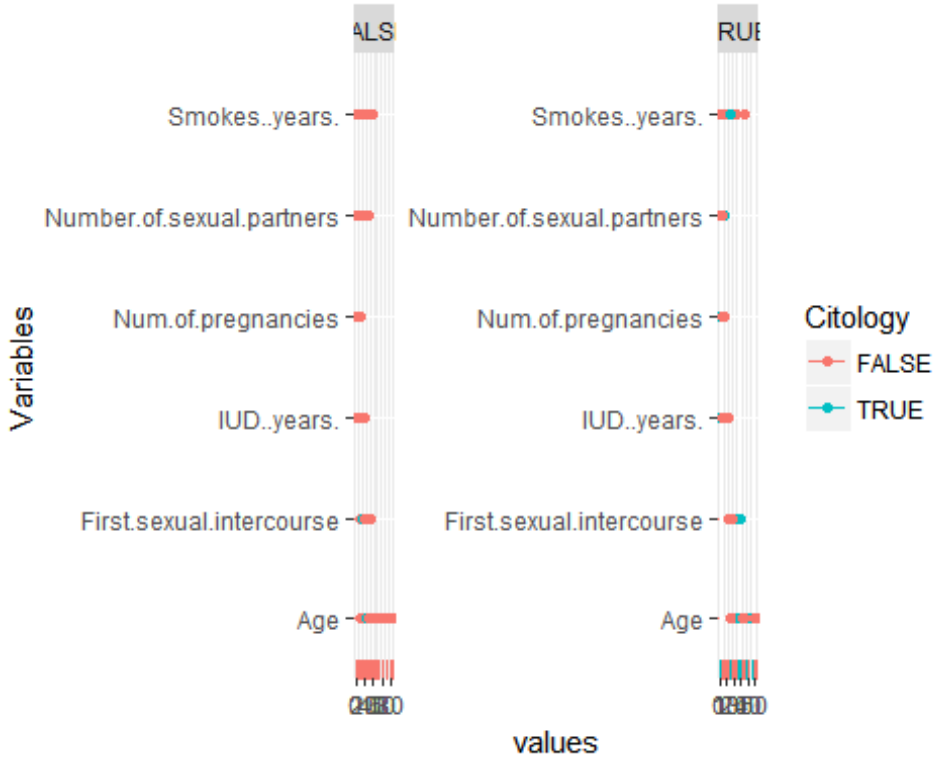

```
df1mod %>% ggplot(aes(x= values, fill= Hinselmann,color = Hinselmann)) + geom
_density() +
  facet_wrap(~Variables,ncol=3,scales="free")
```



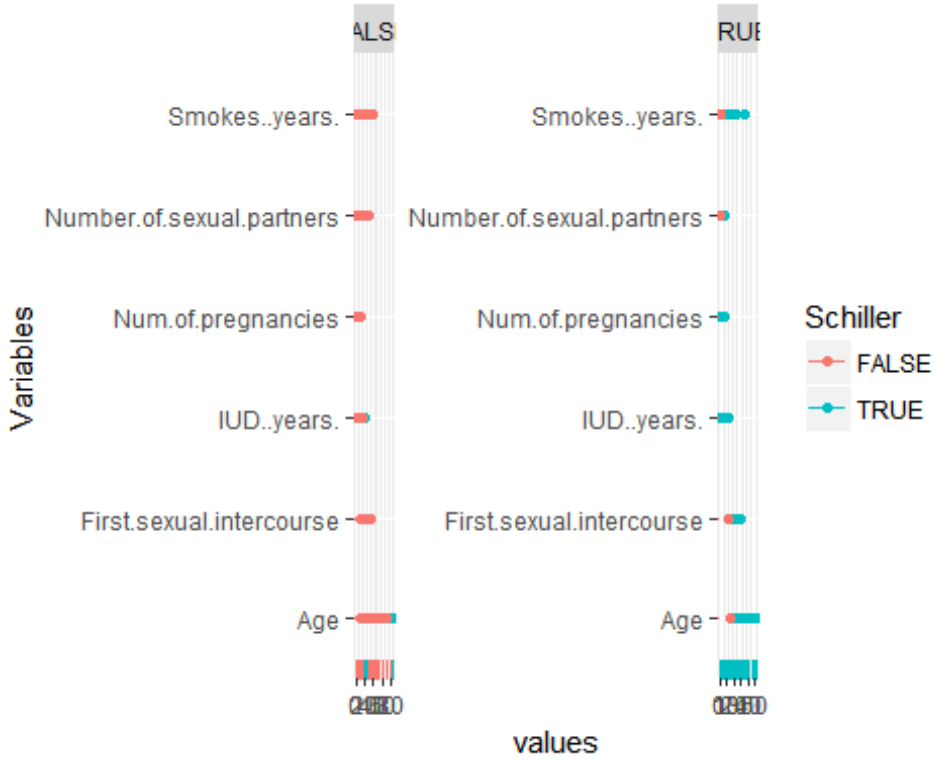

```
df1mod %>% ggplot(aes(x = Variables,y= values, fill= Biopsy,color = Biopsy))
+ geom_point() +
  facet_wrap(~Biopsy,ncol=3,scales="free") + geom_rug() + coord_flip()
```



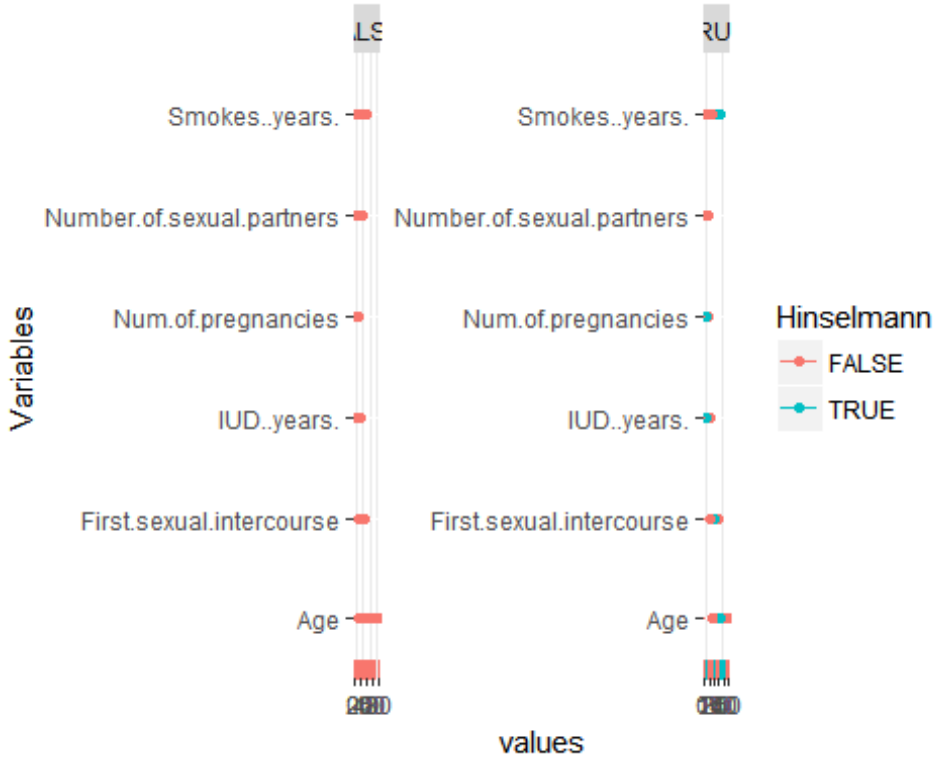

```
df1mod %>% ggplot(aes(x = Variables,y= values, fill=Citology,color = Citolog
y)) + geom_point() +
  facet_wrap(~Biopsy,ncol=3,scales="free") + geom_rug() + coord_flip()
```




```
df1mod %>% ggplot(aes(x = Variables,y= values, fill= Schiller ,color = Schiller)) + geom_point() +
  facet_wrap(~Biopsy,ncol=3,scales="free") + geom_rug() + coord_flip()
```




```
df1mod %>% ggplot(aes(x = Variables,y= values, fill= Hinselmann ,color = Hins
elmann)) + geom_point() +
  facet_wrap(~Biopsy,ncol=3,scales="free") + geom_rug() + coord_flip()
```



머신러닝하기 위한 데이터 전처리

```
# target variables are grouped together to form lables
label <- colnames(df1)[32:35]

label

## [1] "Hinselmann" "Schiller"   "Citology"   "Biopsy"

# dataset parition for training and prediction
set.seed(1234)
index <- sample(nrow(df1), nrow(df1)*0.7)
train <- df1[index,]
test <- df1[-index,]
train.set <- row.names(train)%>%as.integer()
```

머신러닝 모델 적용

- mlr 라이브러리의 method 을 사용해서 classification.

Method 1 - randomforest SRC

```
traintask <- makeMultilabelTask(id ='multi', data = df1, target= label)
Rf.learner <- makeLearner('multilabel.randomForestSRC',predict.type = 'prob')

model1 <- mlr::train(Rf.learner,task = traintask, subset = train.set)
```

```r
prediction1 <- predict(model1,newdata = test)
# hamloss is not that imperssive as well as other metrics - acc, auc
performance(prediction1)
```

```
## multilabel.hamloss
##         0.06589147
```

```r
getMultilabelBinaryPerformances(prediction1, measures = list(acc, mmce, auc))
```

```
##            acc.test.mean mmce.test.mean auc.test.mean
## Hinselmann     0.9534884     0.04651163     0.5535230
## Schiller       0.9031008     0.09689922     0.6051804
## Citology       0.9457364     0.05426357     0.4519906
## Biopsy         0.9341085     0.06589147     0.6730372
```

random forest SRC 의 hyper parameter tuning 을 사용해서 예측

```r
# checking through available parameters for hypertuning
getParamSet(Rf.learner)
```

```
##                      Type  len       Def
## ntree             integer    -      1000
## bootstrap        discrete    -   by.root
## mtry              integer    -    <NULL>
## nodesize          integer    -    <NULL>
## nodedepth         integer    -    <NULL>
## splitrule        discrete    -      NULL
## nsplit            integer    -         0
## split.null        logical    -     FALSE
## importance       discrete    -     FALSE
## na.action        discrete    - na.impute
## nimpute           integer    -         1
## proximity        discrete    -     FALSE
## sampsize          integer    -    <NULL>
## samptype         discrete    -       swr
## samp              untyped    -         -
## xvar.wt       numericvector <NA>    <NULL>
## forest            logical    -      TRUE
## var.used         discrete    -     FALSE
## split.depth      discrete    -     FALSE
## seed              integer    -    <NULL>
## do.trace          logical    -     FALSE
## membership        logical    -     FALSE
## statistics        logical    -     FALSE
## tree.err          logical    -     FALSE
## coerce.factor      untyped    -    <NULL>
##                                                Constr Req Tunable Trafo
## ntree                                        1 to Inf   -    TRUE     -
## bootstrap          by.root,by.node,none,by.user   -    TRUE     -
## mtry                                         1 to Inf   -    TRUE     -
```

```
## nodesize                                        1 to Inf   -    TRUE     -
## nodedepth                                     -Inf to Inf   -    TRUE     -
## splitrule         gini,gini.unwt,gini.hvwt,random,NULL   -    TRUE     -
## nsplit                                          0 to Inf   Y    TRUE     -
## split.null                                             -   -    TRUE     -
## importance    FALSE,TRUE,none,permute,random,anti,p...   -   FALSE     -
## na.action                              na.omit,na.impute   -    TRUE     -
## nimpute                                         1 to Inf   -    TRUE     -
## proximity                      inbag,oob,all,TRUE,FALSE   -   FALSE     -
## sampsize                                        1 to Inf   Y    TRUE     -
## samptype                                        swr,swor   Y    TRUE     -
## samp                                                   -   Y    TRUE     -
## xvar.wt                                         0 to Inf   -    TRUE     -
## forest                                                 -   -   FALSE     -
## var.used                        FALSE,all.trees,by.tree   -   FALSE     -
## split.depth                     FALSE,all.trees,by.tree   -   FALSE     -
## seed                                          -Inf to 0   -   FALSE     -
## do.trace                                               -   -   FALSE     -
## membership                                             -   -   FALSE     -
## statistics                                             -   -   FALSE     -
## tree.err                                               -   -   FALSE     -
## coerce.factor                                          -   -    TRUE     -

# making resampling strategy
res <- makeResampleDesc('CV', iter = 3, stratify = F)
ream.learner <- resample(Rf.learner, task = traintask, resampling = res,show.
info = T)
# random grid search
ctrl <- makeTuneControlRandom(maxit = 5L)
# parameter set definition
params <- makeParamSet(makeIntegerParam('ntree',upper=2000,lower=1200),makeIn
tegerParam('mtry', upper=8,lower=3))
# tuning the model
tuning <- tuneParams(learner= Rf.learner,task= traintask,par.set = params, re
sampling = res, control= ctrl,
                     measures=list(multilabel.subset01, multilabel.hamloss, m
ultilabel.acc,
                                   multilabel.f1, timepredict), show.info = T)
# taking the optimum parameters
tuning$x

## $ntree
## [1] 1438
##
## $mtry
## [1] 3

# traing the model with selected parameter
learnerparset <- setHyperPars(learner = Rf.learner, par.vals = tuning$x)
```

```
mod.learner <- mlr::train(learner = learnerparset, task = traintask, subset =
 train.set)
modprediction <- predict(mod.learner, newdata= test)

# hamloss is significantly reduced and acc & auc has increased
performance(modprediction,measures = list(multilabel.subset01, multilabel.ham
loss, multilabel.acc, multilabel.f1))
```

```
## multilabel.subset01  multilabel.hamloss      multilabel.acc
##           0.1201550           0.0629845           0.8798450
##       multilabel.f1
##           0.8798450
```

```
getMultilabelBinaryPerformances(modprediction, measures = list(acc, mmce, au
c))
```

```
##            acc.test.mean mmce.test.mean auc.test.mean
## Hinselmann     0.9534884     0.04651163     0.4579946
## Schiller       0.9108527     0.08914729     0.6392229
## Citology       0.9457364     0.05426357     0.4578454
## Biopsy         0.9379845     0.06201550     0.6949897
```

Method 2 - Problem transformation method

Binary relevance method

```
# creating basic learner and wraping the learner with multilabel probel trans
formation procedures
basic.learner <- makeLearner('classif.gbm', predict.type ='prob',fix.factors.
prediction=TRUE)

binrelv.learner <- makeMultilabelBinaryRelevanceWrapper(basic.learner)
model2 <- mlr::train(binrelv.learner, task =  traintask, subset = train.set)
```

```
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
```

```
prediction2 <- predict(model2,newdata = test)
performance(prediction2,measures = list(multilabel.subset01, multilabel.hamlo
ss, multilabel.acc, multilabel.f1))
```

```
## multilabel.subset01  multilabel.hamloss      multilabel.acc
##           0.1201550           0.0629845           0.8798450
##       multilabel.f1
##           0.8798450
```

```
getMultilabelBinaryPerformances(prediction2, measures = list(acc, mmce, auc))
```

```
##            acc.test.mean mmce.test.mean auc.test.mean
## Hinselmann     0.9534884     0.04651163     0.6089092
## Schiller       0.9108527     0.08914729     0.7045328
## Citology       0.9457364     0.05426357     0.6312939
## Biopsy         0.9379845     0.06201550     0.7475465
```

Method 3 - Classifier chains method

```
classchain.learner <- makeMultilabelClassifierChainsWrapper(basic.learner)
model3 <- mlr::train(classchain.learner, task =  traintask, subset = train.se
t)

## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...

prediction3 <- predict(model3,newdata = test)
performance(prediction3,measures = list(multilabel.subset01, multilabel.hamlo
ss, multilabel.acc, multilabel.f1))

## multilabel.subset01  multilabel.hamloss      multilabel.acc
##           0.1201550           0.0629845           0.8798450
##       multilabel.f1
##           0.8798450

getMultilabelBinaryPerformances(prediction3, measures = list(acc, mmce, auc))

##            acc.test.mean mmce.test.mean auc.test.mean
## Hinselmann     0.9534884     0.04651163     0.5000000
## Schiller       0.9108527     0.08914729     0.6025902
## Citology       0.9457364     0.05426357     0.6195843
## Biopsy         0.9379845     0.06201550     0.5000000
```

Method 4 - Dependent binary relevance method

```
depbin.learner <- makeMultilabelDBRWrapper(basic.learner)
model4 <- mlr::train(depbin.learner, task =  traintask, subset = train.set)

## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...

prediction4 <- predict(model4,newdata = test)
performance(prediction4,measures = list(multilabel.subset01, multilabel.hamlo
ss, multilabel.acc, multilabel.f1))
```

```
## multilabel.subset01  multilabel.hamloss      multilabel.acc
##           0.1201550           0.0629845           0.8798450
##       multilabel.f1
##           0.8798450
```

```
getMultilabelBinaryPerformances(prediction4, measures = list(acc, mmce, auc))
```

```
##            acc.test.mean mmce.test.mean auc.test.mean
## Hinselmann     0.9534884     0.04651163     0.5000000
## Schiller       0.9108527     0.08914729     0.5000000
## Citology       0.9457364     0.05426357     0.5837237
## Biopsy         0.9379845     0.06201550     0.5000000
```

Method 5 - stacking

```
stack.learner <- makeMultilabelStackingWrapper(basic.learner)
model5 <- mlr::train(stack.learner, task =  traintask, subset = train.set)
```

```
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
```

```
prediction5 <- predict(model5,newdata = test)
performance(prediction5,measures = list(multilabel.subset01, multilabel.hamlo
ss, multilabel.acc, multilabel.f1))
```

```
## multilabel.subset01  multilabel.hamloss      multilabel.acc
##           0.1201550           0.0629845           0.8798450
##       multilabel.f1
##           0.8798450
```

```
getMultilabelBinaryPerformances(prediction5, measures = list(acc, mmce, auc))
```

```
##            acc.test.mean mmce.test.mean auc.test.mean
## Hinselmann     0.9534884     0.04651163     0.6080623
## Schiller       0.9108527     0.08914729     0.6958372
## Citology       0.9457364     0.05426357     0.6150468
## Biopsy         0.9379845     0.06201550     0.7678202
```

Method 6 - Nested Stacking

```
nest.learner <- makeMultilabelNestedStackingWrapper(basic.learner)
model6 <- mlr::train(nest.learner, task =  traintask, subset = train.set)

## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...
## Distribution not specified, assuming bernoulli ...

prediction6 <- predict(model6,newdata = test)
performance(prediction6,measures = list(multilabel.subset01, multilabel.hamlo
ss, multilabel.acc, multilabel.f1))

## multilabel.subset01  multilabel.hamloss      multilabel.acc
##           0.1201550           0.0629845           0.8798450
##       multilabel.f1
##           0.8798450

getMultilabelBinaryPerformances(prediction6, measures = list(acc, mmce, auc))

##             acc.test.mean mmce.test.mean auc.test.mean
## Hinselmann     0.9534884     0.04651163     0.6332995
## Schiller       0.9108527     0.08914729     0.7187789
## Citology       0.9457364     0.05426357     0.5862119
## Biopsy         0.9379845     0.06201550     0.7798295
```

결과

- problem transformation models 와 hypertuned multilabel random forest 본질적으로 거의 비슷하다.
- 종속변수의 가정은 도메인 지식을 활용해서 악용될 수 있다.
- 종속변수 순서는 타켓변수 다음에 오는 종속변수 순서와 똑같이 중요한다.