

Aluno: Ernani de Souza Cubas Neto

Seed utilizado(CPF): 728078902

Trabalho Final

CLASSIFICAÇÃO

Veículo

Técnica	Parâmetro	Acurácia	Matriz de Confusão																									
SVM – Melhor modelo	C=100 Sigma=0.01	0.8235	<table><tr><td></td><td>bus</td><td>opel</td><td>saab</td><td>van</td></tr><tr><td>bus</td><td>37</td><td>1</td><td>0</td><td>1</td></tr><tr><td>opel</td><td>0</td><td>23</td><td>13</td><td>0</td></tr><tr><td>saab</td><td>0</td><td>14</td><td>31</td><td>0</td></tr><tr><td>van</td><td>1</td><td>0</td><td>0</td><td>49</td></tr></table>		bus	opel	saab	van	bus	37	1	0	1	opel	0	23	13	0	saab	0	14	31	0	van	1	0	0	49
	bus	opel	saab	van																								
bus	37	1	0	1																								
opel	0	23	13	0																								
saab	0	14	31	0																								
van	1	0	0	49																								
RNA – Melhor modelo	size=11 decay=0.4	0.8118	<table><tr><td></td><td>bus</td><td>opel</td><td>saab</td><td>van</td></tr><tr><td>bus</td><td>38</td><td>0</td><td>0</td><td>1</td></tr><tr><td>opel</td><td>0</td><td>24</td><td>14</td><td>1</td></tr><tr><td>saab</td><td>0</td><td>14</td><td>30</td><td>2</td></tr><tr><td>van</td><td>0</td><td>0</td><td>0</td><td>46</td></tr></table>		bus	opel	saab	van	bus	38	0	0	1	opel	0	24	14	1	saab	0	14	30	2	van	0	0	0	46
	bus	opel	saab	van																								
bus	38	0	0	1																								
opel	0	24	14	1																								
saab	0	14	30	2																								
van	0	0	0	46																								
SVM – CV	C=1 Sigma=0.0801345	0.7647	<table><tr><td></td><td>bus</td><td>opel</td><td>saab</td><td>van</td></tr><tr><td>bus</td><td>37</td><td>0</td><td>0</td><td>1</td></tr></table>		bus	opel	saab	van	bus	37	0	0	1															
	bus	opel	saab	van																								
bus	37	0	0	1																								

			<table><tr><td>opel</td><td>0</td><td>21</td><td>18</td><td>0</td></tr><tr><td>saab</td><td>0</td><td>16</td><td>23</td><td>0</td></tr><tr><td>van</td><td>1</td><td>1</td><td>3</td><td>49</td></tr></table>	opel	0	21	18	0	saab	0	16	23	0	van	1	1	3	49										
opel	0	21	18	0																								
saab	0	16	23	0																								
van	1	1	3	49																								
SVM – Hold-out	C=1 Sigma=0.0801345	0.7647	<table><tr><td></td><td>bus</td><td>opel</td><td>saab</td><td>van</td></tr><tr><td>bus</td><td>37</td><td>0</td><td>0</td><td>1</td></tr><tr><td>opel</td><td>0</td><td>21</td><td>18</td><td>0</td></tr><tr><td>saab</td><td>0</td><td>16</td><td>23</td><td>0</td></tr><tr><td>van</td><td>1</td><td>1</td><td>3</td><td>49</td></tr></table>		bus	opel	saab	van	bus	37	0	0	1	opel	0	21	18	0	saab	0	16	23	0	van	1	1	3	49
	bus	opel	saab	van																								
bus	37	0	0	1																								
opel	0	21	18	0																								
saab	0	16	23	0																								
van	1	1	3	49																								
RF – Melhor modelo	mtry=9	0.7471	<table><tr><td></td><td>bus</td><td>opel</td><td>saab</td><td>van</td></tr><tr><td>bus</td><td>37</td><td>3</td><td>1</td><td>1</td></tr><tr><td>opel</td><td>0</td><td>18</td><td>17</td><td>0</td></tr><tr><td>saab</td><td>0</td><td>15</td><td>23</td><td>0</td></tr><tr><td>van</td><td>1</td><td>2</td><td>3</td><td>49</td></tr></table>		bus	opel	saab	van	bus	37	3	1	1	opel	0	18	17	0	saab	0	15	23	0	van	1	2	3	49
	bus	opel	saab	van																								
bus	37	3	1	1																								
opel	0	18	17	0																								
saab	0	15	23	0																								
van	1	2	3	49																								
RF – CV	mtry=2	0.7412	<table><tr><td></td><td>bus</td><td>opel</td><td>saab</td><td>van</td></tr><tr><td>bus</td><td>37</td><td>2</td><td>0</td><td>1</td></tr><tr><td>opel</td><td>0</td><td>20</td><td>19</td><td>0</td></tr><tr><td>saab</td><td>0</td><td>15</td><td>20</td><td>0</td></tr><tr><td>van</td><td>1</td><td>1</td><td>5</td><td>49</td></tr></table>		bus	opel	saab	van	bus	37	2	0	1	opel	0	20	19	0	saab	0	15	20	0	van	1	1	5	49
	bus	opel	saab	van																								
bus	37	2	0	1																								
opel	0	20	19	0																								
saab	0	15	20	0																								
van	1	1	5	49																								
RF – Hold-out	mtry=2	0.7294	<table><tr><td></td><td>bus</td><td>opel</td><td>saab</td><td>van</td></tr><tr><td>bus</td><td>37</td><td>2</td><td>0</td><td>1</td></tr></table>		bus	opel	saab	van	bus	37	2	0	1															
	bus	opel	saab	van																								
bus	37	2	0	1																								

			<table><tr><td>opel</td><td>0</td><td>18</td><td>20</td><td>0</td></tr><tr><td>saab</td><td>0</td><td>17</td><td>20</td><td>0</td></tr><tr><td>van</td><td>1</td><td>1</td><td>4</td><td>49</td></tr></table>	opel	0	18	20	0	saab	0	17	20	0	van	1	1	4	49										
opel	0	18	20	0																								
saab	0	17	20	0																								
van	1	1	4	49																								
KNN	k=1	0.6529	<table><tr><td></td><td>bus</td><td>opel</td><td>saab</td><td>van</td></tr><tr><td>bus</td><td>30</td><td>2</td><td>4</td><td>1</td></tr><tr><td>opel</td><td>1</td><td>18</td><td>23</td><td>0</td></tr><tr><td>saab</td><td>7</td><td>16</td><td>16</td><td>2</td></tr><tr><td>van</td><td>0</td><td>2</td><td>1</td><td>47</td></tr></table>		bus	opel	saab	van	bus	30	2	4	1	opel	1	18	23	0	saab	7	16	16	2	van	0	2	1	47
	bus	opel	saab	van																								
bus	30	2	4	1																								
opel	1	18	23	0																								
saab	7	16	16	2																								
van	0	2	1	47																								
RNA – Hold-out	size=5 decay=0.1	0.6412	<table><tr><td></td><td>bus</td><td>opel</td><td>saab</td><td>van</td></tr><tr><td>bus</td><td>34</td><td>3</td><td>3</td><td>1</td></tr><tr><td>opel</td><td>3</td><td>29</td><td>34</td><td>4</td></tr><tr><td>saab</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>van</td><td>1</td><td>6</td><td>6</td><td>45</td></tr></table>		bus	opel	saab	van	bus	34	3	3	1	opel	3	29	34	4	saab	0	0	1	0	van	1	6	6	45
	bus	opel	saab	van																								
bus	34	3	3	1																								
opel	3	29	34	4																								
saab	0	0	1	0																								
van	1	6	6	45																								
RNA – CV	size=5 decay=0.1	0.5882	<table><tr><td></td><td>bus</td><td>opel</td><td>saab</td><td>van</td></tr><tr><td>bus</td><td>32</td><td>1</td><td>0</td><td>0</td></tr><tr><td>opel</td><td>3</td><td>19</td><td>24</td><td>0</td></tr><tr><td>saab</td><td>1</td><td>7</td><td>8</td><td>9</td></tr><tr><td>van</td><td>2</td><td>11</td><td>12</td><td>41</td></tr></table>		bus	opel	saab	van	bus	32	1	0	0	opel	3	19	24	0	saab	1	7	8	9	van	2	11	12	41
	bus	opel	saab	van																								
bus	32	1	0	0																								
opel	3	19	24	0																								
saab	1	7	8	9																								
van	2	11	12	41																								

Melhor modelo: SVM C=100 Sigma=0.01

Novos casos:

	Comp	Circ	DCirc	RadRa	PrAxisRa	MaxLRa	ScatRa	Elong	PrAxisRect	MaxLRect	ScVarMaxis
1	100	48	83	178	72	10	162	42	20	159	176
2	91	40	84	141	57	9	149	45	18	143	170
3	92	50	106	209	66	10	207	32	23	160	223

ScVarmaxis	RaGyr	SkewMaxis	Skewmaxis	Kurtmaxis	KurtMaxis	HollRa	predict.melhor_modelo
400	184	70	6	16	187	197	van
330	70	72	9	14	189	199	van
635	220	73	14	9	188	230	saab

Código (em R):

```
library("caret")
library(mlbench)
library(mice)

##Maquina MP
setwd('C:\\Users\\escneto\\Documents\\Estudos\\Pos_IA_UFPR\\pos_ia_a
prendizado_maquina\\Bases_de_teste')
barra = "\\"
##Note
setwd('/Users/MPPR/Documents/Pos_IA/pos_ia_aprendizado_maquina/Bases
_de_teste')
barra = "/"

dados <- read.csv(file = paste('veiculos', 'veiculos.csv', sep
=barra))
dados_novos <- read.csv(file =
paste('veiculos', 'veiculos_novos.csv', sep =barra))

### retira id
dados$a <- NULL
dados_novos$a <- NULL

###Cria arquivo de treino e teste
set.seed(728078902)
ran <- sample(1:nrow(dados), 0.8 * nrow(dados))
```

```
treino <- dados[ran,]
teste <- dados[-ran,]

##### KNN
set.seed(728078902)
tuneGrid <- expand.grid(k = c(1))
knn <- train(tipo ~ ., data = treino, method =
"kn", tuneGrid=tuneGrid)
knn
predict.knn <- predict(knn, teste)
confusionMatrix(predict.knn, as.factor(teste$tipo))

##### KNN

##### RNA
set.seed(728078902)
rna <- train(tipo~., data=treino, method="nnet", trace=FALSE)
rna
predict.rna <- predict(rna, teste)
confusionMatrix(predict.rna, as.factor(teste$tipo))

###Cross Validation
set.seed(728078902)
ctrl <- trainControl(method = "cv", number = 10)
rna_cv <-
train(tipo~., data=treino, method="nnet", trace=FALSE, trControl=ctrl)
rna_cv
predict.rna_cv <- predict(rna_cv, teste)
confusionMatrix(predict.rna_cv, as.factor(teste$tipo))

###Parametrização
set.seed(728078902)
grid <- expand.grid(size=seq(from=1, to=45, by=10), decay=seq(from=0.1,
to=0.9, by=0.3))
rna_par <- train(form=tipo~., data=treino, method="nnet",
tuneGrid=grid, trControl=ctrl, maxit=2000, trace=FALSE)
rna_par
predict.rna_par <- predict(rna_par, teste)
confusionMatrix(predict.rna_par, as.factor(teste$tipo))
```

```
##### RNA

##### SVM
set.seed(728078902)
svm <- train(tipo~.,data=treino,method="svmRadial")
svm
predict.svm <- predict(svm,teste)
confusionMatrix(predict.svm,as.factor(teste$tipo))

###Cross Validation
set.seed(728078902)
ctrl <- trainControl(method="cv",number=10)
svm_cv <-
train(tipo~.,data=treino,method="svmRadial",trControl=ctrl)
svm_cv
predict.svm_cv <- predict(svm_cv,teste)
confusionMatrix(predict.svm_cv,as.factor(teste$tipo))

###Parametrização
set.seed(728078902)
tuneGrid = expand.grid(C=c(1,2,10,50,100),sigma=c(.01,.015,0.2))
svm_par <-
train(tipo~.,data=treino,method="svmRadial",trControl=ctrl,tuneGrid=
tuneGrid)
svm_par
predict.svm_par <- predict(svm_par,teste)
confusionMatrix(predict.svm_par,as.factor(teste$tipo))
##### SVM

##### Random Forest
set.seed(728078902)
rf <- train(tipo~.,data=treino,method="rf")
rf
predict.rf <- predict(rf,teste)
confusionMatrix(predict.rf,as.factor(teste$tipo))

###Cross Validation
set.seed(728078902)
ctrl <- trainControl(method="cv",number=10)
```

```
rf_cv <- train(tipo~.,data=treino,method="rf",trControl=ctrl)
rf_cv
predict.rf_cv <- predict(rf_cv,teste)
confusionMatrix(predict.rf_cv,as.factor(teste$tipo))

###Parametrização
set.seed(728078902)
tuneGrid = expand.grid(mtry=c(2, 5, 7, 9))
rf_par <-
train(tipo~.,data=treino,method="rf",trControl=ctrl,tuneGrid=tuneGrid)
rf_par
predict.rf_par <- predict(rf_par,teste)
confusionMatrix(predict.rf_par,as.factor(teste$tipo))

##### Random Forest

##### Novos casos
dados_novos$tipo <- NULL
predict.melhor_modelo <- predict(svm_par,dados_novos)
dados_novos <- cbind(dados_novos,predict.melhor_modelo)
View(dados_novos)
##### Novos casos
```

Imposto de Renda

Técnica	Parâmetro	Acurácia	Matriz de Confusão																
RNA – Hold-out	size=5 decay=0.1	1	<table> <tr> <th></th><th>sim</th><th>não</th><th>talvez</th></tr> <tr> <th>sim</th><td>5</td><td>0</td><td>0</td></tr> <tr> <th>não</th><td>0</td><td>2</td><td>0</td></tr> <tr> <th>talvez</th><td>0</td><td>0</td><td>3</td></tr> </table>		sim	não	talvez	sim	5	0	0	não	0	2	0	talvez	0	0	3
	sim	não	talvez																
sim	5	0	0																
não	0	2	0																
talvez	0	0	3																
RNA – CV	size=5 decay=0.1	1	<table> <tr> <th></th><th>sim</th><th>não</th><th>talvez</th></tr> <tr> <th>sim</th><td>5</td><td>0</td><td>0</td></tr> <tr> <th>não</th><td>0</td><td>2</td><td>0</td></tr> <tr> <th>talvez</th><td>0</td><td>0</td><td>3</td></tr> </table>		sim	não	talvez	sim	5	0	0	não	0	2	0	talvez	0	0	3
	sim	não	talvez																
sim	5	0	0																
não	0	2	0																
talvez	0	0	3																
RNA – Melhor modelo	size=11 decay=0.1	1	<table> <tr> <th></th><th>sim</th><th>não</th><th>talvez</th></tr> <tr> <th>sim</th><td>5</td><td>0</td><td>0</td></tr> <tr> <th>não</th><td>0</td><td>2</td><td>0</td></tr> <tr> <th>talvez</th><td>0</td><td>0</td><td>3</td></tr> </table>		sim	não	talvez	sim	5	0	0	não	0	2	0	talvez	0	0	3
	sim	não	talvez																
sim	5	0	0																
não	0	2	0																
talvez	0	0	3																
KNN	k=1	1	<table> <tr> <th></th><th>sim</th><th>não</th><th>talvez</th></tr> <tr> <th>sim</th><td>5</td><td>0</td><td>0</td></tr> <tr> <th>não</th><td>0</td><td>2</td><td>0</td></tr> <tr> <th>talvez</th><td>0</td><td>0</td><td>3</td></tr> </table>		sim	não	talvez	sim	5	0	0	não	0	2	0	talvez	0	0	3
	sim	não	talvez																
sim	5	0	0																
não	0	2	0																
talvez	0	0	3																

SVM – Hold-out	C=1 Sigma=0.8047658	1	<table> <tr> <th></th><th>sim</th><th>não</th><th>talvez</th></tr> <tr> <th>sim</th><td>5</td><td>0</td><td>0</td></tr> <tr> <th>não</th><td>0</td><td>2</td><td>0</td></tr> <tr> <th>talvez</th><td>0</td><td>0</td><td>3</td></tr> </table>		sim	não	talvez	sim	5	0	0	não	0	2	0	talvez	0	0	3
	sim	não	talvez																
sim	5	0	0																
não	0	2	0																
talvez	0	0	3																
SVM – CV	C=0.5 Sigma=0.8047658	1	<table> <tr> <th></th><th>sim</th><th>não</th><th>talvez</th></tr> <tr> <th>sim</th><td>5</td><td>0</td><td>0</td></tr> <tr> <th>não</th><td>0</td><td>2</td><td>0</td></tr> <tr> <th>talvez</th><td>0</td><td>0</td><td>3</td></tr> </table>		sim	não	talvez	sim	5	0	0	não	0	2	0	talvez	0	0	3
	sim	não	talvez																
sim	5	0	0																
não	0	2	0																
talvez	0	0	3																
SVM – Melhor modelo	C=1 Sigma=0.2	1	<table> <tr> <th></th><th>sim</th><th>não</th><th>talvez</th></tr> <tr> <th>sim</th><td>5</td><td>0</td><td>0</td></tr> <tr> <th>não</th><td>0</td><td>2</td><td>0</td></tr> <tr> <th>talvez</th><td>0</td><td>0</td><td>3</td></tr> </table>		sim	não	talvez	sim	5	0	0	não	0	2	0	talvez	0	0	3
	sim	não	talvez																
sim	5	0	0																
não	0	2	0																
talvez	0	0	3																
RF – Hold-out	mtry=4	1	<table> <tr> <th></th><th>sim</th><th>não</th><th>talvez</th></tr> <tr> <th>sim</th><td>5</td><td>0</td><td>0</td></tr> <tr> <th>não</th><td>0</td><td>2</td><td>0</td></tr> <tr> <th>talvez</th><td>0</td><td>0</td><td>3</td></tr> </table>		sim	não	talvez	sim	5	0	0	não	0	2	0	talvez	0	0	3
	sim	não	talvez																
sim	5	0	0																
não	0	2	0																
talvez	0	0	3																
RF – CV	mtry=2	1	<table> <tr> <th></th><th>sim</th><th>não</th><th>talvez</th></tr> <tr> <th>sim</th><td>5</td><td>0</td><td>0</td></tr> </table>		sim	não	talvez	sim	5	0	0								
	sim	não	talvez																
sim	5	0	0																

			<table> <tr> <td>não</td><td>0</td><td>2</td><td>0</td></tr> <tr> <td>talvez</td><td>0</td><td>0</td><td>3</td></tr> </table>	não	0	2	0	talvez	0	0	3								
não	0	2	0																
talvez	0	0	3																
RF – Melhor modelo	mtry=2	1	<table> <tr> <td></td><td>sim</td><td>não</td><td>talvez</td></tr> <tr> <td>sim</td><td>5</td><td>0</td><td>0</td></tr> <tr> <td>não</td><td>0</td><td>2</td><td>0</td></tr> <tr> <td>talvez</td><td>0</td><td>0</td><td>3</td></tr> </table>		sim	não	talvez	sim	5	0	0	não	0	2	0	talvez	0	0	3
	sim	não	talvez																
sim	5	0	0																
não	0	2	0																
talvez	0	0	3																

Melhor modelo: Todos com mesmo resultado

Novos casos:

rest	ecivil	rendimento	predict.melhor_modelo
Sim	Solteiro	99000	Sim
Nao	Casado	9999	Talvez
Nao	Solteiro	73200	Talvez

Código (em R):

```
library("caret")
library(mlbench)
library(mice)

##Maquina MP
setwd('C:\\Users\\escneto\\Documents\\Estudos\\Pos_IA_UFPR\\pos_ia_a
prendizado_maquina\\Bases_de_teste')
barra = "\\"
##Note
setwd('/Users/MPPR/Documents/Pos_IA/pos_ia_aprendizado_maquina/Bases
_de_teste')
barra = "/"
```

```
dados <- read.csv(file = paste('IR', 'IR.csv', sep = 'barra'))
dados_novos <- read.csv(file = paste('IR', 'IR_novos.csv', sep = 'barra'))

### Cria arquivos de treino e teste
set.seed(728078902)
ran <- sample(1:nrow(dados), 0.8 * nrow(dados))
treino <- dados[ran,]
teste <- dados[-ran,]

##### KNN
set.seed(728078902)
tuneGrid <- expand.grid(k = c(1,3,5,7,9))
knn <- train(sonegador~., data = treino, method = "knn", tuneGrid=tuneGrid)
knn
predict.knn <- predict(knn, teste)
confusionMatrix(predict.knn, as.factor(teste$sonegador))
##### KNN

##### RNA
###Hold-out
set.seed(728078902)
rna <- train(sonegador~., data=treino, method="nnet", trace=FALSE)
rna
predict.rna <- predict(rna, teste)
confusionMatrix(predict.rna, as.factor(teste$sonegador))

###Cross Validation
set.seed(728078902)
ctrl <- trainControl(method = "cv", number = 10)
rna_cv <-
train(sonegador~., data=treino, method="nnet", trace=FALSE, trControl=ctrl)
rna_cv
predict.rna_cv <- predict(rna_cv, teste)
confusionMatrix(predict.rna_cv, as.factor(teste$sonegador))

###Parametrização
```

```
set.seed(728078902)
grid <- expand.grid(size=seq(from=1,to=45,by=10),decay=seq(from=0.1,
to=0.9, by=0.3))
rna_par <- train(form=sonegador~., data=treino, method="nnet",
tuneGrid=grid, trControl=ctrl, maxit=2000, trace=FALSE)
rna_par
predict.rna_par <- predict(rna_par, teste)
confusionMatrix(predict.rna_par, as.factor(teste$sonegador))
##### RNA

##### SVM
###Hold-out
set.seed(728078902)
svm <- train(sonegador~., data=treino, method="svmRadial")
svm
predict.svm <- predict(svm, teste)
confusionMatrix(predict.svm, as.factor(teste$sonegador))

###Cross Validation
set.seed(728078902)
ctrl <- trainControl(method="cv", number=10)
svm_cv <-
train(sonegador~., data=treino, method="svmRadial", trControl=ctrl)
svm_cv
predict.svm_cv <- predict(svm_cv, teste)
confusionMatrix(predict.svm_cv, as.factor(teste$sonegador))

###Parametrização
set.seed(728078902)
tuneGrid = expand.grid(C=c(1,2,10,50,100), sigma=c(.01, .015, 0.2))
svm_par <-
train(sonegador~., data=treino, method="svmRadial", trControl=ctrl, tune
Grid=tuneGrid)
svm_par
predict.svm_par <- predict(svm_par, teste)
confusionMatrix(predict.svm_par, as.factor(teste$sonegador))
##### SVM

##### Random Forest
```

```
set.seed(728078902)
rf <- train(sonegador~.,data=treino,method="rf")
rf
predict.rf <- predict(rf,teste)
confusionMatrix(predict.rf,as.factor(teste$sonegador))

###Cross Validation
set.seed(728078902)
ctrl <- trainControl(method="cv",number=10)
rf_cv <- train(sonegador~.,data=treino,method="rf",trControl=ctrl)
rf_cv
predict.rf_cv <- predict(rf_cv,teste)
confusionMatrix(predict.rf_cv,as.factor(teste$sonegador))

###Parametrização
set.seed(728078902)
tuneGrid = expand.grid(mtry=c(2, 5, 7, 9))
rf_par <-
train(sonegador~.,data=treino,method="rf",trControl=ctrl,tuneGrid=tu
neGrid)
rf_par
predict.rf_par <- predict(rf_par,teste)
confusionMatrix(predict.rf_par,as.factor(teste$sonegador))

##### Random Forest

##### Novos casos
dados_novos$sonegador <-NULL
predict.melhor_modelo <- predict(rna,dados_novos)
dados_novos <-cbind(dados_novos,predict.melhor_modelo)
View(dados_novos)
##### Novos casos
```

Diabetes

Técnica	Parâmetro	Acurácia	Matriz de Confusão		
SVM – Melhor modelo	C=1 Sigma=0.015	0.7597		neg	pos
			neg	85	25
			pos	12	32
RNA – Melhor modelo	size=11 decay=0.4	0.7597		neg	pos
			neg	81	21
			pos	16	36
SVM – CV	C=0.25 Sigma=0.163625	0.7403		neg	pos
			neg	86	29
			pos	11	28
SVM – Hold-out	C=0.5 Sigma=0.163625	0.7338		neg	pos
			neg	85	29
			pos	12	28
KNN	k=9	0.7273		neg	pos
			neg	83	28
			pos	14	29
RF – Hold-out	mtry=2	0.7208		neg	pos



UFPR – Universidade Federal do Paraná
Setor de Educação Profissional e Tecnológica
Especialização em Inteligência Artificial Aplicada
Turma 2022



Disciplina: Laboratório de IA – Prof Jaime Wojciechowski

			<table><tr><td>neg</td><td>84</td><td>30</td></tr><tr><td>pos</td><td>13</td><td>27</td></tr></table>	neg	84	30	pos	13	27			
neg	84	30										
pos	13	27										
RF – Melhor modelo	mtry=2	0.7143	<table><tr><td></td><td>neg</td><td>pos</td></tr><tr><td>neg</td><td>83</td><td>30</td></tr><tr><td>pos</td><td>14</td><td>27</td></tr></table>		neg	pos	neg	83	30	pos	14	27
	neg	pos										
neg	83	30										
pos	14	27										
RF – CV	mtry=2	0.7013	<table><tr><td></td><td>neg</td><td>pos</td></tr><tr><td>neg</td><td>85</td><td>34</td></tr><tr><td>pos</td><td>12</td><td>23</td></tr></table>		neg	pos	neg	85	34	pos	12	23
	neg	pos										
neg	85	34										
pos	12	23										
RNA – CV	size=3 decay=0.1	0.6494	<table><tr><td></td><td>neg</td><td>pos</td></tr><tr><td>neg</td><td>66</td><td>23</td></tr><tr><td>pos</td><td>31</td><td>34</td></tr></table>		neg	pos	neg	66	23	pos	31	34
	neg	pos										
neg	66	23										
pos	31	34										
RNA – Hold-out	size=3 decay=0.1	0.6429	<table><tr><td></td><td>neg</td><td>pos</td></tr><tr><td>neg</td><td>75</td><td>33</td></tr><tr><td>pos</td><td>22</td><td>24</td></tr></table>		neg	pos	neg	75	33	pos	22	24
	neg	pos										
neg	75	33										
pos	22	24										

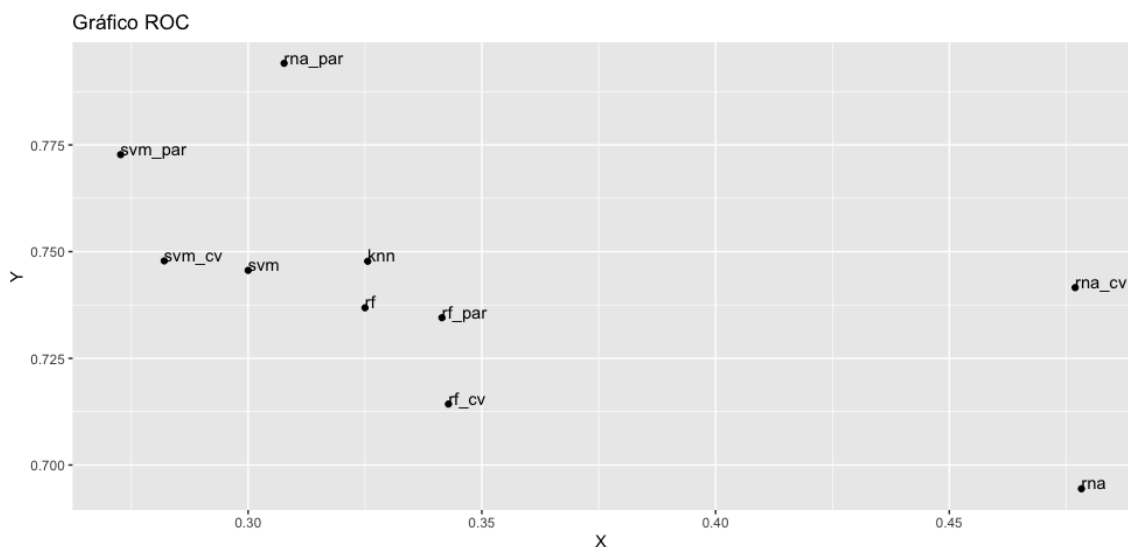
Melhor modelo: SVM C=1 Sigma=0.015

Análise ROC:

	X	Y	distancia
knn	0.3255814	0.7477477	0.4118670
rna	0.4782609	0.6944444	0.5675365
rna_cv	0.4769231	0.7415730	0.5424390
rna_par	0.3076923	0.7941176	0.3702190

	X	Y	distancia
svm	0.3000000	0.7456140	0.3933347
svm_cv	0.2820513	0.7478261	0.3783446
svm_par	0.2727273	0.7727273	0.3550113 <- Menor distância
rf	0.3250000	0.7368421	0.4181831
rf_cv	0.3428571	0.7142857	0.4463000
rf_par	0.3414634	0.7345133	0.4325280

Gráfico ROC:



Novos casos:

preg0nt	glucose	pressure	triceps	insulin	mass	pedigree	age	predict.melhor_modelo
7	130	72	37	0	33.6	0.980	50	pos
2	81	66	29	0	32.6	0.351	31	neg
5	23	64	0	0	23.3	0.672	15	neg

Código (em R):

```
library("caret")
library(mlbench)
library(mice)

##Maquina MP
setwd('C:\\Users\\escneto\\Documents\\Estudos\\Pos_IA_UFPR\\pos_ia_a
```



```
aprendizado_maquina\\Bases_de_teste')
barra = "\\\"
##Note
setwd('/Users/MPPR/Documents/Pos_IA/pos_ia_aprendizado_maquina/Bases
_de_teste')
barra = "/"

dados <- read.csv(file = paste('diabetes','diabetes.csv',sep
=barra))
dados_novos <- read.csv(file =
paste('diabetes','diabetes_novos.csv',sep =barra))
dados$num <- NULL
dados_novos$num <-NULL

### Cria arquivos de treino e teste
set.seed(728078902)
ran <- sample(1:nrow(dados), 0.8 * nrow(dados))
treino <- dados[ran,]
teste <- dados[-ran,]

##### KNN
set.seed(728078902)
tuneGrid <- expand.grid(k = c(1,3,5,7,9))
knn <- train(diabetes~, data = treino, method =
"knn",tuneGrid=tuneGrid)
knn
predict.knn <- predict(knn, teste)
confusionMatrix(predict.knn, as.factor(teste$diabetes))
##### KNN

##### RNA
###Hold-out
set.seed(728078902)
rna <- train(diabetes~, data=treino, method="nnet",trace=FALSE)
rna
predict.rna <- predict(rna, teste)
confusionMatrix(predict.rna, as.factor(teste$diabetes))

###Cross-validation
```

```
set.seed(728078902)
ctrl <- trainControl(method = "cv", number = 10)
rna_cv <- train(diabetes~., data=treino, method="nnet", trace=FALSE,
trControl=ctrl)
rna_cv
predict.rna_cv <- predict(rna_cv, teste)
confusionMatrix(predict.rna_cv, as.factor(teste$diabetes))

##### Parametrização
set.seed(728078902)
grid <- expand.grid(size = seq(from = 1, to = 45, by = 10), decay =
seq(from = 0.1, to = 0.9, by = 0.3))
rna_par <- train(form = diabetes~., data = treino, method =
"nnet", tuneGrid = grid, trControl = ctrl, maxit = 2000, trace=FALSE)
rna_par
predict.rna_par <- predict(rna_par, teste)
confusionMatrix(predict.rna_par, as.factor(teste$diabetes))
##### RNA

##### SVN
### Hold-out
set.seed(728078902)
svm <- train(diabetes~., data=treino, method="svmRadial")
svm
predict.svm <- predict(svm, teste)
confusionMatrix(predict.svm, as.factor(teste$diabetes))

#### Cross-validation
set.seed(728078902)
ctrl <- trainControl(method = "cv", number = 10)
svm_cv <- train(diabetes~., data=treino, method="svmRadial",
trControl=ctrl)
svm_cv
predict.svm_cv <- predict(svm_cv, teste)
confusionMatrix(predict.svm_cv, as.factor(teste$diabetes))

#### Parametrização
set.seed(728078902)
tuneGrid = expand.grid(C=c(1, 2, 10, 50, 100), sigma=c(.01, .015,
```

```
0.2))
svm_par <- train(diabetes~., data=treino, method="svmRadial",
trControl=ctrl, tuneGrid=tuneGrid)
svm_par
predict.svm_par <- predict(svm_par, teste)
confusionMatrix(predict.svm_par, as.factor(teste$diabetes))

##### Random Forest
set.seed(728078902)
rf <- train(diabetes~.,data=treino,method="rf")
rf
predict.rf <- predict(rf,teste)
confusionMatrix(predict.rf,as.factor(teste$diabetes))

###Cross Validation
set.seed(728078902)
ctrl <- trainControl(method="cv",number=10)
rf_cv <- train(diabetes~.,data=treino,method="rf",trControl=ctrl)
rf_cv
predict.rf_cv <- predict(rf_cv,teste)
confusionMatrix(predict.rf_cv,as.factor(teste$diabetes))

###Parametrização
set.seed(728078902)
tuneGrid = expand.grid(mtry=c(2, 5, 7, 9))
rf_par <-
train(diabetes~.,data=treino,method="rf",trControl=ctrl,tuneGrid=tun
eGrid)
rf_par
predict.rf_par <- predict(rf_par,teste)
confusionMatrix(predict.rf_par,as.factor(teste$diabetes))

##### Random Forest

##### Novos casos
dados_novos$diabetes <-NULL
predict.melhor_modelo <- predict(svm_par,dados_novos)
dados_novos <-cbind(dados_novos,predict.melhor_modelo)
View(dados_novos)
```

```
##### Novos casos
```

```
##### Analise ROC
```

```
cmknn <- confusionMatrix(predict.knn,as.factor(teste$diabetes))
cmrna <- confusionMatrix(predict.rna,as.factor(teste$diabetes))
cmrna_cv <-
confusionMatrix(predict.rna_cv,as.factor(teste$diabetes))
cmrna_par <-
confusionMatrix(predict.rna_par,as.factor(teste$diabetes))
cmsvm <- confusionMatrix(predict.svm,as.factor(teste$diabetes))
cmsvm_cv <-
confusionMatrix(predict.svm_cv,as.factor(teste$diabetes))
cmsvm_par <-
confusionMatrix(predict.svm_par,as.factor(teste$diabetes))
cmrf <- confusionMatrix(predict.rf,as.factor(teste$diabetes))
cmrf_cv <- confusionMatrix(predict.rf_cv,as.factor(teste$diabetes))
cmrf_par <-
confusionMatrix(predict.rf_par,as.factor(teste$diabetes))
```

```
####Função
```

```
df.ROC <- data.frame(
  modelo =
c('knn', 'rna', 'rna_cv', 'rna_par', 'svm', 'svm_cv', 'svm_par', 'rf', 'rf_c
v', 'rf_par'),
  X = c(rep(0,10)),
  Y = c(rep(0,10)),
  distancia = c(rep(0,10))
)
```

```
func_ROC <- function(df,cm,modelo) {
  VP <- cm[1,1]
  FP <- cm[2,1]
  VN <- cm[2,2]
  FN <- cm[1,2]

  X <- 1-(VN / (VN+FP))
  Y <- VP / (VP+FN)
```

```
distancia <- sqrt(X^2 + (Y-1)^2)

df[df$modelo==modelo, "X"] <- X
df[df$modelo==modelo, "Y"] <- Y
df[df$modelo==modelo, "distancia"] <- distancia

return (df)
}

df.ROC <- func_ROC(df.ROC, cmknn$table, "knn")
df.ROC <- func_ROC(df.ROC, cmrna$table, "rna")
df.ROC <- func_ROC(df.ROC, cmrna_cv$table, "rna_cv")
df.ROC <- func_ROC(df.ROC, cmrna_par$table, "rna_par")
df.ROC <- func_ROC(df.ROC, cmsvm$table, "svm")
df.ROC <- func_ROC(df.ROC, cmsvm_cv$table, "svm_cv")
df.ROC <- func_ROC(df.ROC, cmsvm_par$table, "svm_par")
df.ROC <- func_ROC(df.ROC, cmrf$table, "rf")
df.ROC <- func_ROC(df.ROC, cmrf_cv$table, "rf_cv")
df.ROC <- func_ROC(df.ROC, cmrf_par$table, "rf_par")

ggplot(df.ROC, aes(x=X, y=Y, label=modelo)) +
  geom_point() +
  labs(x="X", y="Y", title="Gráfico ROC") + geom_text(hjust=0,
vjust=0)
##### Analise ROC
```

Banco

Técnica	Parâmetro	Acurácia	Matriz de Confusão									
RNA – Hold-out	size=1 decay=0	0.85	<table><tr><td></td><td>no</td><td>yes</td></tr><tr><td>no</td><td>51</td><td>9</td></tr><tr><td>yes</td><td>0</td><td>0</td></tr></table>		no	yes	no	51	9	yes	0	0
	no	yes										
no	51	9										
yes	0	0										
RNA – CV	size=1 decay=1e-04	0.85	<table><tr><td></td><td>no</td><td>yes</td></tr><tr><td>no</td><td>51</td><td>9</td></tr><tr><td>yes</td><td>0</td><td>0</td></tr></table>		no	yes	no	51	9	yes	0	0
	no	yes										
no	51	9										
yes	0	0										
RNA – Melhor modelo	size=1 decay=0.7	0.85	<table><tr><td></td><td>no</td><td>yes</td></tr><tr><td>no</td><td>51</td><td>9</td></tr><tr><td>yes</td><td>0</td><td>0</td></tr></table>		no	yes	no	51	9	yes	0	0
	no	yes										
no	51	9										
yes	0	0										
KNN	k=9	0.85	<table><tr><td></td><td>no</td><td>yes</td></tr><tr><td>no</td><td>51</td><td>9</td></tr><tr><td>yes</td><td>0</td><td>0</td></tr></table>		no	yes	no	51	9	yes	0	0
	no	yes										
no	51	9										
yes	0	0										
SVM – Hold-out	C=0.25 Sigma=0.03609159	0.85	<table><tr><td></td><td>no</td><td>yes</td></tr><tr><td>no</td><td>51</td><td>9</td></tr><tr><td>yes</td><td>0</td><td>0</td></tr></table>		no	yes	no	51	9	yes	0	0
	no	yes										
no	51	9										
yes	0	0										
SVM – CV	C=0.25 Sigma=0.03609159	0.85	<table><tr><td></td><td>no</td><td>yes</td></tr></table>		no	yes						
	no	yes										

			<table><tr><td>no</td><td>51</td><td>9</td></tr><tr><td>yes</td><td>0</td><td>0</td></tr></table>	no	51	9	yes	0	0			
no	51	9										
yes	0	0										
RF – Hold-out	mtry=2	0.85	<table><tr><td></td><td>no</td><td>yes</td></tr><tr><td>no</td><td>51</td><td>9</td></tr><tr><td>yes</td><td>0</td><td>0</td></tr></table>		no	yes	no	51	9	yes	0	0
	no	yes										
no	51	9										
yes	0	0										
RF – CV	mtry=2	0.85	<table><tr><td></td><td>no</td><td>yes</td></tr><tr><td>no</td><td>51</td><td>9</td></tr><tr><td>yes</td><td>0</td><td>0</td></tr></table>		no	yes	no	51	9	yes	0	0
	no	yes										
no	51	9										
yes	0	0										
SVM – Melhor modelo	C=1 Sigma=0.2	0.8333	<table><tr><td></td><td>no</td><td>yes</td></tr><tr><td>no</td><td>50</td><td>9</td></tr><tr><td>yes</td><td>1</td><td>0</td></tr></table>		no	yes	no	50	9	yes	1	0
	no	yes										
no	50	9										
yes	1	0										
RF – Melhor modelo	mtry=5	0.8333	<table><tr><td></td><td>no</td><td>yes</td></tr><tr><td>no</td><td>50</td><td>9</td></tr><tr><td>yes</td><td>1</td><td>0</td></tr></table>		no	yes	no	50	9	yes	1	0
	no	yes										
no	50	9										
yes	1	0										

Melhor modelo: Excetuando os modelos SVM e RF parametrizados, todos tiveram o mesmo resultado

Novos casos:

age	job	marital	education	default	balance	housing	loan	predict.melhor_modelo
60	unemployed	married	primary	no	2000	yes	yes	no
33	services	married	secondary	yes	3000	yes	no	no
15	management	single	tertiary	no	1350	yes	no	no

Código (em R):

```
library("caret")
library(mlbench)
library(mice)

##Maquina MP
setwd('C:\\Users\\escneto\\Documents\\Estudos\\Pos_IA_UFPR\\pos_ia_a
prendizado_maquina\\Bases_de_teste')
barra = "\\"
##Note
setwd('/Users/MPPR/Documents/Pos_IA/pos_ia_aprendizado_maquina/Bases
_de_teste')
barra = "/"

dados <- read.csv(file = paste('banco', 'banco.csv', sep = barra))
dados_novos <- read.csv(file = paste('banco', 'banco_novos.csv', sep
=barra))

### Cria arquivos de treino e teste
set.seed(728078902)
ran <- sample(1:nrow(dados), 0.8 * nrow(dados))
treino <- dados[ran,]
teste <- dados[-ran,]

##### KNN
set.seed(728078902)
tuneGrid <- expand.grid(k = c(1,3,5,7,9))
knn <- train(y~, data = treino, method = "knn", tuneGrid=tuneGrid)
knn
predict.knn <- predict(knn, teste)
confusionMatrix(predict.knn, as.factor(teste$y))
##### KNN

##### RNA
### HoId-out
set.seed(728078902)
rna <- train(y~, data=treino, method="nnet", trace=FALSE)
```



```
rna
predict.rna <- predict(rna, teste)
confusionMatrix(predict.rna, as.factor(teste$y))

### Cross-validation
set.seed(728078902)
ctrl <- trainControl(method = "cv", number = 10)
rna_cv <- train(y~., data=treino, method="nnet", trace=FALSE,
trControl=ctrl)
rna_cv
predict.rna_cv <- predict(rna_cv, teste)
confusionMatrix(predict.rna_cv, as.factor(teste$y))

### Parametrização
set.seed(728078902)
grid <- expand.grid(size = seq(from = 1, to = 45, by = 10), decay =
seq(from = 0.1, to = 0.9, by = 0.3))
rna_par <- train(form = y~., data = treino, method = "nnet", tuneGrid =
grid, trControl = ctrl, maxit = 2000, trace=FALSE)
rna_par
predict.rna_par <- predict(rna_par, teste)
confusionMatrix(predict.rna_par, as.factor(teste$y))

##### RNA

##### SVN
set.seed(728078902)
svm <- train(y~., data=treino, method="svmRadial")
svm
predict.svm <- predict(svm, teste)
confusionMatrix(predict.svm, as.factor(teste$y))

#### Cross-validation SVM
set.seed(728078902)
ctrl <- trainControl(method = "cv", number = 10)
svm_cv <- train(y~., data=treino, method="svmRadial",
trControl=ctrl)
svm_cv
predict.svm_cv <- predict(svm_cv, teste)
```

```
confusionMatrix(predict.svm_cv, as.factor(teste$y))

#### Parametrização
set.seed(728078902)
tuneGrid = expand.grid(C=c(1, 2, 10, 50, 100), sigma=c(.01, .015,
0.2))
svm_par <- train(y~., data=treino, method="svmRadial",
trControl=ctrl, tuneGrid=tuneGrid)
svm_par
predict.svm_par <- predict(svm_par, teste)
confusionMatrix(predict.svm_par, as.factor(teste$y))

##### SVN

##### Random Forest
set.seed(728078902)
rf <- train(y~.,data=treino,method="rf")
rf
predict.rf <-predict(rf,teste)
confusionMatrix(predict.rf,as.factor(teste$y))

##Cross Validation
set.seed(728078902)
ctrl <- trainControl(method = "cv", number = 10)
rf_cv <- train(y~.,data=treino,method="rf",trControl=ctrl)
rf_cv
predict.rf_cv <-predict(rf_cv,teste)
confusionMatrix(predict.rf_cv,as.factor(teste$y))

##Parametrização
tuneGrid = expand.grid(mtry=c(2,5,7,9))
rf_par <-
train(y~.,data=treino,method="rf",trControl=ctrl,tuneGrid=tuneGrid)
rf_par
predict.rf_par <- predict(rf_par,teste)
confusionMatrix(predict.rf_par,as.factor(teste$y))

##### Random Forest
```



UFPR – Universidade Federal do Paraná
Setor de Educação Profissional e Tecnológica
Especialização em Inteligência Artificial Aplicada
Turma 2022



Disciplina: Laboratório de IA – Prof Jaime Wojciechowski

```
##### Novos casos
dados_novos$y <-NULL
predict.melhor_modelo <- predict(svm,dados_novos)
dados_novos <-cbind(dados_novos,predict.melhor_modelo)
View(dados_novos)
##### Novos casos
```

Previsão do Tempo

Técnica	Parâmetro	Acurácia	Matriz de Confusão									
RNA – Hold-out	size=5 decay=0	1	<table><tr><td></td><td>NAO</td><td>SIM</td></tr><tr><td>NAO</td><td>4</td><td>0</td></tr><tr><td>SIM</td><td>0</td><td>4</td></tr></table>		NAO	SIM	NAO	4	0	SIM	0	4
	NAO	SIM										
NAO	4	0										
SIM	0	4										
RNA – CV	size=3 decay=1e-04	1	<table><tr><td></td><td>NAO</td><td>SIM</td></tr><tr><td>NAO</td><td>4</td><td>0</td></tr><tr><td>SIM</td><td>0</td><td>4</td></tr></table>		NAO	SIM	NAO	4	0	SIM	0	4
	NAO	SIM										
NAO	4	0										
SIM	0	4										
RNA – Melhor modelo	size=1 decay=0.01	1	<table><tr><td></td><td>NAO</td><td>SIM</td></tr><tr><td>NAO</td><td>4</td><td>0</td></tr><tr><td>SIM</td><td>0</td><td>4</td></tr></table>		NAO	SIM	NAO	4	0	SIM	0	4
	NAO	SIM										
NAO	4	0										
SIM	0	4										
KNN	k=1	1	<table><tr><td></td><td>NAO</td><td>SIM</td></tr><tr><td>NAO</td><td>4</td><td>0</td></tr><tr><td>SIM</td><td>0</td><td>4</td></tr></table>		NAO	SIM	NAO	4	0	SIM	0	4
	NAO	SIM										
NAO	4	0										
SIM	0	4										
SVM – Melhor modelo	C=2 Sigma=0.2	1	<table><tr><td></td><td>NAO</td><td>SIM</td></tr><tr><td>NAO</td><td>4</td><td>0</td></tr><tr><td>SIM</td><td>0</td><td>4</td></tr></table>		NAO	SIM	NAO	4	0	SIM	0	4
	NAO	SIM										
NAO	4	0										
SIM	0	4										
RF – Hold-out	mtry=6	1	<table><tr><td></td><td>NAO</td><td>SIM</td></tr></table>		NAO	SIM						
	NAO	SIM										

			<table><tr><td>NAO</td><td>4</td><td>0</td></tr><tr><td>SIM</td><td>0</td><td>4</td></tr></table>	NAO	4	0	SIM	0	4			
NAO	4	0										
SIM	0	4										
RF – CV	mtry=6	1	<table><tr><td></td><td>NAO</td><td>SIM</td></tr><tr><td>NAO</td><td>4</td><td>0</td></tr><tr><td>SIM</td><td>0</td><td>4</td></tr></table>		NAO	SIM	NAO	4	0	SIM	0	4
	NAO	SIM										
NAO	4	0										
SIM	0	4										
RF – Melhor modelo	mtry=5	1	<table><tr><td></td><td>NAO</td><td>SIM</td></tr><tr><td>NAO</td><td>4</td><td>0</td></tr><tr><td>SIM</td><td>0</td><td>4</td></tr></table>		NAO	SIM	NAO	4	0	SIM	0	4
	NAO	SIM										
NAO	4	0										
SIM	0	4										
SVM – CV	C=1 Sigma=0.119708	0.875	<table><tr><td></td><td>NAO</td><td>SIM</td></tr><tr><td>NAO</td><td>4</td><td>1</td></tr><tr><td>SIM</td><td>0</td><td>3</td></tr></table>		NAO	SIM	NAO	4	1	SIM	0	3
	NAO	SIM										
NAO	4	1										
SIM	0	3										
SVM – Hold-out	C=1 Sigma=0.119708	0.875	<table><tr><td></td><td>NAO</td><td>SIM</td></tr><tr><td>NAO</td><td>4</td><td>1</td></tr><tr><td>SIM</td><td>0</td><td>3</td></tr></table>		NAO	SIM	NAO	4	1	SIM	0	3
	NAO	SIM										
NAO	4	1										
SIM	0	3										

Melhor modelo: Excetuando SVM Hold-Out e com Cross Validation, todos com mesmo resultado

Novos casos:

Ceu	Temperatura	Umidade	Vento	predict.melhor_modelo
Ensolarado	Elevada	Alta	Sim	SIM
Chuvoso	Elevada	Normal	Sim	NAO
Coberto	Elevada	Alta	Nao	NAO

Código (em R):

```
library("caret")
library("mice")
library(mlbench)

##Maquina MP
setwd('C:\\Users\\escnetto\\Documents\\Estudos\\Pos_IA_UFPR\\pos_ia_a
prendizado_maquina\\Bases_de_teste')
barra = "\\\"

##Note
setwd('/Users/MPPR/Documents/Pos_IA/pos_ia_aprendizado_maquina/Bases
_de_teste')
barra = "/"

dados <- read.csv(file =
paste('previsao_tempo', 'previsao_tempo.csv', sep = barra))
dados_novos <- read.csv(file =
paste('previsao_tempo', 'previsao_tempo_novos.csv', sep = barra))

### Cria arquivos de treino e teste
set.seed(728078902)
ran <- sample(1:nrow(dados), 0.8 * nrow(dados))
treino <- dados[ran,]
teste <- dados[-ran,]

##### KNN
set.seed(728078902)
tuneGrid <- expand.grid(k = c(1,3,5,7,9))
knn <- train(Chovera~., data = treino, method =
"knn", tuneGrid=tuneGrid)
knn
predict.knn <- predict(knn, teste)
confusionMatrix(predict.knn, as.factor(teste$Chovera))
##### KNN

##### RNA
```

```
set.seed(728078902)
rna <- train(Chovera~.,data=treino,method="nnet",trace=FALSE)
rna
predict.rna <- predict(rna,teste)
confusionMatrix(predict.rna,as.factor(teste$Chovera))

###Cross Validation
set.seed(728078902)
ctrl <- trainControl(method = "cv", number = 10)
rna_cv <-
train(Chovera~.,data=treino,method="nnet",trace=FALSE,trControl=ctrl
)
rna_cv
predict.rna_cv <- predict(rna_cv,teste)
confusionMatrix(predict.rna_cv,as.factor(teste$Chovera))

###Parametrização
set.seed(728078902)
grid <- expand.grid(size=seq(from=1,to=45,by=10),decay=seq(from=0.01,
to=0.9, by=0.3))
rna_par <- train(form=Chovera~., data=treino, method="nnet",
tuneGrid=grid, trControl=ctrl, maxit=2000, trace=FALSE)
rna_par
predict.rna_par <- predict(rna_par,teste)
confusionMatrix(predict.rna_par,as.factor(teste$Chovera))
##### RNA

##### SVM
set.seed(728078902)
svm <- train(Chovera~.,data=treino,method="svmRadial")
svm
predict.svm <- predict(svm,teste)
confusionMatrix(predict.svm,as.factor(teste$Chovera))

###Cross Validation
set.seed(728078902)
ctrl <- trainControl(method="cv",number=10)
svm_cv <-
```

```
train(Chovera~.,data=treino,method="svmRadial",trControl=ctrl)
svm_cv
predict.svm_cv <- predict(svm_cv,teste)
confusionMatrix(predict.svm_cv,as.factor(teste$Chovera))

###Parametrização
set.seed(728078902)
tuneGrid = expand.grid(C=c(1,2,10,50,100),sigma=c(.01,.015,0.2))
svm_par <-
train(Chovera~.,data=treino,method="svmRadial",trControl=ctrl,tuneGrid=tuneGrid)
svm_par
predict.svm_par <- predict(svm_par,teste)
confusionMatrix(predict.svm_par,as.factor(teste$Chovera))
##### SVM

##### Random Forest
set.seed(728078902)
rf <- train(Chovera~.,data=treino,method="rf")
rf
predict.rf <- predict(rf,teste)
confusionMatrix(predict.rf,as.factor(teste$Chovera))

###Cross Validation
set.seed(728078902)
ctrl <- trainControl(method="cv",number=10)
rf_cv <- train(Chovera~.,data=treino,method="rf",trControl=ctrl)
rf_cv
predict.rf_cv <- predict(rf_cv,teste)
confusionMatrix(predict.rf_cv,as.factor(teste$Chovera))

###Parametrização
set.seed(728078902)
tuneGrid = expand.grid(mtry=c(2, 5, 7, 9))
rf_par <-
train(Chovera~.,data=treino,method="rf",trControl=ctrl,tuneGrid=tuneGrid)
rf_par
predict.rf_par <- predict(rf_par,teste)
```




UFPR – Universidade Federal do Paraná
Setor de Educação Profissional e Tecnológica
Especialização em Inteligência Artificial Aplicada
Turma 2022



Disciplina: Laboratório de IA – Prof Jaime Wojciechowski

```
confusionMatrix(predict.rf_par,as.factor(teste$Chovera))

##### Random Forest

##### Novos casos
dados_novos$Chovera <-NULL
predict.melhor_modelo <- predict(rf_par,dados_novos)
dados_novos <-cbind(dados_novos,predict.melhor_modelo)
View(dados_novos)
##### Novos casos
```

REGRESSÃO

Biomassa

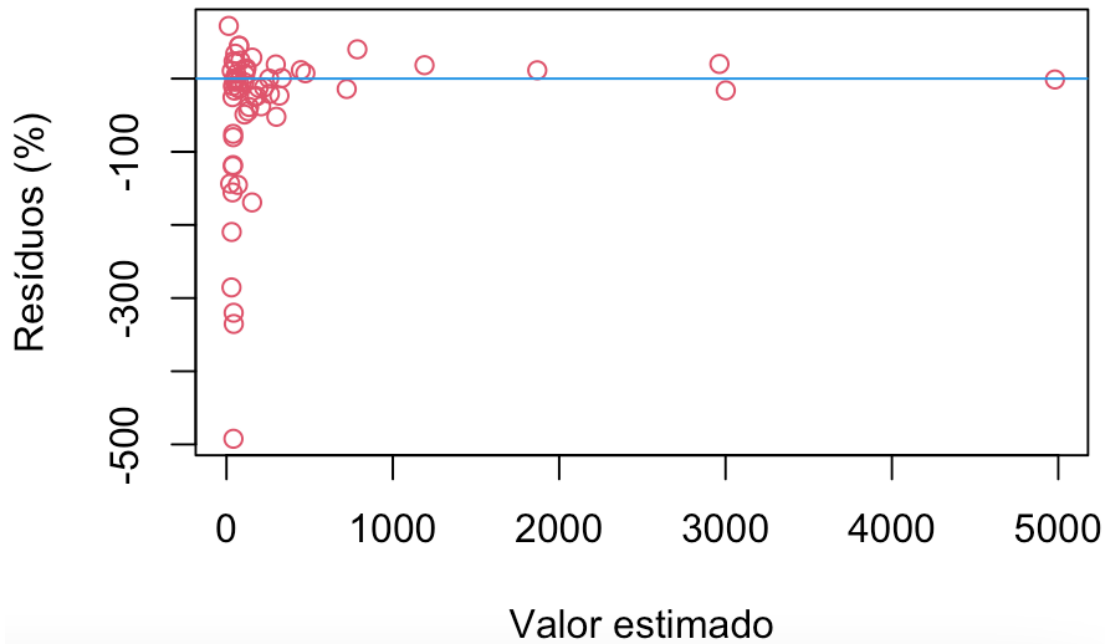
Técnica	Parâmetro	R2	Syx	Pearson	Rmse	MAE
RNA – Melhor modelo	size=9 decay=0.7	0.9711721	145.865	0.9874293	142.1716	63.3358
RF – Hold-out	mtry=2	0.954812	200.7104	0.9774692	195.6283	72.58058
SVM – Melhor modelo	C=100 Sigma=0.01	0.9506294	224.3826	0.9802945	218.7012	90.73753
RF – Melhor modelo	mtry=9	0.9360046	244.0757	0.9681312	237.8956	86.04137
RF – CV	mtry=3	0.934733	249.9613	0.9680431	243.6322	87.5254
KNN	k=1	0.8080971	592.1916	0.9412175	577.197	145.7097
SVM – Hold-out	C=1 Sigma=0.7524557	0.8076903	300.2824	0.9643472	292.6791	137.6448
SVM – CV	C=1 Sigma=0.7524557	0.8076903	300.2824	0.9643472	292.6791	137.6448
RNA – Hold-out	size=3 decay=0.1	0.5981283	460.4957	0.867414	448.8357	198.1331
RNA – CV	size=3 decay=0.1	0.5981283	460.4957	0.867414	448.8357	198.1331

Melhor modelo: RNA size=9 decay=0.7

Novos casos:

dap	h	Me	predict.melhor_caso
6.4	7.0	1.04	44.91187
7.3	10.0	1.04	53.89512
7.8	5.5	1.04	45.05009
12.2	7.5	1.04	65.25170

Gráfico de resíduos:



Código (em R):

```
library(mlbench)
library(caret)
library(mice)
library(Metrics)

##Maquina MP
setwd('C:\\Users\\escneto\\Documents\\Estudos\\Pos_IA_UFPR\\pos_ia_a
prendizado_maquina\\Bases_de_teste')
barra = "\\"
##Note
setwd('/Users/MPPR/Documents/Pos_IA/pos_ia_aprendizado_maquina/Bases
_de_teste')
barra = "/"

dados <- read.csv(file = paste('biomassa', 'biomassa.csv', sep
=barra))
```

```
dados_novos <- read.csv(file =
paste('biomassa','biomassa_novos.csv',sep =barra))

### Cria arquivos de treino e teste
set.seed(728078902)
ind <- createDataPartition(dados$biomassa, p=0.80, list = FALSE)
treino <- dados[ind,]
teste <- dados[-ind,]

### Função R2
F_r2 <- function(observado,predito) {
  return (1 - (sum((predito-observado)^2) /
sum((predito-mean(observado))^2)))
}

### Função MAE
F_MAE <- function(observado,predito,base) {
  return(sum(abs(observado-predito)) / nrow(base))
}

### Função RMSE
F_RMSE <- function(observado,predito,base) {
  return( sqrt(sum((observado-predito)^2) / nrow(base)) )
}

### Função Syx
F_SYX <- function(observado,predito,base) {
  val1 = sum((observado-predito)^2)
  val2 = nrow(base) - (length(base)-1)
  return (sqrt(val1 / val2))
}

### Função Pearson
F_PEARSON <- function(observado,predito) {
  val1 = sum((observado-mean(observado)) * (predito-mean(predito)))
  val2 = sqrt(sum((observado-mean(observado))^2))
  val3 = sqrt(sum((predito-mean(predito))^2))
  return (val1 / (val2 * val3))
}

##### KNN
set.seed(728078902)
```

```
tuneGrid <- expand.grid(k = c(1,3,5,7,9))
knn <- train(biomassa ~ ., data = treino, method = "knn",
             tuneGrid=tuneGrid)

knn
predict.knn <- predict(knn, teste)

F_r2(teste$biomassa,predict.knn)
F_SYX(teste$biomassa,predict.knn,teste)
F_PEARSON(teste$biomassa,predict.knn)
F_RMSE(teste$biomassa,predict.knn,teste)
F_MAE(teste$biomassa,predict.knn,teste)
##### KNN

##### RNA
set.seed(728078902)
rna <- train(biomassa~., data=treino, method="nnet", linout=T,
             trace=FALSE)
rna
predict.rna <- predict(rna, teste)

F_r2(teste$biomassa,predict.rna)
F_SYX(teste$biomassa,predict.rna,teste)
F_PEARSON(teste$biomassa,predict.rna)
F_RMSE(teste$biomassa,predict.rna,teste)
F_MAE(teste$biomassa,predict.rna,teste)

### CV
set.seed(728078902)
control <- trainControl(method = "cv", number = 10)
rna_cv <- train(biomassa~., data=treino, method="nnet",
               trainControl=control, linout=T, trace=F)
rna_cv
predict.rna_cv <- predict(rna_cv, teste)

F_r2(teste$biomassa,predict.rna_cv)
F_SYX(teste$biomassa,predict.rna_cv,teste)
F_PEARSON(teste$biomassa,predict.rna_cv)
F_RMSE(teste$biomassa,predict.rna_cv,teste)
F_MAE(teste$biomassa,predict.rna_cv,teste)
```

```
###Parametrização
set.seed(728078902)
tuneGrid <- expand.grid(size = seq(from = 1, to = 10, by = 1), decay
= seq(from = 0.1, to = 0.9, by = 0.3))
rna_par <- train(biomassa~., data=treino, method="nnet",
trainControl=control, tuneGrid=tuneGrid, linout=T, MaxNWts=10000,
maxit=2000, trace=F)
rna_par
predict.rna_par <- predict(rna_par, teste)

F_r2(teste$biomassa,predict.rna_par)
F_SYX(teste$biomassa,predict.rna_par,teste)
F_PEARSON(teste$biomassa,predict.rna_par)
F_RMSE(teste$biomassa,predict.rna_par,teste)
F_MAE(teste$biomassa,predict.rna_par,teste)

##### RNA

##### SVN
set.seed(728078902)
svm <- train(biomassa~., data=treino, method="svmRadial")
svm
predict.svm <- predict(svm, teste)

F_r2(teste$biomassa,predict.svm)
F_SYX(teste$biomassa,predict.svm,teste)
F_PEARSON(teste$biomassa,predict.svm)
F_RMSE(teste$biomassa,predict.svm,teste)
F_MAE(teste$biomassa,predict.svm,teste)

#### Cross-validation SVM
set.seed(728078902)
ctrl <- trainControl(method = "cv", number = 10)
svm_cv <- train(biomassa~., data=treino, method="svmRadial",
trControl=ctrl)
svm_cv
predict.svm_cv <- predict(svm_cv, teste)
```

```
F_r2(teste$biomassa,predict.svm_cv)
F_SYX(teste$biomassa,predict.svm_cv,teste)
F_PEARSON(teste$biomassa,predict.svm_cv)
F_RMSE(teste$biomassa,predict.svm_cv,teste)
F_MAE(teste$biomassa,predict.svm_cv,teste)

#### Parametrização
set.seed(728078902)
tuneGrid = expand.grid(C=c(1, 2, 10, 50, 100), sigma=c(.01, .015,
0.2))
svm_par <- train(biomassa~., data=treino, method="svmRadial",
trControl=ctrl, tuneGrid=tuneGrid)
svm_par
predict.svm_par <- predict(svm_par, teste)

F_r2(teste$biomassa,predict.svm_par)
F_SYX(teste$biomassa,predict.svm_par,teste)
F_PEARSON(teste$biomassa,predict.svm_par)
F_RMSE(teste$biomassa,predict.svm_par,teste)
F_MAE(teste$biomassa,predict.svm_par,teste)

##### SVN

##### Random Forest
set.seed(728078902)
rf <- train(biomassa~.,data=treino,method="rf")
rf
predict.rf <- predict(rf,teste)

F_r2(teste$biomassa,predict.rf)
F_SYX(teste$biomassa,predict.rf,teste)
F_PEARSON(teste$biomassa,predict.rf)
F_RMSE(teste$biomassa,predict.rf,teste)
F_MAE(teste$biomassa,predict.rf,teste)

###Cross Validation
set.seed(728078902)
ctrl <-trainControl(method="cv",number=10)
rf_cv <- train(biomassa~.,data=treino,method="rf",trControl=ctrl)
```

```
rf_cv
predict.rf_cv <- predict(rf_cv, teste)

F_r2(teste$biomassa, predict.rf_cv)
F_SYX(teste$biomassa, predict.rf_cv, teste)
F_PEARSON(teste$biomassa, predict.rf_cv)
F_RMSE(teste$biomassa, predict.rf_cv, teste)
F_MAE(teste$biomassa, predict.rf_cv, teste)

###Parametrização
set.seed(728078902)
tuneGrid = expand.grid(mtry=c(2,5,7,9))
rf_par <-
train(biomassa~, data=treino, method="rf", trControl=ctrl, tuneGrid=tuneGrid)
rf_par
predict.rf_par <- predict(rf_par, teste)

F_r2(teste$biomassa, predict.rf_par)
F_SYX(teste$biomassa, predict.rf_par, teste)
F_PEARSON(teste$biomassa, predict.rf_par)
F_RMSE(teste$biomassa, predict.rf_par, teste)
F_MAE(teste$biomassa, predict.rf_par, teste)
##### Random Forest

##### Novos Casos
dados_novos$biomassa<-NULL
predict.melhor_caso<-predict(rna_par, dados_novos)
dados_novos <- cbind(dados_novos, predict.melhor_caso)
View(dados_novos)
##### Novos Casos

##### Gráfico de Resíduos

resid = ((teste$biomassa - predict.rna_par)/teste$biomassa) * 100

plot(resid ~ predict.rna_par,
      xlab="Valor estimado",
      ylab="Resíduos (%)",
```




UFPR – Universidade Federal do Paraná
Setor de Educação Profissional e Tecnológica
Especialização em Inteligência Artificial Aplicada
Turma 2022



Disciplina: Laboratório de IA – Prof Jaime Wojciechowski

```
col=2)  
abline(h=0,col=4)
```

```
##### Gráfico de Resíduos
```

Admissão

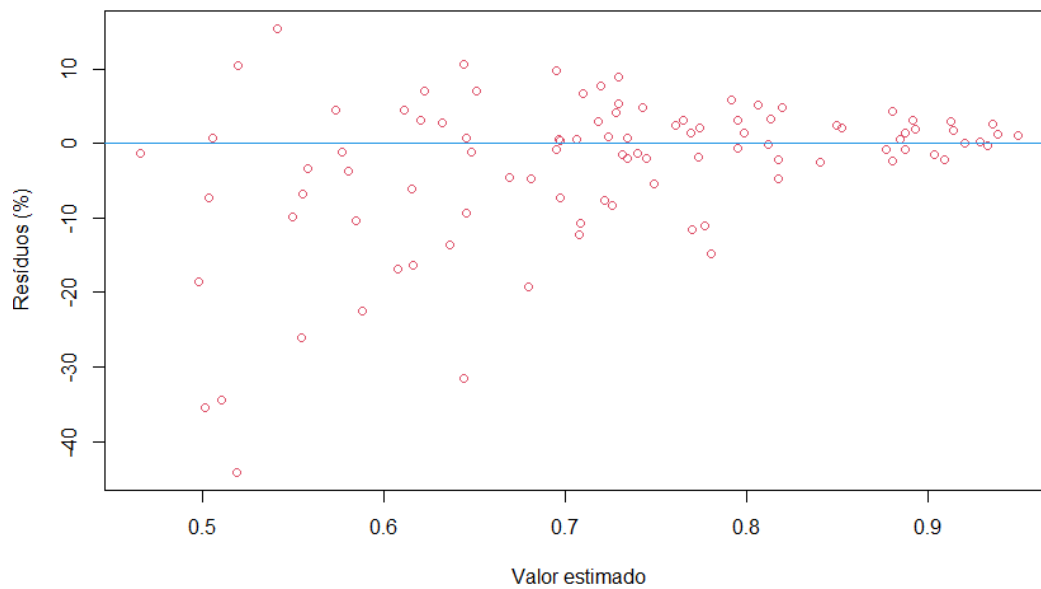
Técnica	Parâmetro	R2	Syx	Pearson	Rmse	MAE
RF – CV	mtry=2	0.8241163	0.05481691	0.9456885	0.05253186	0.03858572
RF – Melhor modelo	mtry=2	0.8168844	0.05553285	0.9445869	0.05321795	0.03869255
RF – Hold-out	mtry=2	0.8164555	0.05567193	0.9443351	0.05335123	0.03900982
RNA – Melhor modelo	size=9 decay=0.1	0.7875726	0.06358605	0.9139389	0.06093545	0.04526597
SVM – Melhor modelo	C=50 Sigma=0.01	0.7839405	0.06064641	0.9368963	0.05811835	0.04245389
SVM – CV	C=1 Sigma=0.1235567	0.7660959	0.0616459	0.9346271	0.05907618	0.04328504
RNA – CV	size=3 decay=0.1	0.7275993	0.06623201	0.9108705	0.06347112	0.04886074
RNA – Hold-out	size=3 decay=0.1	0.7275993	0.06623201	0.9108705	0.06347112	0.04886074
SVM – Hold-out	C=0.5 Sigma=0.1235567	0.7189279	0.06570313	0.9282732	0.06296429	0.04505394
KNN	k=7	0.40189	0.08460298	0.8537346	0.08107629	0.06357143

Melhor modelo: Random Forest com Cross Validation

Novos casos:

Serial.No.	GRE.Score	TOEFL.Score	University.Rating	SOP	LOR	CGPA	Research	predict.melhor_caso
1	337	118	4	4.5	4.5	9.65	1	0.9257554
2	324	107	4	4.0	4.5	8.87	1	0.7510095
3	316	104	3	3.0	3.5	8.00	1	0.6772650

Gráfico de resíduos:



Código (em R):

```
library(mlbench)
library(caret)
library(mice)
library(Metrics)

##Maquina MP
setwd('C:\\Users\\escnetto\\Documents\\Estudos\\Pos_IA_UFPR\\pos_ia_a
prendizado_maquina\\Bases_de_teste')
barra = "\\"
##Note
setwd('/Users/MPPR/Documents/Pos_IA/pos_ia_aprendizado_maquina/Bases
_de_teste')
barra = "/"

dados <- read.csv(file = paste('admissao','admissao.csv',sep
=barra))
dados_novos <- read.csv(file =
paste('admissao','admissao_novos.csv',sep =barra))
```

```
### Cria arquivos de treino e teste
set.seed(728078902)
ind <- createDataPartition(dados$ChanceOfAdmit, p=0.80, list =
FALSE)
treino <- dados[ind,]
teste <- dados[-ind,]

### Função R2
F_r2 <- function(observado,predito) {
  return (1 - (sum((predito-observado)^2) /
sum((predito-mean(observado))^2)))
}

### Função MAE
F_MAE <- function(observado,predito,base) {
  return(sum(abs(observado-predito)) / nrow(base))
}

### Função RMSE
F_RMSE <- function(observado,predito,base) {
  return( sqrt(sum((observado-predito)^2) / nrow(base)) )
}

### Função Syx
F_SYX <- function(observado,predito,base) {
  val1 = sum((observado-predito)^2)
  val2 = nrow(base) - (length(base)-1)
  return (sqrt(val1 / val2))
}

### Função Pearson
F_PEARSON <- function(observado,predito) {
  val1 = sum((observado-mean(observado)) * (predito-mean(predito)))
  val2 = sqrt(sum((observado-mean(observado))^2))
  val3 = sqrt(sum((predito-mean(predito))^2))
  return (val1 / (val2 * val3))
}

##### KNN
set.seed(728078902)
tuneGrid <- expand.grid(k = c(1,3,5,7,9))
knn <- train(ChanceOfAdmit ~ ., data = treino, method =
```

```
"knn", tuneGrid=tuneGrid)
knn
predict.knn <- predict(knn, teste)

F_r2(teste$ChanceOfAdmit, predict.knn)
F_SYX(teste$ChanceOfAdmit, predict.knn, teste)
F_PEARSON(teste$ChanceOfAdmit, predict.knn)
F_RMSE(teste$ChanceOfAdmit, predict.knn, teste)
F_MAE(teste$ChanceOfAdmit, predict.knn, teste)
##### KNN

##### RNA
set.seed(728078902)
rna <- train(ChanceOfAdmit~., data=treino, method="nnet", linout=T,
trace=FALSE)
rna
predict.rna <- predict(rna, teste)

### Mostra as métricas
F_r2(teste$ChanceOfAdmit, predict.rna)
F_SYX(teste$ChanceOfAdmit, predict.rna, teste)
F_PEARSON(teste$ChanceOfAdmit, predict.rna)
F_RMSE(teste$ChanceOfAdmit, predict.rna, teste)
F_MAE(teste$ChanceOfAdmit, predict.rna, teste)

### CV
set.seed(728078902)
control <- trainControl(method = "cv", number = 10)
rna_cv <- train(ChanceOfAdmit~., data=treino, method="nnet",
trainControl=control, linout=T, trace=F)
rna_cv
predict.rna_cv <- predict(rna_cv, teste)
F_r2(teste$ChanceOfAdmit, predict.rna_cv)
F_SYX(teste$ChanceOfAdmit, predict.rna_cv, teste)
F_PEARSON(teste$ChanceOfAdmit, predict.rna_cv)
F_RMSE(teste$ChanceOfAdmit, predict.rna_cv, teste)
F_MAE(teste$ChanceOfAdmit, predict.rna_cv, teste)

###Parametrização
```

```
set.seed(728078902)
tuneGrid <- expand.grid(size = seq(from = 1, to = 10, by = 1), decay
= seq(from = 0.1, to = 0.9, by = 0.3))
rna_par <- train(ChanceOfAdmit~., data=treino, method="nnet",
trainControl=control, tuneGrid=tuneGrid, linout=T, MaxNWts=10000,
maxit=2000, trace=F)
rna_par
predict.rna_par <- predict(rna_par, teste)

F_r2(teste$ChanceOfAdmit,predict.rna_par)
F_SYX(teste$ChanceOfAdmit,predict.rna_par,teste)
F_PEARSON(teste$ChanceOfAdmit,predict.rna_par)
F_RMSE(teste$ChanceOfAdmit,predict.rna_par,teste)
F_MAE(teste$ChanceOfAdmit,predict.rna_par,teste)

##### RNA

##### SVN
set.seed(728078902)
svm <- train(ChanceOfAdmit~., data=treino, method="svmRadial")
svm
predict.svm <- predict(svm, teste)

F_r2(teste$ChanceOfAdmit,predict.svm)
F_SYX(teste$ChanceOfAdmit,predict.svm,teste)
F_PEARSON(teste$ChanceOfAdmit,predict.svm)
F_RMSE(teste$ChanceOfAdmit,predict.svm,teste)
F_MAE(teste$ChanceOfAdmit,predict.svm,teste)

#### Cross-validation SVM
set.seed(728078902)
ctrl <- trainControl(method = "cv", number = 10)
svm_cv <- train(ChanceOfAdmit~., data=treino, method="svmRadial",
trControl=ctrl)
svm_cv
predict.svm_cv<- predict(svm_cv, teste)

F_r2(teste$ChanceOfAdmit,predict.svm_cv)
F_SYX(teste$ChanceOfAdmit,predict.svm_cv,teste)
```

```
F_PEARSON(teste$ChanceOfAdmit,predict.svm_cv)
F_RMSE(teste$ChanceOfAdmit,predict.svm_cv,teste)
F_MAE(teste$ChanceOfAdmit,predict.svm_cv,teste)

#### Parametrização
set.seed(728078902)
tuneGrid = expand.grid(C=c(1, 2, 10, 50, 100), sigma=c(.01, .015,
0.2))
svm_par <- train(ChanceOfAdmit~., data=treino, method="svmRadial",
trControl=ctrl, tuneGrid=tuneGrid)
svm_par
predict.svm_par <- predict(svm_par, teste)

F_r2(teste$ChanceOfAdmit,predict.svm_par)
F_SYX(teste$ChanceOfAdmit,predict.svm_par,teste)
F_PEARSON(teste$ChanceOfAdmit,predict.svm_par)
F_RMSE(teste$ChanceOfAdmit,predict.svm_par,teste)
F_MAE(teste$ChanceOfAdmit,predict.svm_par,teste)
##### SVN

##### Random Forest
set.seed(728078902)
rf <- train(ChanceOfAdmit~.,data=treino,method="rf")
rf
predict.rf <- predict(rf,teste)

F_r2(teste$ChanceOfAdmit,predict.rf)
F_SYX(teste$ChanceOfAdmit,predict.rf,teste)
F_PEARSON(teste$ChanceOfAdmit,predict.rf)
F_RMSE(teste$ChanceOfAdmit,predict.rf,teste)
F_MAE(teste$ChanceOfAdmit,predict.rf,teste)

##Cross Validation
set.seed(728078902)
ctrl <- trainControl(method="cv",number = 10)
rf_cv <-
train(ChanceOfAdmit~.,data=treino,method="rf",trControl=ctrl)
rf_cv
predict.rf_cv <- predict(rf_cv,teste)
```

```
F_r2(teste$ChanceOfAdmit,predict.rf_cv)
F_SYX(teste$ChanceOfAdmit,predict.rf_cv,teste)
F_PEARSON(teste$ChanceOfAdmit,predict.rf_cv)
F_RMSE(teste$ChanceOfAdmit,predict.rf_cv,teste)
F_MAE(teste$ChanceOfAdmit,predict.rf_cv,teste)

##parametrização
set.seed(728078902)
tuneGrid = expand.grid(mtry=c(1,2,5,7,9,11))
rf_par <-
train(ChanceOfAdmit~.,data=treino,method="rf",trControl=ctrl,tuneGrid=tuneGrid)
rf_par
predict.rf_par <- predict(rf_par,teste)

F_r2(teste$ChanceOfAdmit,predict.rf_par)
F_SYX(teste$ChanceOfAdmit,predict.rf_par,teste)
F_PEARSON(teste$ChanceOfAdmit,predict.rf_par)
F_RMSE(teste$ChanceOfAdmit,predict.rf_par,teste)
F_MAE(teste$ChanceOfAdmit,predict.rf_par,teste)

##### Random Forest

##### Novos Casos
dados_novos$Chance.ofAdmit <- NULL
predict.melhor_caso <- predict(rf_cv, dados_novos)
dados_novos <- cbind(dados_novos, predict.melhor_caso)
View(dados_novos)
##### Novos Casos

##### Gráfico de Resíduos

resid = ((teste$ChanceOfAdmit - predict.rf_cv)/teste$ChanceOfAdmit)
* 100

plot(resid ~ predict.rf_cv,
      xlab="Valor estimado",
```




UFPR – Universidade Federal do Paraná
Setor de Educação Profissional e Tecnológica
Especialização em Inteligência Artificial Aplicada
Turma 2022



Disciplina: Laboratório de IA – Prof Jaime Wojciechowski

```
ylab="Resíduos (%)",  
col=2)  
abline(h=0,col=4)  
  
##### Gráfico de Resíduos
```

Alunos do Ensino Médio

Técnica	Parâmetro	R2	Syx	Pearson	Rmse	MAE
RF – CV	mtry=41	0.8334 009	2.32628 8	0.92251 78	1.7783 79	1.0396 68
RF – Melhor modelo	mtry=31	0.8323 078	2.28531 4	0.92451 65	1.7470 55	1.0533 75
RF – Hold-out	mtry=21	0.8032 457	2.34501 7	0.92053 41	1.7926 97	1.1269 71
KNN	k=9	0.7770 519	2.43299 8	0.91587 4	1.8599 56	1.2873 76
RNA – Melhor modelo	size=10 decay=0.7	0.7746 389	2.73990 4	0.89548 31	2.0945 77	1.4420 59
SVM – Melhor modelo	C=2 Sigma=0.01	0.6611 407	2.82201	0.88370 92	2.1573 44	1.3587 75
SVM – Hold-out	C=1 Sigma=0.01366 968	0.6083 002	2.89608 4	0.88052 19	2.2139 72	1.4304 68
SVM – CV	C=1 Sigma=0.01366 968	0.6083 002	2.89608 4	0.88052 19	2.2139 72	1.4304 68
RNA – Hold-out	size=5 decay=0.1	0.5494 744	3.83015 7	0.78875 67	2.9280 43	1.8768
RNA – CV	size=5 decay=0.1	0.5494 744	3.83015 7	0.78875 67	2.9280 43	1.8768

Melhor modelo: Random Forest com Cross Validation

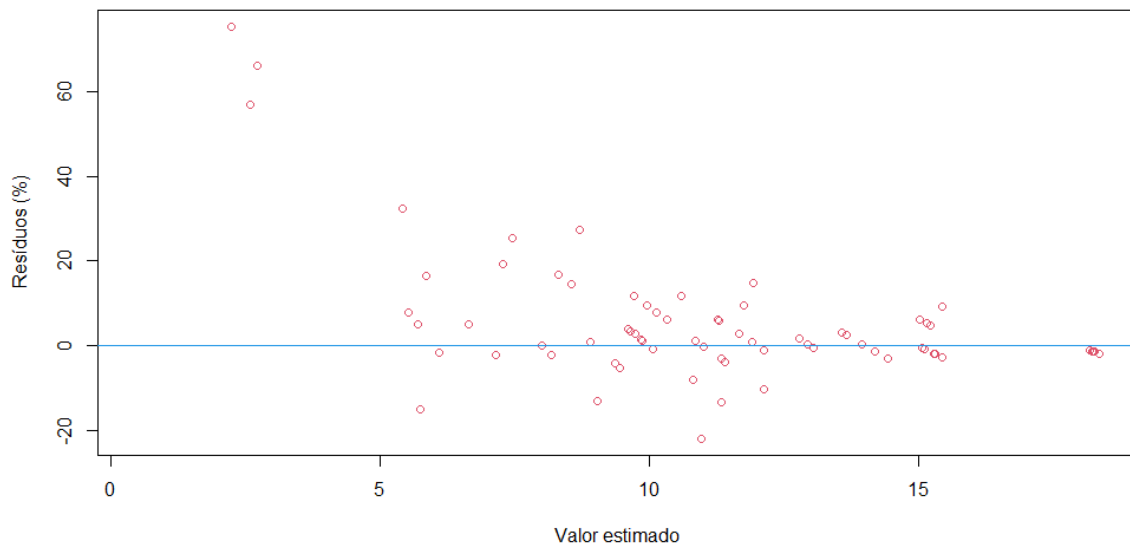
Novos casos:

school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reason	guardian	traveltime	
1	GP	2	16	R	GT3	A	4	4	at_home	teacher	course	mother	2
2	GP	1	17	U	GT3	T	1	1	at_home	other	course	father	1
3	GP	1	18	U	LE3	T	1	1	at_home	other	other	mother	1

studytime	failures	schoolsup	famsup	paid	activities	nursery	higher	internet	romantic	famrel	freetime
1	0	no	no	no	no	yes	yes	no	no	4	3
2	0	no	no	no	yes	no	yes	yes	yes	5	3
2	3	yes	yes	yes	yes	yes	yes	yes	no	4	3

goout	Dalc	Walc	health	absences	G1	G2	predict.melhor_modelo
1	1	1	3	6	2	3	5.935367
3	1	1	3	4	15	15	15.256133
2	2	3	3	10	10	10	9.656967

Gráfico de resíduos:



Código (em R):

```
library(mlbench)
library(caret)
library(mice)
library(Metrics)

##Maquina MP
setwd('C:\\Users\\escneto\\Documents\\Estudos\\Pos_IA_UFPR\\pos_ia_a
prendizado_maquina\\Bases_de_teste')
barra = "\\"
##Note
setwd('/Users/MPPR/Documents/Pos_IA/pos_ia_aprendizado_maquina/Bases
_de_teste')
barra = "/"

dados <- read.csv(file = paste('alunos', 'alunos.csv', sep =barra))
dados_novos <- read.csv(file = paste('alunos', 'alunos_novos.csv', sep
=barra))

### Cria arquivos de treino e teste
```

```
set.seed(728078902)
ind <- createDataPartition(dados$G3, p=0.80, list = FALSE)
treino <- dados[ind,]
teste <- dados[-ind,]

### Função R2
F_r2 <- function(observado,predito) {
  return (1 - (sum((predito-observado)^2) /
sum((predito-mean(observado))^2)))
}

### Função MAE
F_MAE <- function(observado,predito,base) {
  return(sum(abs(observado-predito)) / nrow(base))
}

### Função RMSE
F_RMSE <- function(observado,predito,base) {
  return( sqrt(sum((observado-predito)^2) / nrow(base)) )
}

### Função Syx
F_SYX <- function(observado,predito,base) {
  val1 = sum((observado-predito)^2)
  val2 = nrow(base) - (length(base)-1)
  return (sqrt(val1 / val2))
}

### Função Pearson
F_PEARSON <- function(observado,predito) {
  val1 = sum((observado-mean(observado)) * (predito-mean(predito)))
  val2 = sqrt(sum((observado-mean(observado))^2))
  val3 = sqrt(sum((predito-mean(predito))^2))
  return (val1 / (val2 * val3))
}

##### KNN
set.seed(728078902)
tuneGrid <- expand.grid(k = c(1,3,5,7,9))
knn <- train(G3 ~ ., data = treino, method = "knn",
             tuneGrid=tuneGrid)

knn
predict.knn <- predict(knn, teste)
```

```
### Mostra as métricas
F_r2(teste$G3,predict.knn)
F_SYX(teste$G3,predict.knn,teste)
F_PEARSON(teste$G3,predict.knn)
F_RMSE(teste$G3,predict.knn,teste)
F_MAE(teste$G3,predict.knn,teste)
##### KNN

##### RNA
set.seed(728078902)
rna <- train(G3~., data=treino, method="nnet", linout=T,
trace=FALSE)
rna
predict.rna <- predict(rna, teste)

### Mostra as métricas
F_r2(teste$G3,predict.rna)
F_SYX(teste$G3,predict.rna,teste)
F_PEARSON(teste$G3,predict.rna)
F_RMSE(teste$G3,predict.rna,teste)
F_MAE(teste$G3,predict.rna,teste)

### CV
set.seed(728078902)
control <- trainControl(method = "cv", number = 10)
rna_cv <- train(G3~., data=treino, method="nnet",
trainControl=control, linout=T, trace=F)
rna_cv
predict.rna_cv <- predict(rna_cv, teste)

F_r2(teste$G3,predict.rna_cv)
F_SYX(teste$G3,predict.rna_cv,teste)
F_PEARSON(teste$G3,predict.rna_cv)
F_RMSE(teste$G3,predict.rna_cv,teste)
F_MAE(teste$G3,predict.rna_cv,teste)

###Parametrização
set.seed(728078902)
```

```
tuneGrid <- expand.grid(size = seq(from = 1, to = 10, by = 1), decay
= seq(from = 0.1, to = 0.9, by = 0.3))
rna_par <- train(G3~., data=treino, method="nnet",
trainControl=control, tuneGrid=tuneGrid, linout=T, MaxNWts=10000,
maxit=2000, trace=F)
rna_par
predict.rna_par <- predict(rna_par, teste)

F_r2(teste$G3,predict.rna_par)
F_SYX(teste$G3,predict.rna_par,teste)
F_PEARSON(teste$G3,predict.rna_par)
F_RMSE(teste$G3,predict.rna_par,teste)
F_MAE(teste$G3,predict.rna_par,teste)
##### RNA

##### SVN
set.seed(728078902)
svm <- train(G3~., data=treino, method="svmRadial")
svm
predict.svm <- predict(svm, teste)

F_r2(teste$G3,predict.svm)
F_SYX(teste$G3,predict.svm,teste)
F_PEARSON(teste$G3,predict.svm)
F_RMSE(teste$G3,predict.svm,teste)
F_MAE(teste$G3,predict.svm,teste)

#### Cross-validation SVM
set.seed(728078902)
ctrl <- trainControl(method = "cv", number = 10)
svm_cv <- train(G3~., data=treino, method="svmRadial",
trControl=ctrl)
svm_cv
predict.svm_cv <- predict(svm_cv, teste)

F_r2(teste$G3,predict.svm_cv)
F_SYX(teste$G3,predict.svm_cv,teste)
F_PEARSON(teste$G3,predict.svm_cv)
F_RMSE(teste$G3,predict.svm_cv,teste)
```

```
F_MAE(teste$G3,predict.svm_cv,teste)

#### Parametrização
set.seed(728078902)
tuneGrid = expand.grid(C=c(1, 2, 10, 50, 100), sigma=c(.01, .015,
0.2))
svm_par <- train(G3~., data=treino, method="svmRadial",
tuneGrid=tuneGrid)
svm_par
predict.svm_par <- predict(svm_par, teste)

F_r2(teste$G3,predict.svm_par)
F_SYX(teste$G3,predict.svm_par,teste)
F_PEARSON(teste$G3,predict.svm_par)
F_RMSE(teste$G3,predict.svm_par,teste)
F_MAE(teste$G3,predict.svm_par,teste)

##### SVN

##### Random Forest
set.seed(728078902)
rf <- train(G3~.,data=treino,method="rf")
rf
predict.rf <- predict(rf,teste)

F_r2(teste$G3,predict.rf)
F_SYX(teste$G3,predict.rf,teste)
F_PEARSON(teste$G3,predict.rf)
F_RMSE(teste$G3,predict.rf,teste)
F_MAE(teste$G3,predict.rf,teste)

##Cross Validation
set.seed(728078902)
ctrl <- trainControl(method="cv",number = 10)
rf_cv <- train(G3~.,data=treino,method="rf",trControl=ctrl)
rf_cv
predict.rf_cv <- predict(rf_cv,teste)

F_r2(teste$G3,predict.rf_cv)
```

```
F_SYX(teste$G3,predict.rf_cv,teste)
F_PEARSON(teste$G3,predict.rf_cv)
F_RMSE(teste$G3,predict.rf_cv,teste)
F_MAE(teste$G3,predict.rf_cv,teste)

##Varios mtry
set.seed(728078902)
tuneGrid = expand.grid(mtry=c(21,31,41,43,45))
rf_par <-
train(G3~.,data=treino,method="rf",trControl=ctrl,tuneGrid=tuneGrid)
rf_par
predict.rf_par <- predict(rf_par,teste)

F_r2(teste$G3,predict.rf_par)
F_SYX(teste$G3,predict.rf_par,teste)
F_PEARSON(teste$G3,predict.rf_par)
F_RMSE(teste$G3,predict.rf_par,teste)
F_MAE(teste$G3,predict.rf_par,teste)

predict.rf <- predict(rf, dados_novos)
dados_novos <- cbind(dados_novos, predict.rf)
predict.rf2 <- predict(rf2, dados_novos)
dados_novos <- cbind(dados_novos, predict.rf2)
predict.rf3 <- predict(rf3, dados_novos)
dados_novos <- cbind(dados_novos, predict.rf3)
##### Random Forest

##### Novos Casos
dados_novos$G3 <- NULL
predict.melhor_modelo <- predict(rf_cv,dados_novos)
dados_novos <- cbind(dados_novos, predict.melhor_modelo)
View(dados_novos)

##### Novos Casos

##### Gráfico de Resíduos

resid = ((teste$G3 - predict.rf_cv)/teste$G3) * 100
```




UFPR – Universidade Federal do Paraná
Setor de Educação Profissional e Tecnológica
Especialização em Inteligência Artificial Aplicada
Turma 2022



Disciplina: Laboratório de IA – Prof Jaime Wojciechowski

```
plot(resid ~ predict.rf_cv,  
     xlab="Valor estimado",  
     ylab="Resíduos (%)",  
     col=2)  
abline(h=0,col=4)
```

Gráfico de Resíduos

AGRUPAMENTO

Íris

Lista de Clusters gerados:

K-means clustering with 3 clusters of sizes 48, 50, 52

Cluster means:

Petal.Length Petal.Width

1	5.595833	2.037500
2	1.462000	0.246000
3	4.269231	1.342308

10 primeiras linhas do arquivo com o cluster correspondente:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species	irisCluster\$cluster
1	5.1	3.5	1.4	0.2	setosa	2
2	4.9	3.0	1.4	0.2	setosa	2
3	4.7	3.2	1.3	0.2	setosa	2
4	4.6	3.1	1.5	0.2	setosa	2
5	5.0	3.6	1.4	0.2	setosa	2
6	5.4	3.9	1.7	0.4	setosa	2
7	4.6	3.4	1.4	0.3	setosa	2
8	5.0	3.4	1.5	0.2	setosa	2
9	4.4	2.9	1.4	0.2	setosa	2
10	4.9	3.1	1.5	0.1	setosa	2

Lista de comandos emitidos no RStudio:

```
data("iris")
dados <- iris
set.seed(728078902)
irisCluster <- kmeans(iris,3)
irisCluster
resultado <- cbind(dados,irisCluster$cluster)
resultado
```

Móveis

Lista de Clusters gerados:

K-modes clustering with 10 clusters of sizes 75, 48, 26, 80, 20, 13, 88, 25, 44, 35

Cluster modes:

	categoria	cor
1	Cozinha / Balcao / Balcao para pia em madeira	Nature/Branco
2	Cozinha / Armario / Armario aereo de parede em madeira	Carvalle
3	Sala de Estar / Racks para TV / Racks para TV	Capuccino/Preto
4	Cozinha / Balcao / Balcao em madeira	Carvalle/Branco
5	Quarto Juvenil / Beliche e treliche	Tabaco
6	Quarto do Bebe / Guarda-roupas / Guarda-roupas com portas de bater	Branco Linho Gelo/Rosa/Azul
7	Cozinha / Balcao / Balcao em madeira	Nature
8	Sala de Estar / Mesas para sala / Mesas de centro	Branco Linho Gelo/Azul/Branco Linho
9	Cozinha / Armario / Armario em Aco	Branco
10	Quarto do Bebe / Mini-camas	Branco
	estilo	
1	moderno	
2	moderno	
3	tradicional	
4	moderno	
5	tradicional	
6	tradicional	
7	moderno	
8	tradicional	
9	tradicional	
10	moderno	

10 primeiras linhas do arquivo com o cluster correspondente:

	cor	estilo	moveisCluster\$cluster
1	Branco Linho Gelo/Azul/Branco Linho	tradicional	8
2	Branco Linho Gelo/Rosa/Azul	tradicional	6
3	Branco Linho Gelo/Azul/Branco Linho	tradicional	8
4	Branco Linho Gelo/Rosa/Azul	tradicional	6
5	Branco Linho Gelo/Azul/Branco Linho	tradicional	8
6	Branco Linho Gelo/Rosa/Azul	tradicional	6
7	Branco Linho Gelo/Rosa/Azul/Capuccino	tradicional	3
8	Branco Linho Gelo/Capuccino/Azul/Rosa	tradicional	5
9	Branco Linho Gelo/Azul/Branco Linho	tradicional	8
10	Branco Linho Gelo/Rosa/Azul	tradicional	6

Lista de comandos emitidos no RStudio:

```
library(klaR)
dados <- read.csv(file = 'moveis.csv')
set.seed(728078902)
moveisCluster <- kmodes(dados,10,iter.max = 10, weighted = FALSE)
moveisCluster
resultado <- cbind(dados,moveisCluster$cluster)
resultado
```

Câncer de Mama

Lista de Clusters gerados:

K-means clustering with 2 clusters of sizes 235, 464

Cluster means:

	Cl.thickness	Cell.size	Cell.shape	Marg.adhesion	Epith.c.size	Bare.nuclei	Bl.cromatin	Normal.nucleoli	Mitoses
1	7.153191	6.765957	6.706383	5.706383	5.442553	7.906383	6.093617	6.063830	2.536170
2	3.032328	1.295259	1.435345	1.338362	2.088362	1.321121	2.092672	1.247845	1.109914

10 primeiras linhas do arquivo com o cluster correspondente:

	Cl.thickness	Cell.size	Cell.shape	Marg.adhesion	Epith.c.size	Bare.nuclei	Bl.cromatin	Normal.nucleoli	Mitoses
1	5	1	1	1	2	1	3	1	1
2	5	4	4	5	7	10	3	2	1
3	3	1	1	1	2	2	3	1	1
4	6	8	8	1	3	4	3	7	1
5	4	1	1	3	2	1	3	1	1
6	8	10	10	8	7	10	9	7	1
7	1	1	1	1	2	10	3	1	1
8	2	1	2	1	2	1	3	1	1
9	2	1	1	1	2	1	1	1	5
10	4	2	1	1	2	1	2	1	1

	Class	cancerCluster\$cluster
1	benign	2
2	benign	1
3	benign	2
4	benign	1
5	benign	2
6	malignant	1
7	benign	2
8	benign	2
9	benign	2
10	benign	2

Lista de comandos emitidos no RStudio:

```
library(klaR)
dados <- read.csv(file = 'cancer_mama.csv')
set.seed(728078902)
cancerCluster <- kmeans(dados[2:10],2)
cancerCluster
resultado <- cbind(dados,cancerCluster$cluster)
resultado[2:12]
```

Veículo

Lista de Clusters gerados:

K-means clustering with 4 clusters of sizes 129, 360, 169, 188

Cluster means:

	Comp	Circ	DCirc	RadRa	PrAxisRa	MaxLRa	ScatRa	Elong	PrAxisRect	MaxLRect	ScVarMaxis
1	98.13953	46.75194	93.37209	198.7907	64.09302	8.604651	187.7054	35.01550	21.91473	150.3101	207.7364
2	89.22500	42.79444	75.72778	158.3833	62.92222	8.811111	153.6306	43.51667	19.30833	143.8278	176.3222
3	87.86391	38.23077	63.94083	133.4497	57.27811	6.674556	130.3373	51.67456	17.79290	133.1893	151.8402
4	104.37234	53.48404	102.84043	200.5798	61.66489	9.776596	219.6277	30.39362	24.61702	167.7128	232.1383
	ScVarmaxis	RaGyr	SkewMaxis	Skewmaxis	Kurtmaxis	KurtMaxis	HollRa				
1	532.1318	183.4109	68.27132	6.046512	13.51938	193.1318	200.0775				
2	351.1694	165.8639	73.60278	6.022222	10.97500	188.3750	194.8972				
3	252.0237	140.6686	72.42012	6.272189	11.62722	188.4852	193.6331				
4	715.4628	216.2500	73.19149	7.377660	15.95213	187.5213	195.7872				

10 primeiras linhas do arquivo com o cluster correspondente:

	Comp	Circ	DCirc	RadRa	PrAxisRa	MaxLRa	ScatRa	Elong	PrAxisRect	MaxLRect	ScVarMaxis	ScVarmaxis	RaGyr	SkewMaxis
1	95	48	83	178	72	10	162	42	20	159	176	379	184	70
2	91	41	84	141	57	9	149	45	19	143	170	330	158	72
3	104	50	106	209	66	10	207	32	23	158	223	635	220	73
4	93	41	82	159	63	9	144	46	19	143	160	309	127	63
5	85	44	70	205	103	52	149	45	19	144	241	325	188	127
6	107	57	106	172	50	6	255	26	28	169	280	957	264	85
7	97	43	73	173	65	6	153	42	19	143	176	361	172	66
8	90	43	66	157	65	9	137	48	18	146	162	281	164	67
9	86	34	62	140	61	7	122	54	17	127	141	223	112	64
10	93	44	98	197	62	11	183	36	22	146	202	505	152	64
	Skewmaxis	Kurtmaxis	KurtMaxis	HollRa	Class	veiculoCluster	\$cluster							
1	6	16	187	197	van		2							
2	9	14	189	199	van		2							
3	14	9	188	196	saab		4							
4	6	10	199	207	van		2							
5	9	11	180	183	bus		2							
6	5	9	181	183	bus		4							
7	13	1	200	204	bus		2							
8	3	3	193	202	van		3							
9	2	14	200	208	van		3							
10	4	14	195	204	saab		1							

Lista de comandos emitidos no RStudio:

```
dados <- read.csv(file = 'veiculos.csv')
set.seed(728078902)
veiculoCluster <- kmeans(dados[2:19],4)
veiculoCluster
resultado <- cbind(dados,veiculoCluster$cluster)
resultado[2:21]
```

Banco

Lista de Clusters gerados:

K-modes clustering with 2 clusters of sizes 124, 176

Cluster modes:

	age	job	marital	education	default	balance	housing	loan	
1	34	management	married	tertiary	no	0	no	no	
2	32	bluecollar	married	secondary	no	0	yes	no	

10 primeiras linhas do arquivo com o cluster correspondente:

	age	job	marital	education	default	balance	housing	loan	y	bancoCluster\$cluster
1	30	unemployed	married	primary	no	1787	no	no	no	1
2	33	services	married	secondary	no	4789	yes	yes	no	2
3	35	management	single	tertiary	no	1350	yes	no	no	1
4	30	management	married	tertiary	no	1476	yes	yes	no	1
5	59	bluecollar	married	secondary	no	0	yes	no	no	2
6	35	management	single	tertiary	no	747	no	no	no	1
7	36	selfemployed	married	tertiary	no	307	yes	no	no	1
8	39	technician	married	secondary	no	147	yes	no	no	2
9	41	entrepreneur	married	tertiary	no	221	yes	no	no	1
10	43	services	married	primary	no	88	yes	yes	no	2

Lista de comandos emitidos no RStudio:

```
library(klaR)
dados <- read.csv(file = 'banco.csv')
set.seed(728078902)
bancoCluster <- kmodes(dados[0:8],2,iter.max = 10, weighted = FALSE)
bancoCluster
resultado <- cbind(dados,bancoCluster$cluster)
resultado
```

REGRAS DE ASSOCIAÇÃO

Lista de Compras

Confiança=0.7, Suporte=0.01

Regras = 21

1 regra com 1 item

6 regras com 2 itens

10 regras com 3 itens

4 regras com 4 itens

	lhs	rhs	support	confidence	coverage	lift	count
[1]	{}	=> {leite}	0.75	0.75	1.00	1.0	3
[2]	{ovos}	=> {leite}	0.25	1.00	0.25	1.3	1
[3]	{cafe}	=> {pao}	0.25	1.00	0.25	2.0	1
[4]	{cafe}	=> {bolacha}	0.25	1.00	0.25	2.0	1
[5]	{suco}	=> {leite}	0.50	1.00	0.50	1.3	2
[6]	{pao}	=> {bolacha}	0.50	1.00	0.50	2.0	2
[7]	{bolacha}	=> {pao}	0.50	1.00	0.50	2.0	2
[8]	{cafe, pao}	=> {bolacha}	0.25	1.00	0.25	2.0	1
[9]	{bolacha, cafe}	=> {pao}	0.25	1.00	0.25	2.0	1
[10]	{pao, suco}	=> {bolacha}	0.25	1.00	0.25	2.0	1

Confiança=0.1, Suporte=0.001

Regras = 43

6 regra com 1 item

18 regras com 2 itens

15 regras com 3 itens

4 regras com 4 itens

	lhs	rhs	support	confidence	coverage	lift	count
[1]	{}	=> {ovos}	0.25	0.25	1.00	1.00	1
[2]	{}	=> {cafe}	0.25	0.25	1.00	1.00	1
[3]	{}	=> {suco}	0.50	0.50	1.00	1.00	2
[4]	{}	=> {pao}	0.50	0.50	1.00	1.00	2
[5]	{}	=> {bolacha}	0.50	0.50	1.00	1.00	2
[6]	{}	=> {leite}	0.75	0.75	1.00	1.00	3
[7]	{ovos}	=> {leite}	0.25	1.00	0.25	1.33	1
[8]	{leite}	=> {ovos}	0.25	0.33	0.75	1.33	1
[9]	{cafe}	=> {pao}	0.25	1.00	0.25	2.00	1
[10]	{pao}	=> {cafe}	0.25	0.50	0.50	2.00	1

Confiança=0.9, Suporte=0.5

Regras = 3

3 regra com 2 item

	lhs	rhs	support	confidence	coverage	lift	count
[1]	{suco}	=> {leite}	0.5	1	0.5	1.3	2
[2]	{pao}	=> {bolacha}	0.5	1	0.5	2.0	2
[3]	{bolacha}	=> {pao}	0.5	1	0.5	2.0	2

Lista de comandos:

```
dados <-  
read.transactions(file="lista_compras.csv",format="basket",sep=";")  
inspect(dados[1:4])  
  
### Confiança=0.7, Suporte=0.01  
set.seed(728078902)  
rules<-apriori(dados,parameter =  
list(supp=0.01,conf=0.7,target="rules"))  
summary(rules)  
inspect(rules)  
  
### Confiança=0.1, Suporte=0.001  
set.seed(728078902)  
rules2<-apriori(dados,parameter =  
list(supp=0.001,conf=0.1,target="rules"))  
summary(rules2)  
inspect(rules2)  
  
### Confiança=0.9, Suporte=0.5  
set.seed(728078902)  
rules3<-apriori(dados,parameter =  
list(supp=0.5,conf=0.9,target="rules"))  
summary(rules3)  
inspect(rules3)
```


Musculação

Confiança=0.7, Suporte=0.01

Regras = 156

1 regra com 1 item

23 regras com 2 itens

56 regras com 3 itens

55 regras com 4 itens

21 regras com 5 itens

lhs	rhs	support	confidence	coverage	lift	count
[1] {}	=> {LegPress}	0.808	0.81	1.000	1.00	21
[2] {Crucifixo}	=> {Afundo}	0.077	1.00	0.077	2.89	2
[3] {Crucifixo}	=> {Gemeos}	0.077	1.00	0.077	1.53	2
[4] {Crucifixo}	=> {LegPress}	0.077	1.00	0.077	1.24	2
[5] {Adutor}	=> {Agachamento}	0.115	1.00	0.115	3.25	3
[6] {Adutor}	=> {LegPress}	0.115	1.00	0.115	1.24	3
[7] {Flexor}	=> {Esteira}	0.077	1.00	0.077	2.17	2
[8] {Flexor}	=> {Extensor}	0.077	1.00	0.077	2.00	2
[9] {Flexor}	=> {Bicicleta}	0.077	1.00	0.077	1.86	2
[10] {Flexor}	=> {LegPress}	0.077	1.00	0.077	1.24	2

Confiança=0.1, Suporte=0.001

Regras = 381

9 regra com 1 item

64 regras com 2 itens

140 regras com 3 itens

128 regras com 4 itens

40 regras com 5 itens

lhs	rhs	support	confidence	coverage	lift	count
[1] {}	=> {Adutor}	0.115	0.12	1.000	1.00	3
[2] {}	=> {Agachamento}	0.308	0.31	1.000	1.00	8
[3] {}	=> {Afundo}	0.346	0.35	1.000	1.00	9
[4] {}	=> {AgachamentoSmith}	0.385	0.38	1.000	1.00	10
[5] {}	=> {Esteira}	0.462	0.46	1.000	1.00	12
[6] {}	=> {Extensor}	0.500	0.50	1.000	1.00	13
[7] {}	=> {Bicicleta}	0.538	0.54	1.000	1.00	14
[8] {}	=> {Gemeos}	0.654	0.65	1.000	1.00	17
[9] {}	=> {LegPress}	0.808	0.81	1.000	1.00	21
[10] {Crucifixo}	=> {Afundo}	0.077	1.00	0.077	2.89	2

Confiança=0.9, Suporte=0.05

Regras = 79

14 regras com 2 itens

32 regras com 3 itens

27 regras com 4 itens

6 regras com 5 itens

	lhs	rhs	support	confidence	coverage
[1]	{Crucifixo}	=> {Afundo}	0.077	1.00	0.077
[2]	{Crucifixo}	=> {Gemeos}	0.077	1.00	0.077
[3]	{Crucifixo}	=> {LegPress}	0.077	1.00	0.077
[4]	{Adutor}	=> {Agachamento}	0.115	1.00	0.115
[5]	{Adutor}	=> {LegPress}	0.115	1.00	0.115
[6]	{Flexor}	=> {Esteira}	0.077	1.00	0.077
[7]	{Flexor}	=> {Extensor}	0.077	1.00	0.077
[8]	{Flexor}	=> {Bicicleta}	0.077	1.00	0.077
[9]	{Flexor}	=> {LegPress}	0.077	1.00	0.077
[10]	{Agachamento}	=> {LegPress}	0.308	1.00	0.308
[11]	{Afundo}	=> {Gemeos}	0.346	1.00	0.346
[12]	{AgachamentoSmith}	=> {Extensor}	0.346	0.90	0.385
[13]	{Esteira}	=> {Extensor}	0.423	0.92	0.462
[14]	{Extensor}	=> {Bicicleta}	0.462	0.92	0.500
[15]	{Afundo, Crucifixo}	=> {Gemeos}	0.077	1.00	0.077
[16]	{Crucifixo, Gemeos}	=> {Afundo}	0.077	1.00	0.077

Lista de comandos:

```
dados <-
read.transactions(file="musculacao.csv",format="basket",sep=";")
inspect(dados[1:4])

### Confiança=0.7, Suporte=0.01
set.seed(728078902)
rules<-apriori(dados,parameter =
list(supp=0.01,conf=0.7,target="rules"))
summary(rules)
inspect(rules)

### Confiança=0.1, Suporte=0.001
set.seed(728078902)
rules2<-apriori(dados,parameter =
list(supp=0.001,conf=0.1,target="rules"))
summary(rules2)
inspect(rules2)

### Confiança=0.9, Suporte=0.5
set.seed(728078902)
rules3<-apriori(dados,parameter =
list(supp=0.05,conf=0.9,target="rules"))
summary(rules3)
inspect(rules3)
```