

DIABETES

Carlos da Costa

2025-09-22

Obtenção dos dados

```
diabetes <- read.csv(  
  file = "file/diabetes.csv"  
)  
  
head(diabetes[1:6])
```

##	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI
## 1	6	148	72	35	0	33.6
## 2	1	85	66	29	0	26.6
## 3	8	183	64	0	0	23.3
## 4	1	89	66	23	94	28.1
## 5	0	137	40	35	168	43.1
## 6	5	116	74	0	0	25.6

Preparação dos dados

```
diabetes$Outcome <- as.factor(diabetes$Outcome)  
  
library(dplyr)  
  
diabetes2 <- diabetes %>%  
  filter(Insulin <= 250)
```

Construção do modelo

Divisão dos dados

```
library(caTools)  
  
set.seed(123)  
index = sample.split(diabetes2$Pregnancies, SplitRatio = .70)
```

```
train = subset(diabetes2, index == TRUE)
test = subset(diabetes2, index == FALSE)
```

Construção do modelo

Treinamento

```
library(caret)
library(e1071)
set.seed(321)

modelo2 <- train(
  Outcome ~., data = train, method = "knn",
  tuneGrid = expand.grid(k = c(1:20)))

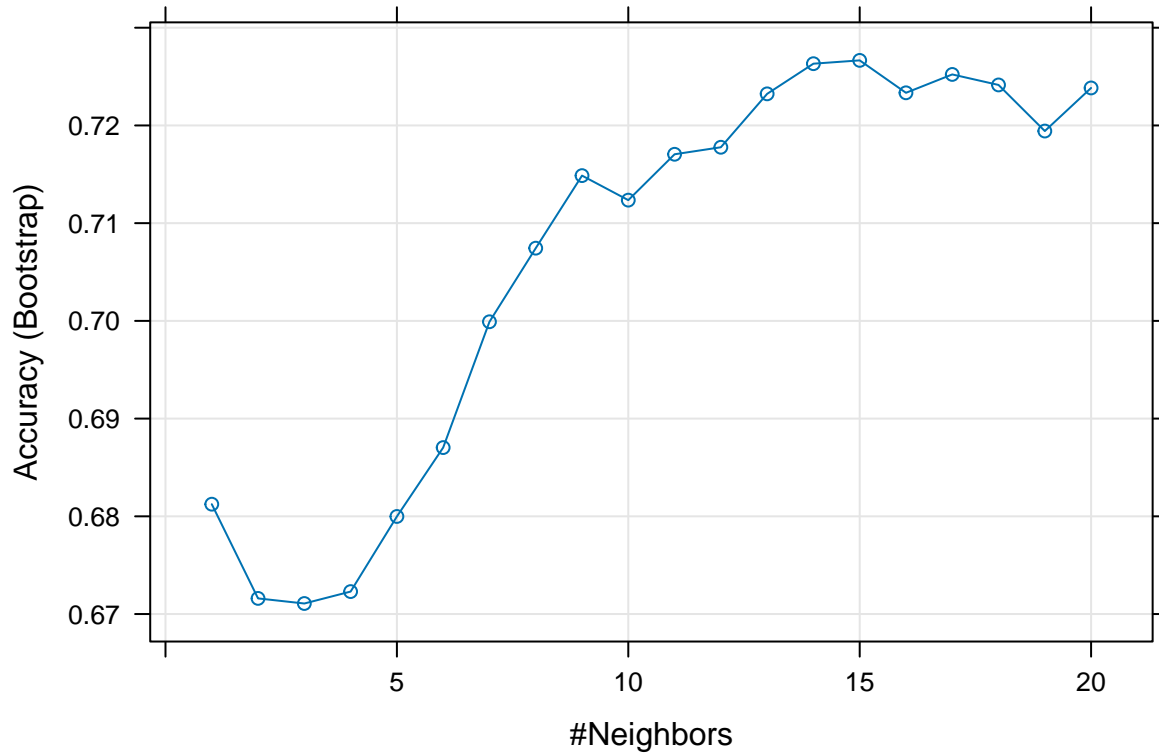
modelo2$results
```

##	k	Accuracy	Kappa	AccuracySD	KappaSD
## 1	1	0.6812417	0.2672857	0.02508823	0.05209127
## 2	2	0.6716012	0.2472091	0.02580538	0.05823044
## 3	3	0.6710795	0.2489375	0.03062290	0.06158122
## 4	4	0.6722990	0.2491852	0.03564545	0.07270520
## 5	5	0.6799853	0.2612326	0.03497202	0.06965532
## 6	6	0.6870351	0.2709889	0.03054408	0.06678610
## 7	7	0.6999105	0.2964678	0.03056224	0.06822750
## 8	8	0.7074421	0.3111849	0.02730385	0.05911386
## 9	9	0.7148655	0.3248247	0.02833120	0.06396979
## 10	10	0.7123562	0.3190123	0.02767569	0.05331421
## 11	11	0.7170498	0.3242597	0.02662759	0.05107199
## 12	12	0.7177685	0.3244747	0.02838302	0.05985968
## 13	13	0.7232339	0.3333408	0.02569841	0.05320402
## 14	14	0.7263198	0.3394999	0.02475220	0.05764762
## 15	15	0.7266595	0.3382824	0.02343778	0.05097264
## 16	16	0.7233392	0.3290788	0.02802407	0.06015891
## 17	17	0.7252203	0.3322175	0.02981879	0.06615667
## 18	18	0.7241440	0.3293330	0.02531860	0.05483174
## 19	19	0.7194279	0.3155865	0.02362044	0.04931024
## 20	20	0.7238362	0.3249042	0.02649810	0.05972954

```
modelo2$bestTune
```

```
##      k
## 15 15
```

```
plot(modelo2)
```



Avaliando o modelo

```
predicoes <- predict(modelo2,test)
```

```
predicoes
```

```
## [1] 1 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1
## [38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 1 0 1 1 0 0 0
## [75] 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 1 0 0 0 1 1 1 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0
## [112] 1 0 0 0 0 1 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1
## [149] 0 0 1 0 0 0 1 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0
## [186] 0 0 0 1 0 0 0 1 1 1 0 0 1 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0
## Levels: 0 1
```

```
?caret::confusionMatrix
confusionMatrix(predicoes, test$Outcome)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0    1
##           0 127  35
```

```

##          1  14  38
##
##          Accuracy : 0.771
##          95% CI : (0.7088, 0.8255)
##    No Information Rate : 0.6589
##    P-Value [Acc > NIR] : 0.0002402
##
##          Kappa : 0.4527
##
##    McNemar's Test P-Value : 0.0042747
##
##          Sensitivity : 0.9007
##          Specificity : 0.5205
##          Pos Pred Value : 0.7840
##          Neg Pred Value : 0.7308
##          Prevalence : 0.6589
##          Detection Rate : 0.5935
##          Detection Prevalence : 0.7570
##          Balanced Accuracy : 0.7106
##
##          'Positive' Class : 0
##

```