```
#
# Atividade de Aprofundamento
# Tarefa 1 = Naive Bayes e Decision Tree Credit Approval
#
# Esse código executa os modelos de Naive Bayes e Decision Tree para o
# dataframe de Credit Approval. Empregue ele para responder a questão 1 do
# questionário e faça a alteração necessária no treinamento da Decision
# Tree para responder a questão 2.
#
# NÃO É NECESSÁRIO POSTAR ESSE CÓDIGO do Credit Approval
#
# Tarefa 2 = Naive Bayes e Decision Tree Mushrooms
#
# Empregue esse cóigo como base para execução dos modelos Naive Bayes e Decision Tree para o
# dataframe mushrooms.
#
# O arquivo pode ser obtido em:
# https://www.openml.org/data/get_csv/24/dataset_24_mushroom.arff
#
# Em seguida responda as questões de 3 a 10 do questionário.
#
# ATENÇÃO: É NECESSÁRIO POSTAR ESSE CÓDIGO do mushrooms!
#



# Verifique outras libraries necessárias na sua execução
library(e1071)
library(party)
library(dummies)
```

```
mushroom =
read.csv("https://www.openml.org/data/get_csv/24/dataset_24_mushroom.arff",header=FALSE)


#Elimina linha de titulo

#print(mushroom[1,])

#V1        V2       V3        V4        V5        V6         V7         V8        V9        V10

#cap-shape cap-surface cap-color bruises%3F odor gill-attachment gill-spacing gill-size gill-color stalk-shape

#V11                V12                V13                V14              V15          V16

#stalk-root stalk-surface-above-ring stalk-surface-below-ring stalk-color-above-ring stalk-color-below-ring veil-type

#V17          V18      V19             V20       V21       V22   V23

#veil-color ring-number ring-type spore-print-color population habitat class


mushroom=mushroom[-1,]


# Elimina valores NA

mushroom[mushroom[]=="?"]=NA

mushroom=na.omit(mushroom)


table(mushroom$V1)


mushroom$class <- ifelse(mushroom$V23 == "'p'", 1, 0)


# factor determina como uma variável categórica

classe <- factor(mushroom$class)


str(classe)


# com o levels foi definido os valores nessas categorias

levels(classe) <- c("e", "p")
```

```
# Comparação de quantidade de valores antes inputar os dados na base
table(classe)
table(mushroom$class)

mushroom$class <- classe
mushroom$V23 <- NULL

summary(mushroom)

# MANTER O SEED PARA GARANTIR AS RESPOSTAS DO QUESTIONÁRIO
RNGversion("3.5.2")
set.seed(1987)

# Gera Conjuntos de Treinamento e Teste
L <- sample(1:nrow(mushroom),round(nrow(mushroom)/3))
train <- mushroom[-L,]
test <- mushroom[L,]

#
#
# Naive Bayes
#
fitBayes = naiveBayes(class~.,data=train,laplace=1)

# predict Output
predBayes = predict(fitBayes, test)
predBayes

# matriz de confusão
c_matrix = table(predBayes,test$class)
```

```
print(c_matrix)


cat('Accuracy Bayes: ', sum(diag(c_matrix))/sum(c_matrix)*100, ' %', "\n")



#

#

# Decision Tree

#


#variáveis categórias sendo transformada em variáveis dummies

mushroom = cbind(mushroom, dummy('V1', data = mushroom, sep = ".", drop = TRUE, fun = as.integer, verbose = FALSE))

mushroom = cbind(mushroom, dummy('V2', data = mushroom, sep = ".", drop = TRUE, fun = as.integer, verbose = FALSE))

mushroom = cbind(mushroom, dummy('V3', data = mushroom, sep = ".", drop = TRUE, fun = as.integer, verbose = FALSE))

mushroom = cbind(mushroom, dummy('V4', data = mushroom, sep = ".", drop = TRUE, fun = as.integer, verbose = FALSE))

mushroom = cbind(mushroom, dummy('V5', data = mushroom, sep = ".", drop = TRUE, fun = as.integer, verbose = FALSE))

mushroom = cbind(mushroom, dummy('V6', data = mushroom, sep = ".", drop = TRUE, fun = as.integer, verbose = FALSE))

mushroom = cbind(mushroom, dummy('V7', data = mushroom, sep = ".", drop = TRUE, fun = as.integer, verbose = FALSE))

mushroom = cbind(mushroom, dummy('V8', data = mushroom, sep = ".", drop = TRUE, fun = as.integer, verbose = FALSE))

mushroom = cbind(mushroom, dummy('V9', data = mushroom, sep = ".", drop = TRUE, fun = as.integer, verbose = FALSE))

mushroom = cbind(mushroom, dummy('V10', data = mushroom, sep = ".", drop = TRUE, fun = as.integer, verbose = FALSE))

mushroom = cbind(mushroom, dummy('V11', data = mushroom, sep = ".", drop = TRUE, fun = as.integer, verbose = FALSE))

mushroom = cbind(mushroom, dummy('V12', data = mushroom, sep = ".", drop = TRUE, fun = as.integer, verbose = FALSE))
```

```
mushroom = cbind(mushroom, dummy('V13', data = mushroom, sep = ".", drop = TRUE, fun = as.integer, verbose = FALSE))

mushroom = cbind(mushroom, dummy('V14', data = mushroom, sep = ".", drop = TRUE, fun = as.integer, verbose = FALSE))

mushroom = cbind(mushroom, dummy('V15', data = mushroom, sep = ".", drop = TRUE, fun = as.integer, verbose = FALSE))

mushroom = cbind(mushroom, dummy('V16', data = mushroom, sep = ".", drop = TRUE, fun = as.integer, verbose = FALSE))

mushroom = cbind(mushroom, dummy('V17', data = mushroom, sep = ".", drop = TRUE, fun = as.integer, verbose = FALSE))

mushroom = cbind(mushroom, dummy('V18', data = mushroom, sep = ".", drop = TRUE, fun = as.integer, verbose = FALSE))

mushroom = cbind(mushroom, dummy('V19', data = mushroom, sep = ".", drop = TRUE, fun = as.integer, verbose = FALSE))

mushroom = cbind(mushroom, dummy('V20', data = mushroom, sep = ".", drop = TRUE, fun = as.integer, verbose = FALSE))

mushroom = cbind(mushroom, dummy('V21', data = mushroom, sep = ".", drop = TRUE, fun = as.integer, verbose = FALSE))

mushroom = cbind(mushroom, dummy('V22', data = mushroom, sep = ".", drop = TRUE, fun = as.integer, verbose = FALSE))


excluir <- c('V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10',
        'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19',
        'V20', 'V21', 'V22')


mushroom <- mushroom[,!(names(mushroom)%in% excluir)]


L <- sample(1:nrow(mushroom),round(nrow(mushroom)/3))

train <- mushroom[-L,]

test <- mushroom[L,]


fitTree = ctree(class~.,data=train)


# predict Output

predTree = predict(fitTree, test)
```

```
predTree


# matriz de confusão
c_matrix = table(predTree,test$class)
print(c_matrix)


cat('Accuracy Dtree: ', sum(diag(c_matrix))/sum(c_matrix)*100, ' %', "\n")


plot(fitTree)
```