

Predição. Dados de 22 abril 2021 - heaartmink de 3

Efeito do tempo de internação até entrada na UTI versus desfecho

José Elvano Moraes

02/junho/2021

SRAG 12/04/2021

URL: <https://s3-sa-east-1.amazonaws.com/ckan.saude.gov.br/SRAG/2021/INFLUD21-12-04-2021.csv>

Do resumo do conjunto de dados Vigilância de Síndrome Respiratória Aguda Grave (SRAG) O Ministério da Saúde (MS), por meio da Secretaria de Vigilância em Saúde (SVS), desenvolve a vigilância da Síndrome Respiratória... Fonte: SRAG 2021 - Banco de Dados de Síndrome Respiratória Aguda Grave - incluindo dados da COVID-19

Dicionário de Dados

URL: <https://opendatasus.saude.gov.br/dataset/9f76e80f-a2f1-4662-9e37-71084eae23e3/resource/b3321e55-24e9-49ab-8651-29cf5c8f3179/download/dicionario-de-dados-srag-hospitalizado-27.07.2020-final.pdf>

Do resumo do conjunto de dados

Vigilância de Síndrome Respiratória Aguda Grave (SRAG) O Ministério da Saúde (MS), por meio da Secretaria de Vigilância em Saúde (SVS), desenvolve a vigilância da Síndrome Respiratória... Fonte: SRAG 2021 - Banco de Dados de Síndrome Respiratória Aguda Grave - incluindo dados da COVID-19

Variáveis disponíveis e números de valores faltantes para cada uma

```
## DT_NOTIFIC      0
## SEM_NOT         0
## DT_SIN_PRI      0
## SEM_PRI         0
## SG_UF_NOT       0
## ID_REGIONA      51139
## CO_REGIONA      51139
## ID_MUNICIP      0
## CO_MUN_NOT      0
## ID_UNIDADE      0
## CO_UNI_NOT      0
## CS_SEXO         0
## DT_NASC         467
## NU_IDADE_N      0
```

```

## TP_IDADE      0
## COD_IDADE     5
## CS_GESTANT    0
## CS_RACA       0
## CS_ETINIA     543745
## CS_ESCOL_N    195124
## ID_PAIS       0
## CO_PAIS       0
## SG_UF         67
## ID_RG_RESI    49762
## CO_RG_RESI    49762
## ID_MN_RESI    67
## CO_MUN_RES    67
## CS_ZONA       65852
## SURTO_SG      303974
## NOSOCOMIAL    131465
## AVE_SUINO     122475
## FEBRE         94804
## TOSSE         76941
## GARGANTA      167574
## DISPNEIA      69599
## DESC_RESP     104682
## SATURACAO     88992
## DIARREIA      171927
## VOMITO        178962
## OUTRO_SIN     170807
## OUTRO_DES     379866
## PUERPERA      351948
## FATOR_RISC     0
## CARDIOPATI    290068
## HEMATOLOGI    351142
## SIND_DOWN     351721
## HEPATICA      351612
## ASMA          348156
## DIABETES      308477
## NEUROLOGIC    346821
## PNEUMOPATI    346942
## IMUNODEPRE    349463
## RENAL         348253
## OBESIDADE     340105
## OBES_IMC      537264
## OUT_MORBI     310984
## MORB_DESC     413692
## VACINA        144773
## DT_UT_DOSE    515312
## MAE_VAC       539340
## DT_VAC_MAE    543745
## M_AMAMENTA    540399
## DT_DOSEUNI    543745
## DT_1_DOSE     543745
## DT_2_DOSE     543745
## ANTIVIRAL     99762
## TP_ANTIVIR    528180
## OUT_ANTIV     541639

```

## DT_ANTIVIR	530185
## HOSPITAL	19704
## DT_INTERNA	42426
## SG_UF_INTE	2
## ID_RG_INTE	121033
## CO_RG_INTE	121033
## ID_MN_INTE	31118
## CO_MU_INTE	31118
## UTI	83155
## DT_ENTUTI	402818
## DT_SAIDUTI	480078
## SUPORT_VEN	81011
## RAIIX_RES	244069
## RAIIX_OUT	528680
## DT_RAIIX	448344
## AMOSTRA	38895
## DT_COLETA	68379
## TP_AMOSTRA	77497
## OUT_AMOST	496533
## PCR_RESUL	75631
## DT_PCR	215118
## POS_PCRFLU	409681
## TP_FLU_PCR	543721
## PCR_FLUASU	543736
## FLUASU_OUT	543745
## PCR_FLUBLI	543745
## FLUBLI_OUT	543745
## POS_PCROUT	320869
## PCR_VSR	541871
## PCR_PARA1	543709
## PCR_PARA2	543726
## PCR_PARA3	543731
## PCR_PARA4	543737
## PCR_ADENO	543670
## PCR_METAP	543744
## PCR_BOCA	543729
## PCR_RINO	543073
## PCR_OUTRO	543498
## DS_PCR_OUT	543745
## CLASSI_FIN	103974
## CLASSI_OUT	543233
## CRITERIO	132247
## EVOLUCAO	193581
## DT_EVOLUCA	223453
## DT_ENCERRA	193851
## DT_DIGITA	0
## HISTO_VGM	0
## PAIS_VGM	543745
## CO_PS_VGM	543690
## LO_PS_VGM	543745
## DT_VGM	543745
## DT_RT_VGM	543745
## PCR_SARS2	324649
## PAC_COCBO	526233

```

## PAC_DSCBO      526233
## OUT_ANIM       543745
## DOR_ABD       187846
## FADIGA        166485
## PERD_OLFT     181929
## PERD_PALA     182275
## TOMO_RES      208995
## TOMO_OUT      530868
## DT_TOMO       345333
## TP_TES_AN     474946
## DT_RES_AN     475975
## RES_AN        94744
## POS_AN_FLU    509408
## TP_FLU_AN     543700
## POS_AN_OUT    495760
## AN_SARS2      496986
## AN_VSR        543265
## AN_PARA1      543742
## AN_PARA2      543743
## AN_PARA3      543744
## AN_ADENO      543744
## AN_OUTRO      543591
## DS_AN_OUT     543745
## TP_AM_SOR     497310
## SOR_OUT       538877
## DT_CO_SOR     503913
## TP_SOR        498300
## OUT_SOR       540607
## DT_RES        497374
## RES_IGG       480084
## RES_IGM       478770
## RES_IGA       500128

```

variáveis usadas

dados contendo NA's

```

## Rows: 543,745
## Columns: 14
## $ CLASSI_FIN <dbl> 4, 5, 4, 4, 4, 4, 5, 5, 5, 5, 4, NA, 5, 5, 5, 4, 5, 4, 5, 4~
## $ DT_SIN_PRI <chr> "04/01/2021", "04/01/2021", "03/01/2021", "03/01/2021", "05~
## $ DT_EVOLUCA <chr> "04/01/2021", NA, NA, "05/01/2021", "13/01/2021", "12/01/20~
## $ EVOLUCAO   <dbl> 1, 1, NA, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 2, NA, NA, 1, 1, 1, ~
## $ TP_IDADE   <dbl> 2, 3, 3, 3, 2, 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, ~
## $ 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, ~
## $ DT_ENTUTI  <chr> NA, "04/01/2021", NA, NA, NA, NA, NA, "09/01/2021", NA, "11~
## $ 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, ~

```

```
## Rows: 42,924
## Columns: 14
## $ CLASSI_FIN <dbl> 5, 4, 4, 5, 5, 5, 5, 5, 5, 5, 4, 5, 4, 5, 4, 4, 5, 5, 5, 5,~
## $ DT_SIN_PRI <chr> "05/01/2021", "10/01/2021", "05/01/2021", "03/01/2021", "05~
## $ DT_EVOLUCA <chr> "21/01/2021", "15/02/2021", "25/01/2021", "22/01/2021", "29~
## $ EVOLUCAO <dbl> 1, 1, 1, 1, 1, 2, 2, 2, 2, 1, 2, 2, 1, 1, 2, 1, 2, 1, 2, 2,~
## $ 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", "MG", "BA", "PR", "SP", "GO", "SP", "SP", "SP", "SP", "BA",~
## $ NU_IDADE_N <dbl> 55, 3, 68, 57, 80, 43, 88, 57, 73, 53, 71, 81, 62, 52, 77, ~
## $ SATURACAO <dbl> 2, 1, 2, 1, 2, 1, 1, 2, 2, 2, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1,~
## $ DIABETES <dbl> 1, 2, 1, 1, 1, 1, 2, 2, 2, 2, 2, 1, 2, 2, 2, 1, 1, 1, 1, 1,~
## $ OBESIDADE <dbl> 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2,~
## $ UTI <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ DT_ENTUTI <chr> "09/01/2021", "11/01/2021", "09/01/2021", "09/01/2021", "11~
## $ CARDIOPATI <dbl> 1, 2, 1, 2, 1, 1, 1, 2, 2, 1, 1, 2, 1, 2, 1, 1, 2, 1, 1, 1,~
## $ SUPORT_VEN <dbl> 9, 2, 3, 2, 9, 1, 2, 2, 1, 2, 1, 1, 1, 2, 2, 2, 2, 2, 1, 2,~
```

Dados com NA's omitidos

```
## Rows: 42,924
## Columns: 14
## $ CLASSI_FIN <dbl> 5, 4, 4, 5, 5, 5, 5, 5, 5, 5, 4, 5, 4, 5, 4, 4, 5, 5, 5, 5,~
## $ DT_SIN_PRI <dtm> 2021-01-05, 2021-01-10, 2021-01-05, 2021-01-03, 2021-01-05~
## $ DT_EVOLUCA <dtm> 2021-01-21, 2021-02-15, 2021-01-25, 2021-01-22, 2021-01-29~
## $ EVOLUCAO <dbl> 1, 1, 1, 1, 1, 2, 2, 2, 2, 1, 2, 2, 1, 1, 2, 1, 2, 1, 2, 2,~
## $ 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", "MG", "BA", "PR", "SP", "GO", "SP", "SP", "SP", "SP", "BA",~
## $ NU_IDADE_N <dbl> 55, 3, 68, 57, 80, 43, 88, 57, 73, 53, 71, 81, 62, 52, 77, ~
## $ SATURACAO <dbl> 2, 1, 2, 1, 2, 1, 1, 2, 2, 2, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1,~
## $ DIABETES <dbl> 1, 2, 1, 1, 1, 1, 2, 2, 2, 2, 2, 1, 2, 2, 2, 1, 1, 1, 1, 1,~
## $ OBESIDADE <dbl> 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2,~
## $ UTI <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ DT_ENTUTI <dtm> 2021-01-09, 2021-01-11, 2021-01-09, 2021-01-09, 2021-01-11~
## $ CARDIOPATI <dbl> 1, 2, 1, 2, 1, 1, 1, 2, 2, 1, 1, 2, 1, 2, 1, 1, 2, 1, 1, 1,~
## $ SUPORT_VEN <dbl> 9, 2, 3, 2, 9, 1, 2, 2, 1, 2, 1, 1, 1, 2, 2, 2, 2, 2, 1, 2,~
```

cria colunas com intervalos de tempo

```
## Rows: 42,656
## Columns: 12
## $ CLASSI_FIN <fct> 5, 4, 4, 5, 5, 5, 5, 5, 5, 5, 4, 5, 4, 5, 4, 4, 5, 5,~
## $ EVOLUCAO <fct> 1, 1, 1, 1, 1, 2, 2, 2, 2, 1, 2, 2, 1, 1, 2, 1, 2, 1,~
## $ SG_UF_NOT <fct> SP, MG, BA, PR, SP, GO, SP, SP, SP, SP, BA, SP, RS, RS, P~
## $ IDADE <fct> "(16,80]", "[1,16]", "(16,80]", "(16,80]", "(16,80]",~
## $ SATURACAO <fct> 2, 1, 2, 1, 2, 1, 1, 2, 2, 2, 2, 1, 2, 1, 1, 1, 1, 1,~
## $ DIABETES <fct> 1, 2, 1, 1, 1, 1, 2, 2, 2, 2, 2, 1, 2, 2, 2, 1, 1, 1,~
## $ OBESIDADE <fct> 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2,~
## $ UTI <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ CARDIOPATI <fct> 1, 2, 1, 2, 1, 1, 1, 2, 2, 1, 1, 2, 1, 2, 1, 1, 2, 1,~
## $ SUPORT_VEN <fct> 9, 2, 3, 2, 9, 1, 2, 2, 1, 2, 1, 1, 1, 2, 2, 2, 2, 2,~
## $ INTER_P_SIN_EVOL <fct> "(13,55]", "(13,55]", "(13,55]", "(13,55]", "(13,55]"~
## $ INTER_P_SIN_UTI <fct> "(3,10]", "(-266,3]", "(3,10]", "(3,10]", "(3,10]", "~
```

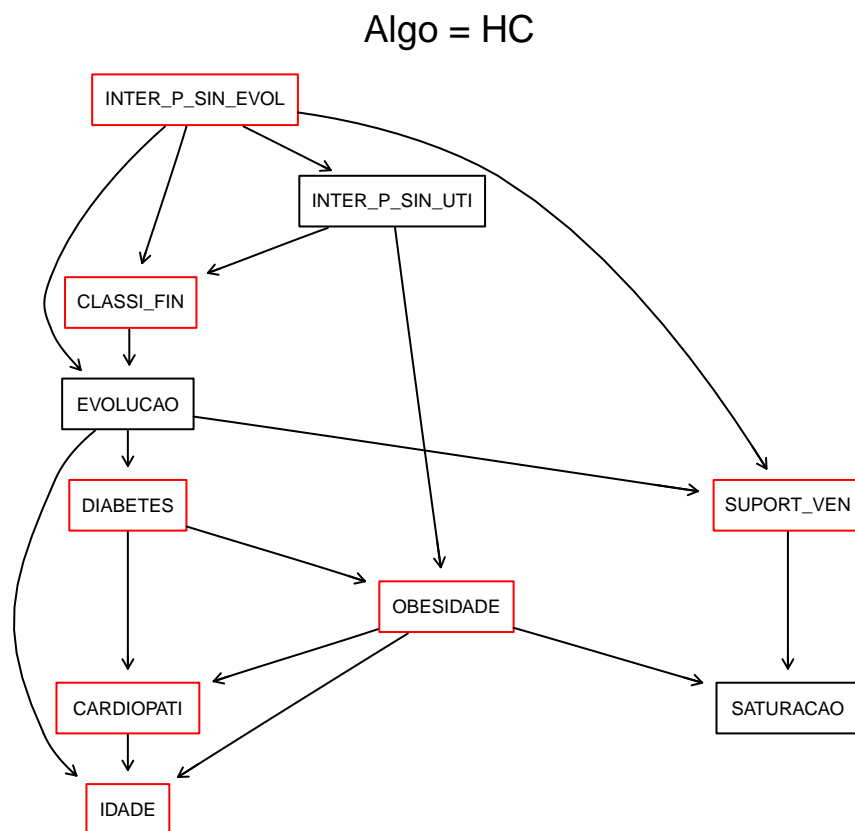
```
##  
##      L      T  
## 37656 5000  
train <- s1  
test  <- s2
```

Aprendizagem da estrutura da rede

Diferentes algoritmos de aprendizagem

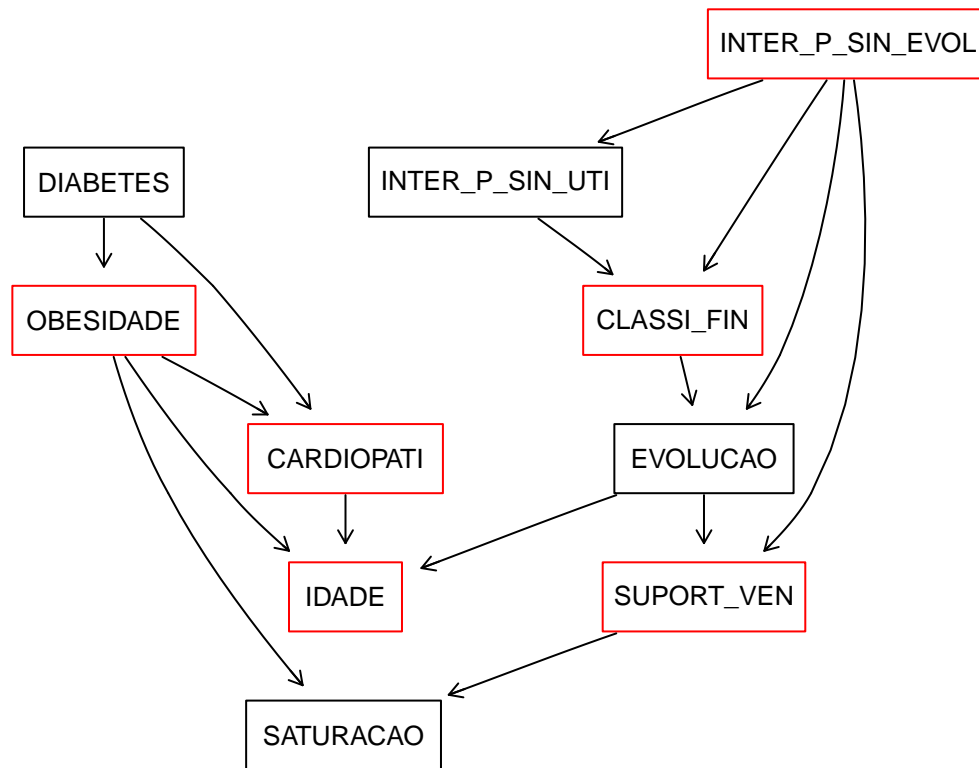
- HC
- GS
- IAMB
- MMHC

Algo = HC

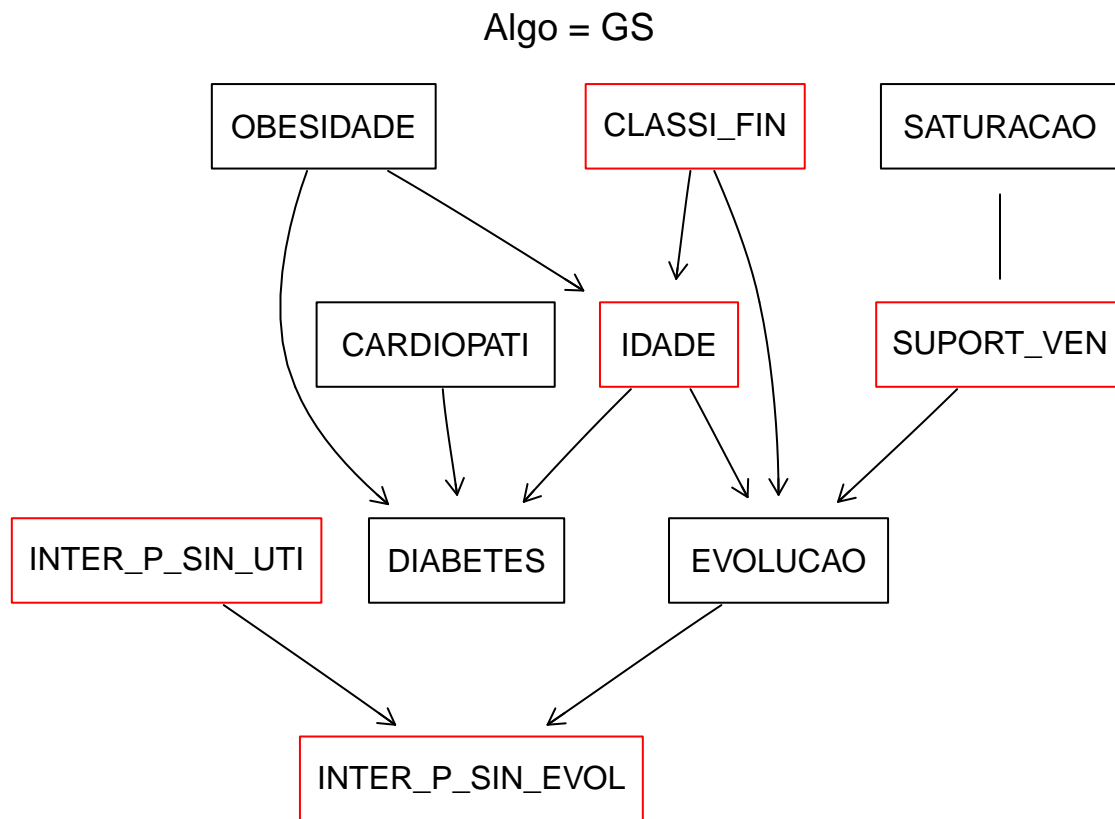


Algo = MMHC

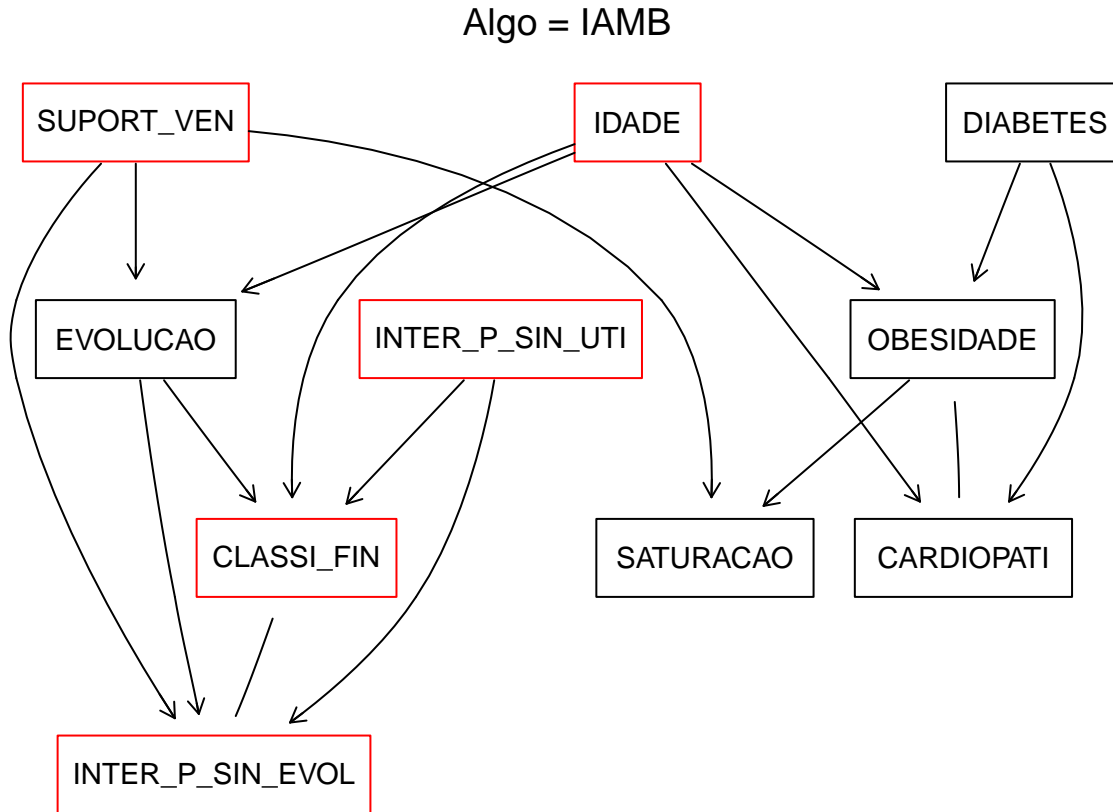
Algo = MMHC



Algo = GS



Algo = IAMB



Distribuição conjunta de probabilidades entre as variáveis.

Algoritmo *HC*

```

forca.real.hc <- suppressWarnings(boot.strength(train,
                                              R = boots.trap,
                                              algorithm = "hc"))
rede.media.train.hc = averaged.network(forca.real.hc)

## Warning in averaged.network.backend(strength = strength, nodes = nodes, : arc
## CLASSI_FIN -> EVOLUCAO would introduce cycles in the graph, ignoring.

## Warning in averaged.network.backend(strength = strength, nodes = nodes, : arc
## CLASSI_FIN -> INTER_P_SIN_EVOL would introduce cycles in the graph, ignoring.

## Warning in averaged.network.backend(strength = strength, nodes = nodes, :
## arc INTER_P_SIN_UTI -> INTER_P_SIN_EVOL would introduce cycles in the graph,
## ignoring.

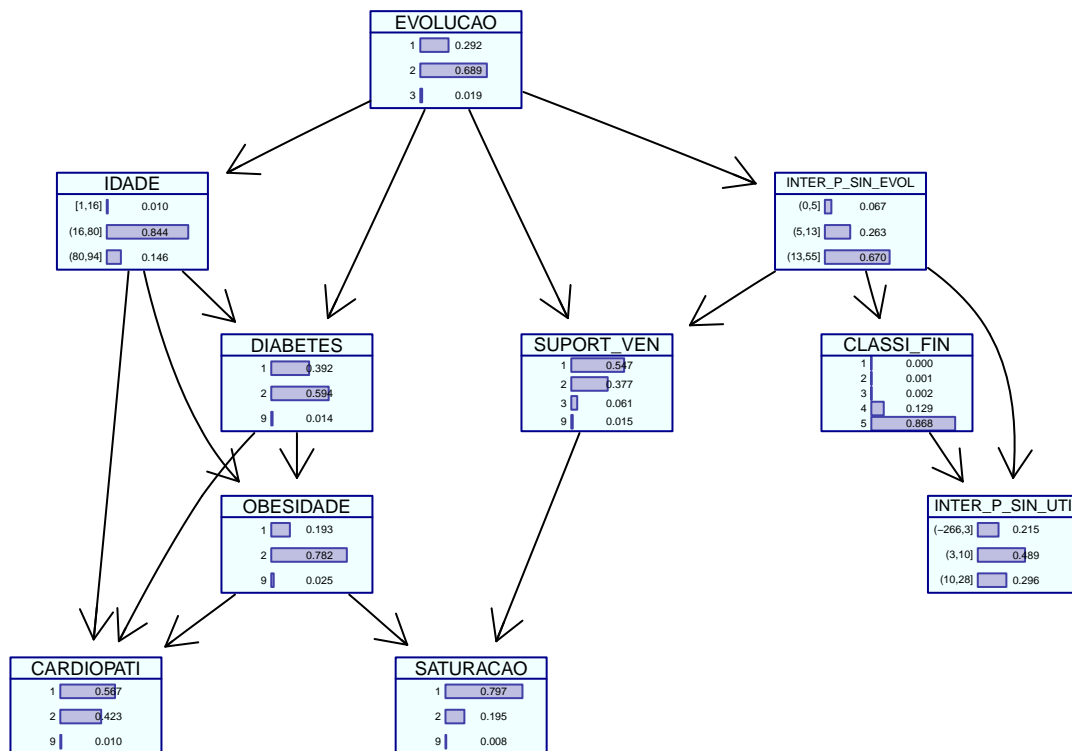
rede.media.train.dagged.hc <- pdag2dag(rede.media.train.hc,
                                      ordering = names(train))
thr.real.gs <- paste('Thr: ', attr(forca.real.hc, "threshold"))

par(mfrow = c(1, 1))
  
```

```
fit.bn.hc <- bn.fit(x = rede.media.train.dagged.hc, data = s1)
graphviz.chart(fit.bn.hc,
  type = "barprob",
  col = "darkblue",
  bg = "azure",
  bar.col = "darkblue",
  main = "Rede de probabilidades")
```

```
## Warning in from.bn.fit.to.grain(x): NaN conditional probabilities in CARDIOPATI,
## replaced with a uniform distribution.
```

Rede de probabilidades



Distribuição conjunta de probabilidades entre as variáveis.

Algoritmo GS

```
forca.real.gs <- suppressWarnings(boot.strength(train,
  R = boots.trap,
  algorithm = "gs"))
rede.media.train.gs = averaged.network(forca.real.gs)

rede.media.train.dagged.gs <- pdag2dag(rede.media.train.gs,
  ordering = names(train))
thr.real.gs <- paste('Thr: ', attr(forca.real.gs, "threshold"))

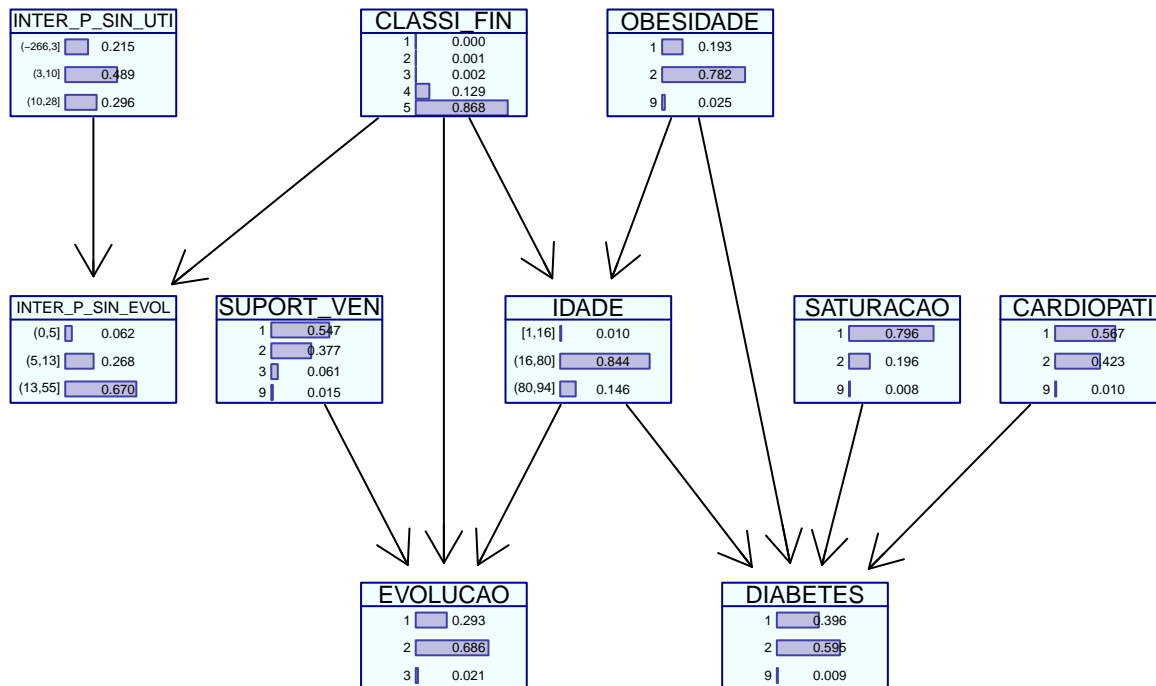
par(mfrow = c(1, 1))
```

```
fit.bn.gs <- bn.fit(x = rede.media.train.dagged.gs, data = s1)
graphviz.chart(fit.bn.gs,
  type = "barprob",
  col = "darkblue",
  bg = "azure",
  bar.col = "darkblue",
  main = "Rede de probabilidades")
```

```
## Warning in from.bn.fit.to.grain(x): NaN conditional probabilities in EVOLUCAO,
## replaced with a uniform distribution.
```

```
## Warning in from.bn.fit.to.grain(x): NaN conditional probabilities in DIABETES,
## replaced with a uniform distribution.
```

Rede de probabilidades



Distribuição conjunta de probabilidades entre as variáveis.

Algoritmo IAMB

```
forca.real.iamb <- suppressWarnings(boot.strength(train,
  R = boots.trap,
  algorithm = "iamb"))
rede.media.train.iamb = averaged.network(forca.real.iamb)

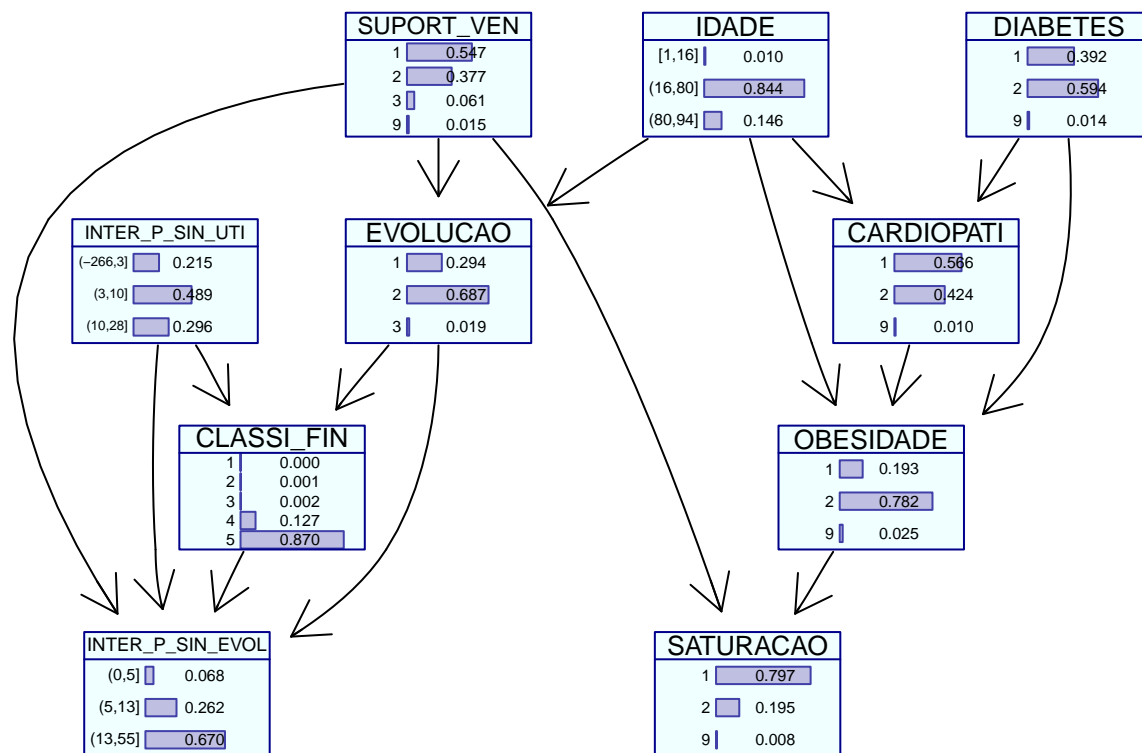
rede.media.train.dagged.iamb <- pdag2dag(rede.media.train.iamb,
  ordering = names(train))
thr.real.gs <- paste('Thr: ', attr(forca.real.iamb, "threshold"))
```

```
fit.bn.iamb <- bn.fit(x = rede.media.train.dagged.iamb, data = s1)
par(mfrow = c(1, 1))
graphviz.chart(fit.bn.iamb,
  type = "barprob",
  col = "darkblue",
  bg = "azure",
  bar.col = "darkblue",
  main = "Rede de probabilidades")
```

```
## Warning in from.bn.fit.to.grain(x): NaN conditional probabilities in OBESIDADE,
## replaced with a uniform distribution.
```

```
## Warning in from.bn.fit.to.grain(x): NaN conditional probabilities in
## INTER_P_SIN_EVOL, replaced with a uniform distribution.
```

Rede de probabilidades



Distribuição conjunta de probabilidades entre as variáveis.

Algoritmo MMHC

```
forca.real.mmhc <- suppressWarnings(boot.strength(train,
  R = boots.trap,
  algorithm = "mmhc"))
rede.media.train.mmhc = averaged.network(forca.real.mmhc)
```

```
## Warning in averaged.network.backend(strength = strength, nodes = nodes, : arc
## CLASSI_FIN -> EVOLUCAO would introduce cycles in the graph, ignoring.
```

```
## Warning in averaged.network.backend(strength = strength, nodes = nodes, : arc
## INTER_P_SIN_UTI -> CLASSI_FIN would introduce cycles in the graph, ignoring.

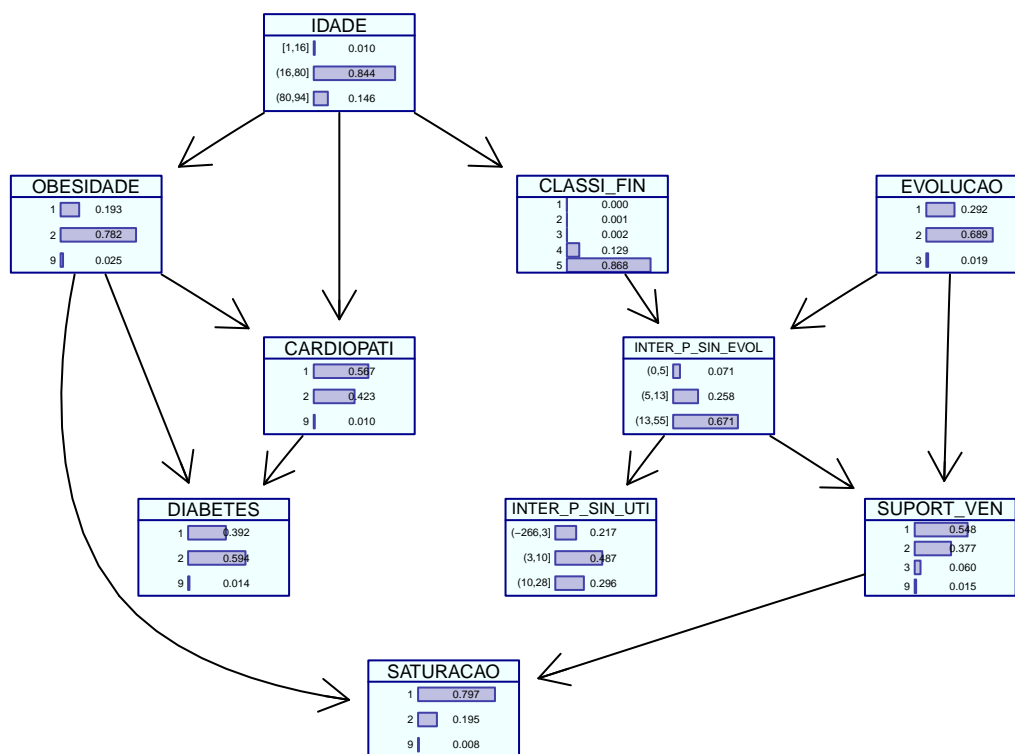
## Warning in averaged.network.backend(strength = strength, nodes = nodes, : arc
## EVOLUCAO -> IDADE would introduce cycles in the graph, ignoring.

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

thr.real.gs <- paste('Thr: ', attr(forca.real.mmhc, "threshold"))
fit.bn.mmhc <- bn.fit(x = rede.media.train.dagged.mmhc, data = s1)
par(mfrow = c(1, 1))
graphviz.chart(fit.bn.mmhc,
               type = "barprob",
               col = "darkblue",
               bg = "azure",
               bar.col = "darkblue",
               main = "Rede de probabilidades")

## Warning in from.bn.fit.to.grain(x): NaN conditional probabilities in
## INTER_P_SIN_EVOL, replaced with a uniform distribution.
```

Rede de probabilidades



Predição

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

  return(pred.rede)
}

estados <- c("5", "(71, 80]", "(13,55]", "(10,28]", "1")

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

print(pred.rede)

## $EVOLUCAO
## EVOLUCAO
##           1           2           3
## 0.127324795 0.869869423 0.002805782
```

predição por proporção dos dados

```
# usando dataset de test
x <- filter(test,
  CLASSI_FIN == 5,
  IDADE == "(71, 80]",
  INTER_P_SIN_EVOL == "(13,55]",
  INTER_P_SIN_UTI == "(10,28]",
  #UTI == 1,
  SUPORT_VEN == 1)
pred.proporcao <- nrow(filter(x, EVOLUCAO == 1))/nrow(x)

x2 <- filter(test,
  CLASSI_FIN == 5,
  IDADE == "(71, 80]",
  INTER_P_SIN_EVOL == "(13,55]",
  INTER_P_SIN_UTI == "(10,28]",
  #UTI == 1,
  SUPORT_VEN == 1)
pred.proporcao2 <- nrow(filter(x2, EVOLUCAO == 2))/nrow(x2)

print(pred.proporcao)

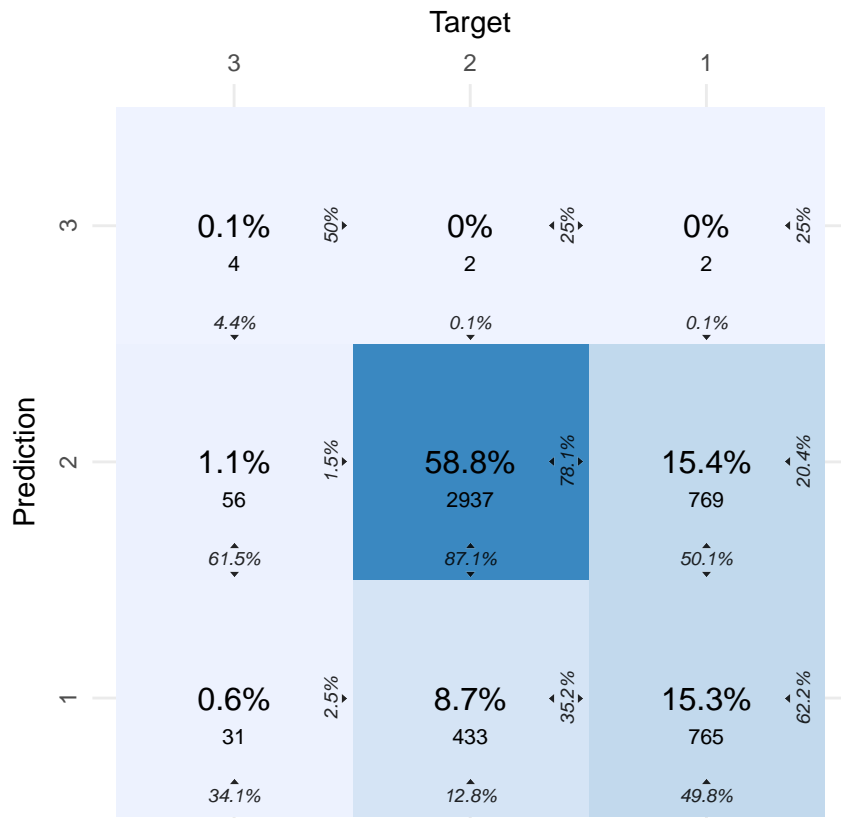
## [1] NaN
print(pred.proporcao2)

## [1] NaN
```

Fit dag to data and predict the value of latent variable

predicting a variable in the test set. EVOLUCAO

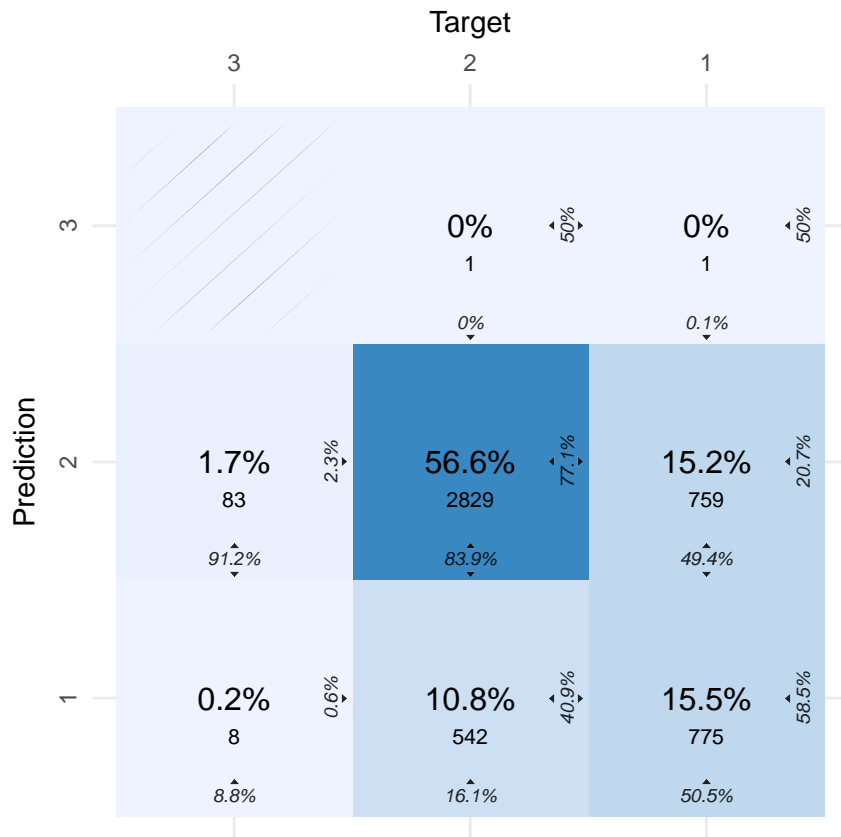
```
## Warning in map.prediction(node = node, fitted = object, data = data, n =
## extra.args$n, : dropping 14 observations because generated samples are NAs.
```



predicting a variable in the test

set. EVOLUCAO rede simplificada ## *HC

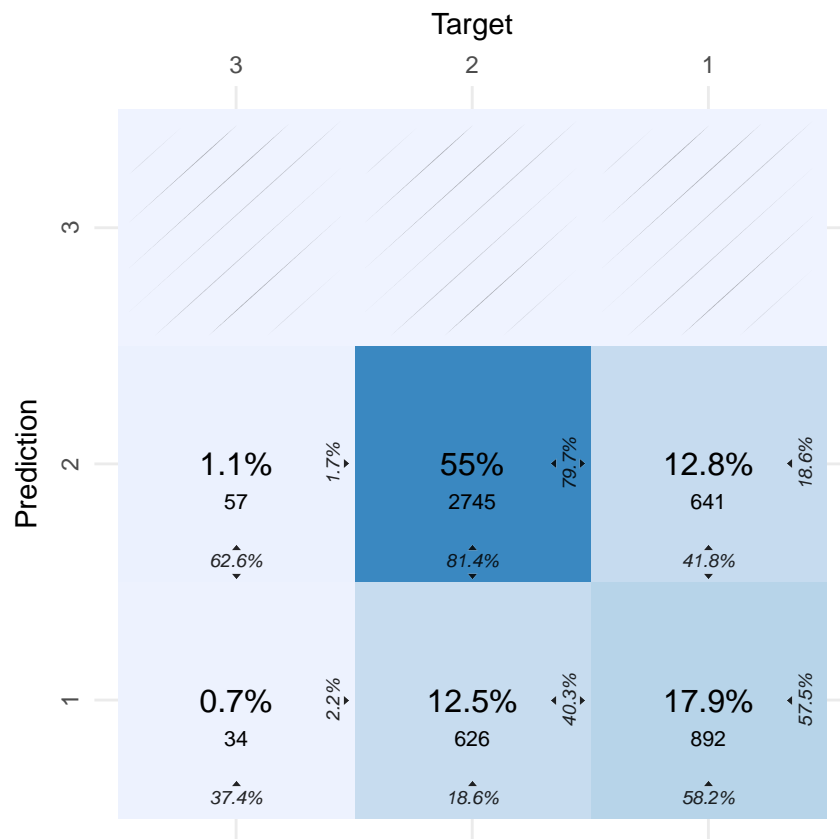
Warning in map.prediction(node = node, fitted = object, data = data, n =
extra.args\$n, : dropping 500 observations because generated samples are NAs.



predicting a variable in the test set. EVOLUCAO rede simplificada

GS

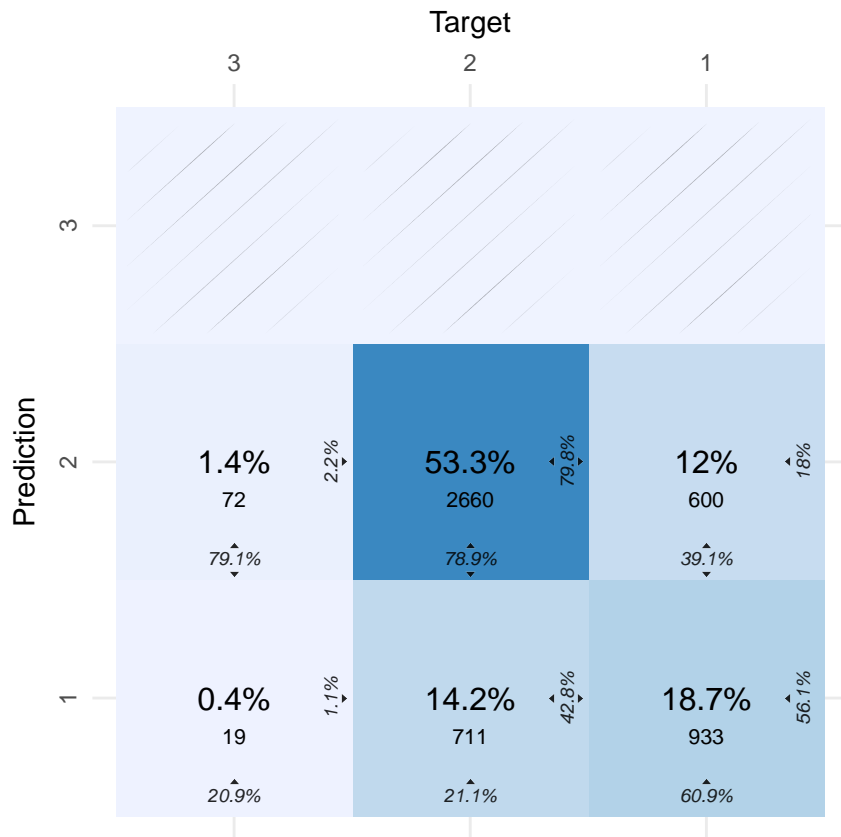
```
## Warning in map.prediction(node = node, fitted = object, data = data, n =
## extra.args$n, : dropping 1000 observations because generated samples are NAs.
```



predicting a variable in the test set. EVOLUCAO rede simplificada

MMHC

```
## Warning in map.prediction(node = node, fitted = object, data = data, n =
## extra.args$n, : dropping 23 observations because generated samples are NAs.
```



predicting a variable in the test set. EVOLUCAO rede simplificada

IAMB

```
## Warning in map.prediction(node = node, fitted = object, data = data, n =
## extra.args$n, : dropping 806 observations because generated samples are NAs.
```

