

Predição

José Elvano Moraes

25/maio/2021

dados contendo NA's

```
## Rows: 543,745
## Columns: 12
## $ CLASSI_FIN <dbl> 4, 5, 4, 4, 4, 4, 5, 5, 5, 5, 4, NA, 5, 5, 5, 4, 5, 4, 5, 4,~
## $ TP_IDADE <dbl> 2, 3, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2,~
## $ SG_UF_NOT <chr> "SP", "BA", "SP", "SP", "MG", "PR", "SP", "SP", "SP", "SP",~
## $ NU_IDADE_N <dbl> 3, 46, 65, 84, 10, 16, 70, 55, 81, 44, 3, 2, 58, 51, 40, 15,~
## $ SATURACAO <dbl> 2, 2, 1, 1, 2, 1, NA, 2, 2, 1, 1, 2, 1, 1, NA, 2, 2, 2, 1, ~
## $ EVOLUCAO <dbl> 1, 1, NA, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 2, NA, NA, 1, 1, 1,~
## $ DIABETES <dbl> NA, 1, NA, 1, NA, 2, 1, 1, 2, NA, 2, NA, 1, NA, NA, 2, NA, ~
## $ OBESIDADE <dbl> NA, 2, NA, 2, NA, 2, NA, 2, 2, NA, 2, NA, 2, NA, NA, 2, NA,~
## $ UTI <dbl> 2, 1, NA, 2, 2, 2, 2, 1, 2, 1, 1, 2, 2, 1, NA, 2, 2, 1, 1, ~
## $ CARDIOPATI <dbl> NA, 1, NA, 1, NA, 2, NA, 1, 1, NA, 2, NA, 1, NA, NA, 2, NA,~
## $ SUPORT_VEN <dbl> 3, NA, NA, 2, 3, 2, 2, 9, 3, 2, 2, 3, 2, 1, 3, 3, 3, 3, 2, ~
## $ OUTRO_SIN <dbl> 2, 2, NA, 1, 2, 1, 1, 2, 1, NA, 2, 1, 1, NA, 1, 2, 2, 1, 1,~
```

Dados com NA's omitidos

```
dd.without.na <- na.omit(dd.with.na)
glimpse(dd.without.na)
```

```
## Rows: 112,796
## Columns: 12
## $ CLASSI_FIN <dbl> 4, 4, 5, 5, 4, 5, 4, 5, 5, 5, 5, 4, 5, 5, 5, 5, 4, 5, 5, 4,~
## $ TP_IDADE <dbl> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,~
## $ SG_UF_NOT <chr> "SP", "PR", "SP", "SP", "MG", "SP", "BA", "PR", "RJ", "SC",~
## $ NU_IDADE_N <dbl> 84, 16, 55, 81, 3, 58, 68, 57, 79, 66, 89, 39, 71, 80, 43, ~
## $ SATURACAO <dbl> 1, 1, 2, 2, 1, 1, 2, 1, 2, 2, 1, 2, 1, 2, 1, 1, 1, 1, 2, 1,~
## $ EVOLUCAO <dbl> 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 2, 2, 2, 2, 2,~
## $ DIABETES <dbl> 1, 2, 1, 2, 2, 1, 1, 1, 2, 1, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2,~
## $ OBESIDADE <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2,~
## $ UTI <dbl> 2, 2, 1, 2, 1, 2, 1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 1, 1, 2,~
## $ CARDIOPATI <dbl> 1, 2, 1, 1, 2, 1, 1, 2, 2, 1, 1, 2, 2, 1, 1, 1, 1, 1, 2, 1,~
## $ SUPORT_VEN <dbl> 2, 2, 9, 3, 2, 2, 3, 2, 2, 2, 2, 3, 2, 9, 2, 1, 1, 2, 2, 1,~
## $ OUTRO_SIN <dbl> 1, 1, 2, 1, 2, 1, 1, 1, 1, 2, 2, 2, 1, 2, 2, 2, 2, 1, 1, 1,~
```

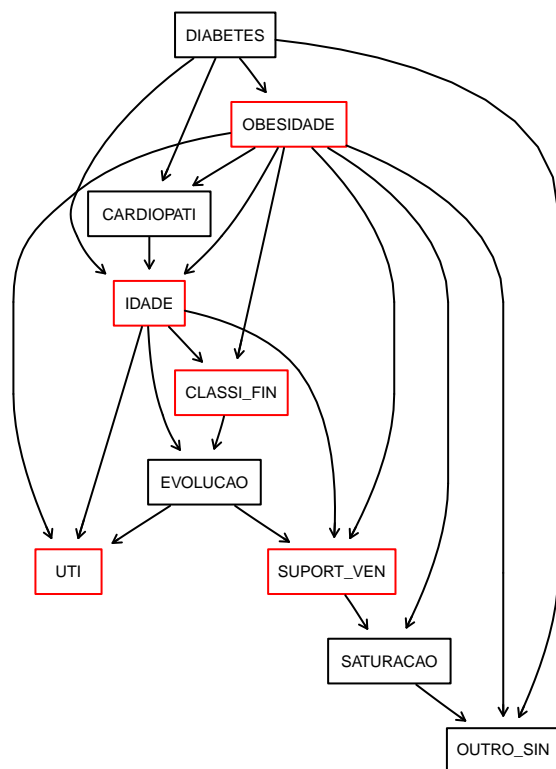
impute data

```
## Rows: 112,174
## Columns: 11
## $ CLASSI_FIN <fct> 4, 4, 5, 5, 4, 5, 4, 5, 5, 5, 5, 4, 5, 5, 5, 5, 4, 5, 5, 4,~
## $ SG_UF_NOT <fct> SP, PR, SP, SP, MG, SP, BA, PR, RJ, SC, SP, SP, SP, SP, SP,~
## $ IDADE <fct> "(76.3333,114]", "[1,38.6667]", "(38.6667,76.3333]", "(76.3~
## $ SATURACAO <fct> 1, 1, 2, 2, 1, 1, 2, 1, 2, 2, 1, 2, 1, 2, 1, 1, 1, 1, 2, 1,~
## $ EVOLUCAO <fct> 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 2, 2, 2, 2, 2,~
## $ DIABETES <fct> 1, 2, 1, 2, 2, 1, 1, 1, 2, 1, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2,~
## $ OBESIDADE <fct> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2,~
## $ UTI <fct> 2, 2, 1, 2, 1, 2, 1, 1, 2, 2, 2, 2, 2, 1, 2, 1, 2, 1, 1, 2,~
## $ CARDIOPATI <fct> 1, 2, 1, 1, 2, 1, 1, 2, 2, 1, 1, 2, 2, 1, 1, 1, 1, 1, 2, 1,~
## $ SUPORT_VEN <fct> 2, 2, 9, 3, 2, 2, 3, 2, 2, 2, 2, 3, 2, 9, 2, 1, 1, 2, 2, 1,~
## $ OUTRO_SIN <fct> 1, 1, 2, 1, 2, 1, 1, 1, 1, 2, 2, 2, 1, 2, 2, 2, 2, 1, 1, 1,~
```

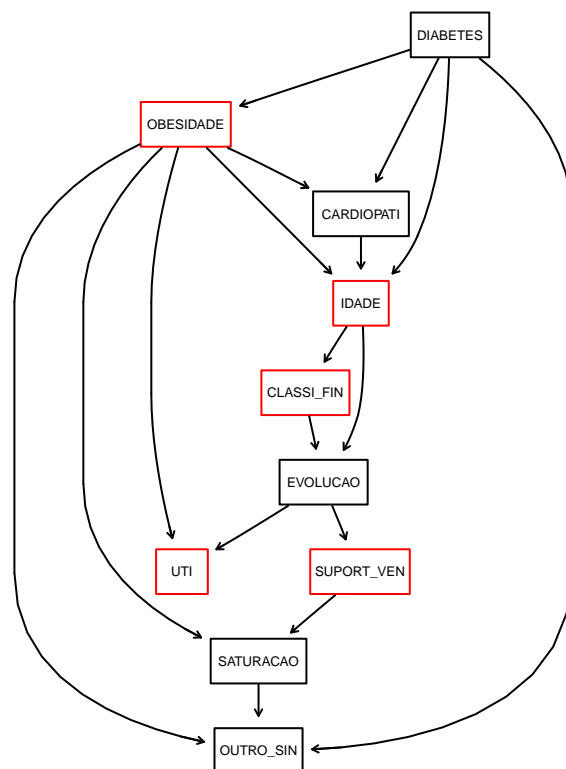
```
##
##      L      T
## 107174  5000
```

```
train <- s1
test  <- s2
```

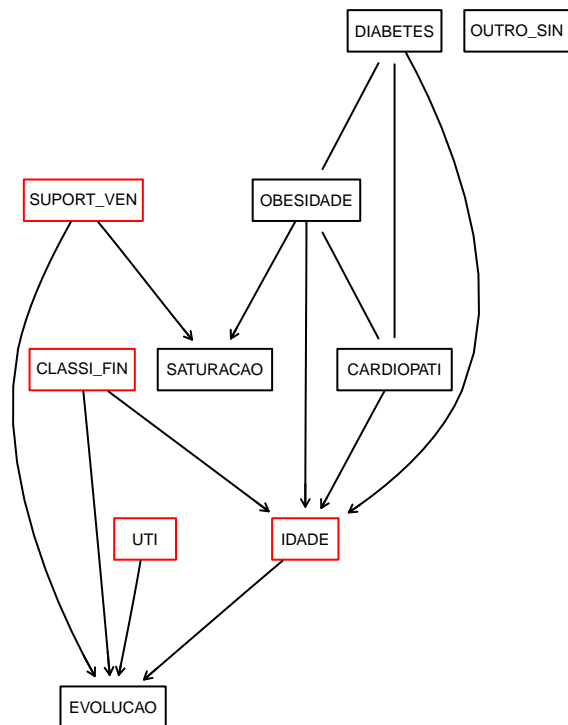
Algo = HC



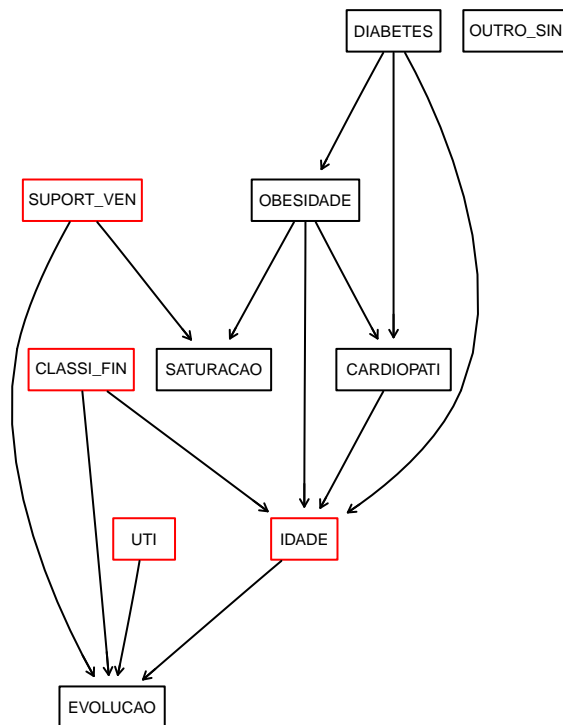
Algo = MMHC



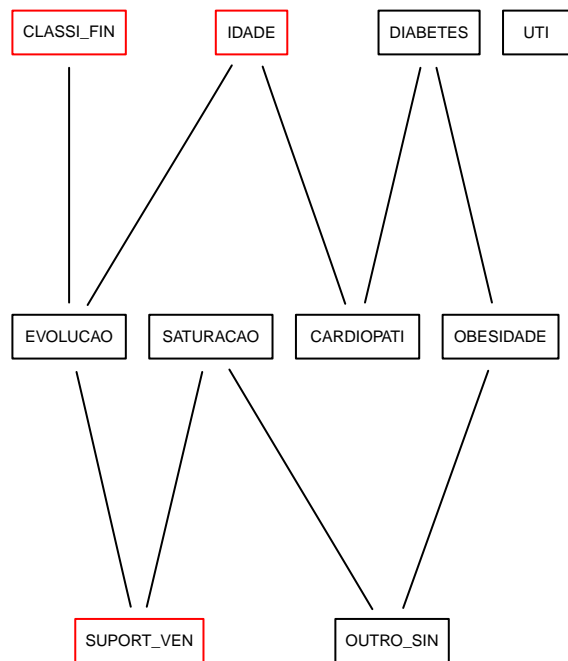
Algo = GS



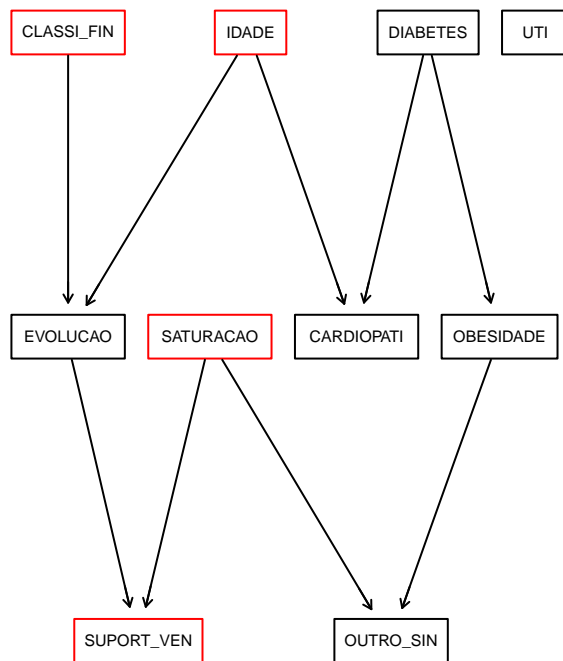
Algo = bn.gs.dagged



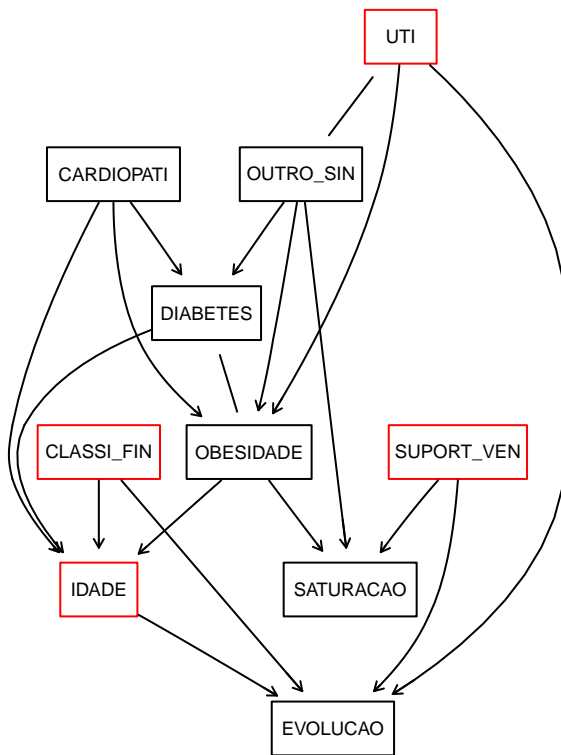
Algo = ARACNE



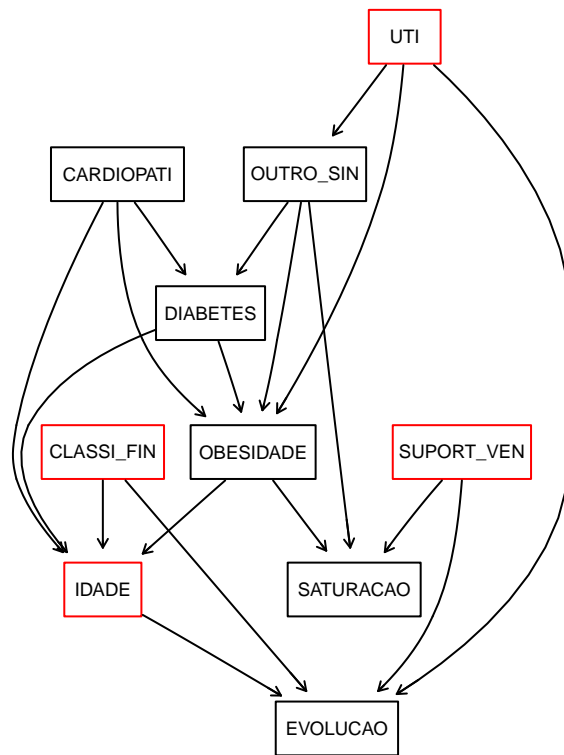
Algo = bn.aracne.dagged



Algo = bn.iamb



Algo = bn.iamb.dagged



Predição

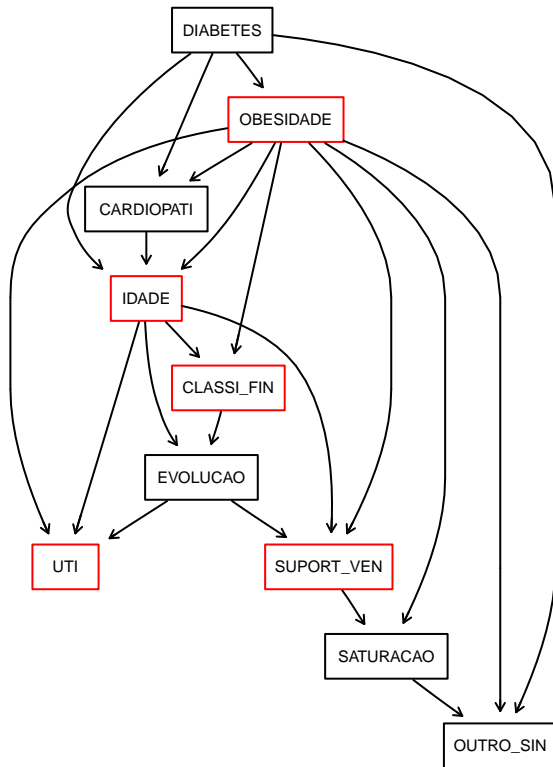
Bootstrap REDE PADRÃO

Tempo para 10 Bootstraps: 14.6458420753479 s

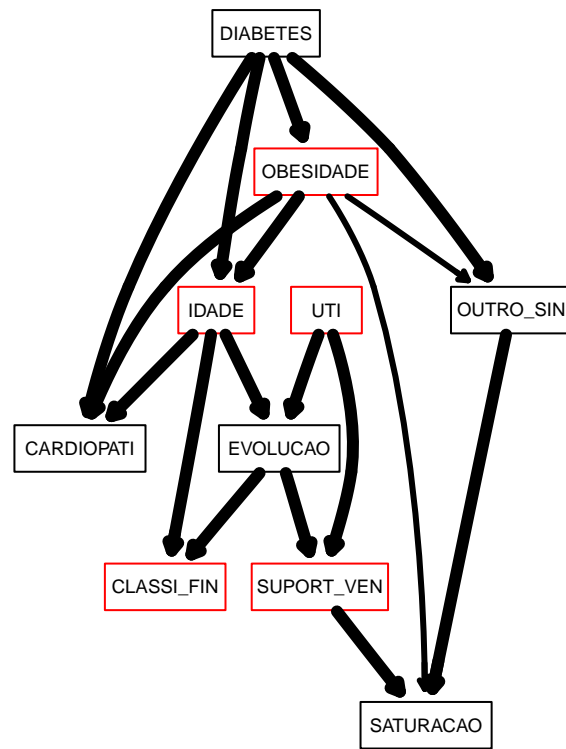
Rede “dagged”

```
rede.media.train.dagged <- pdag2dag(rede.media.train,
                                     ordering = names(train))
```

HC Thr: 0.5



HC



SET.EVIDENCE + QUERYGRAIN

```
#
# [1] "CLASSI_FIN" "IDADE" "OBESIDADE" "UTI" "SUPORT_VEN"
#
# IDADE
# [1,36] (36,71] (71,106]
#
# UTI
# 1-Sim
# 2-Nao
# 9-Ignorado
#
#SUPORT_VEN
# 1-Sim, invasivo
# 2-Sim, nao invasivo
# 3-Nao
# 9-Ignorado
#
prediga <- function(rede, amostras, estados){

  fitt1 <- bn.fit(rede, amostras)
  junction = compile(as.grain(fitt1))
  jedu = setEvidence(propagate = TRUE, junction,
                    node = c("OBESIDADE",
```

```

        "CLASSI_FIN",
        "IDADE",
        "UTI",
        "SUPORT_VEN"),
      states = estados)
suppressWarnings(pred.rede <- querygrain(jedu,
  nodes = c("EVOLUCAO"),
  type = "marginal"))
return(pred.rede)
}

estados <- c("1", "5", "(38.6667,76.3333]", "1", "1")

pred.rede <- suppressWarnings(prediga(bn.hc, train, estados))

print(pred.rede)

## $EVOLUCAO
## EVOLUCAO
##           1           2           3           9
## 0.062881905 0.923699494 0.003294884 0.010123717

```

predição por proporção dos dados

```

# usando dataset de test
x <- filter(test,
  OBESIDADE == 1,
  CLASSI_FIN == 5,
  IDADE == "(38.6667,76.3333]",
  UTI == 1,
  SUPORT_VEN == 1)
pred.proporcao1 <- nrow(filter(x, EVOLUCAO == 1))/nrow(x)

x2 <- filter(test,
  OBESIDADE == 1,
  CLASSI_FIN == 5,
  IDADE == "(38.6667,76.3333]",
  UTI == 1,
  SUPORT_VEN == 1)
pred.proporcao2 <- nrow(filter(x2, EVOLUCAO == 2))/nrow(x2)

x3 <- filter(test,
  OBESIDADE == 1,
  CLASSI_FIN == 5,
  IDADE == "(38.6667,76.3333]",
  UTI == 1,
  SUPORT_VEN == 1)
pred.proporcao3 <- nrow(filter(x2, EVOLUCAO == 3))/nrow(x3)

print(pred.proporcao1)

```

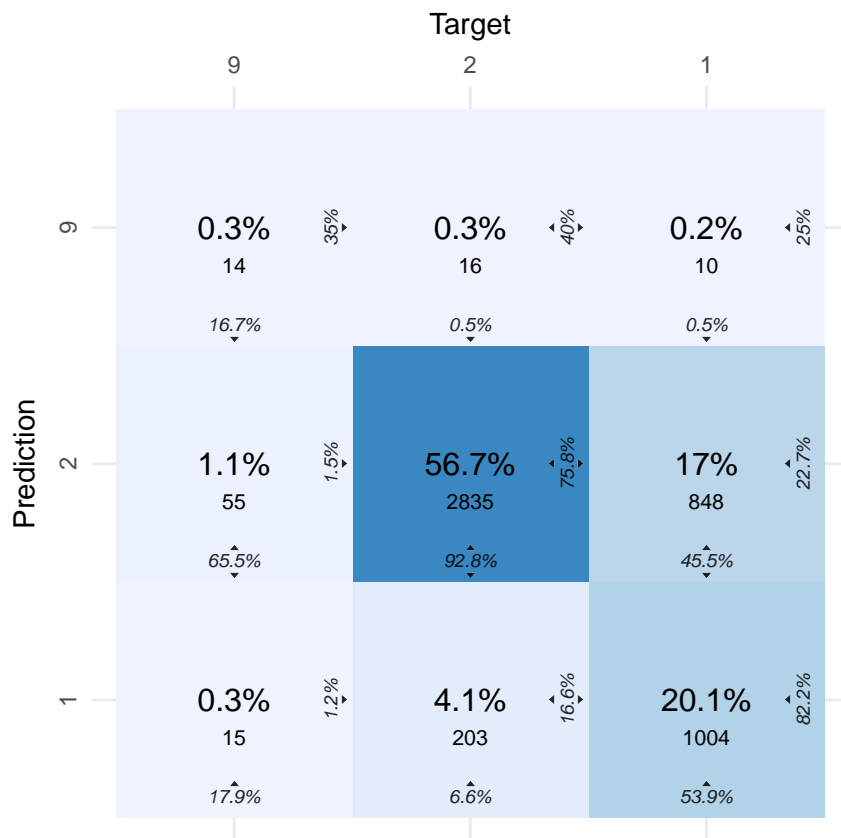
```
## [1] 0.09375
print(pred.proporcao2)

## [1] 0.875
print(pred.proporcao3)

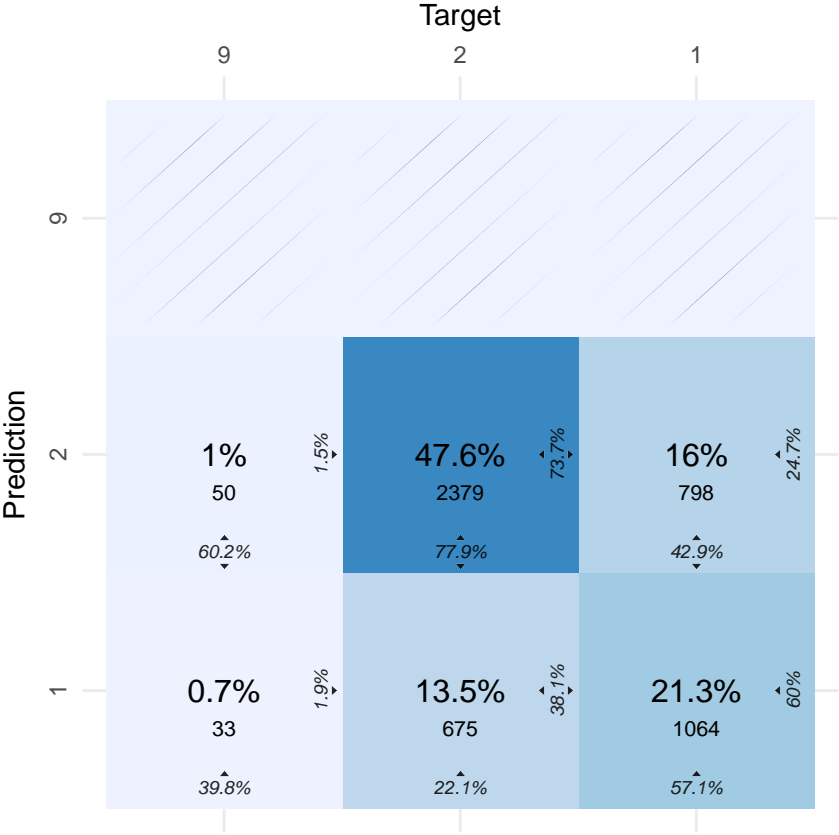
## [1] 0
```

Fit dag to data and predict the value of latent variable

predicting a variable in the test set. UTI rede simplificada



predicting a variable in the test set. UTI rede completa



predicting a variable in the test set. EVOLUCAO completa

