Previsão da saúde de um cavalo

Projeto final da disciplina de Data Mining

16 de Outubro de 2019

Aluno: Gabriel Taranto Pereira Magrina Ferreres

gabrieltaranto91@gmail.com

Professora: Manoela Kohler

Sumário

1 Introdução

Este projeto tem por finalidade coletar dados, analisá-los e prever se um cavalo pode sobreviver ou não baseado nas condições médicas passadas.

2 Análise exploratória, missing values e atributos desnecessários

Nesta seção, verifico se existem *outliers*, faço o tratamento dos dados faltantes (missing values) e retiro atributos desnecessários.

Ao analisarmos o *box plot* dos atributos, foi possível observar muitos *outliers*, conforme esperado em uma base sem nenhum tipo de tratamento. Os *outliers* foram removidos um a um da base de dados.

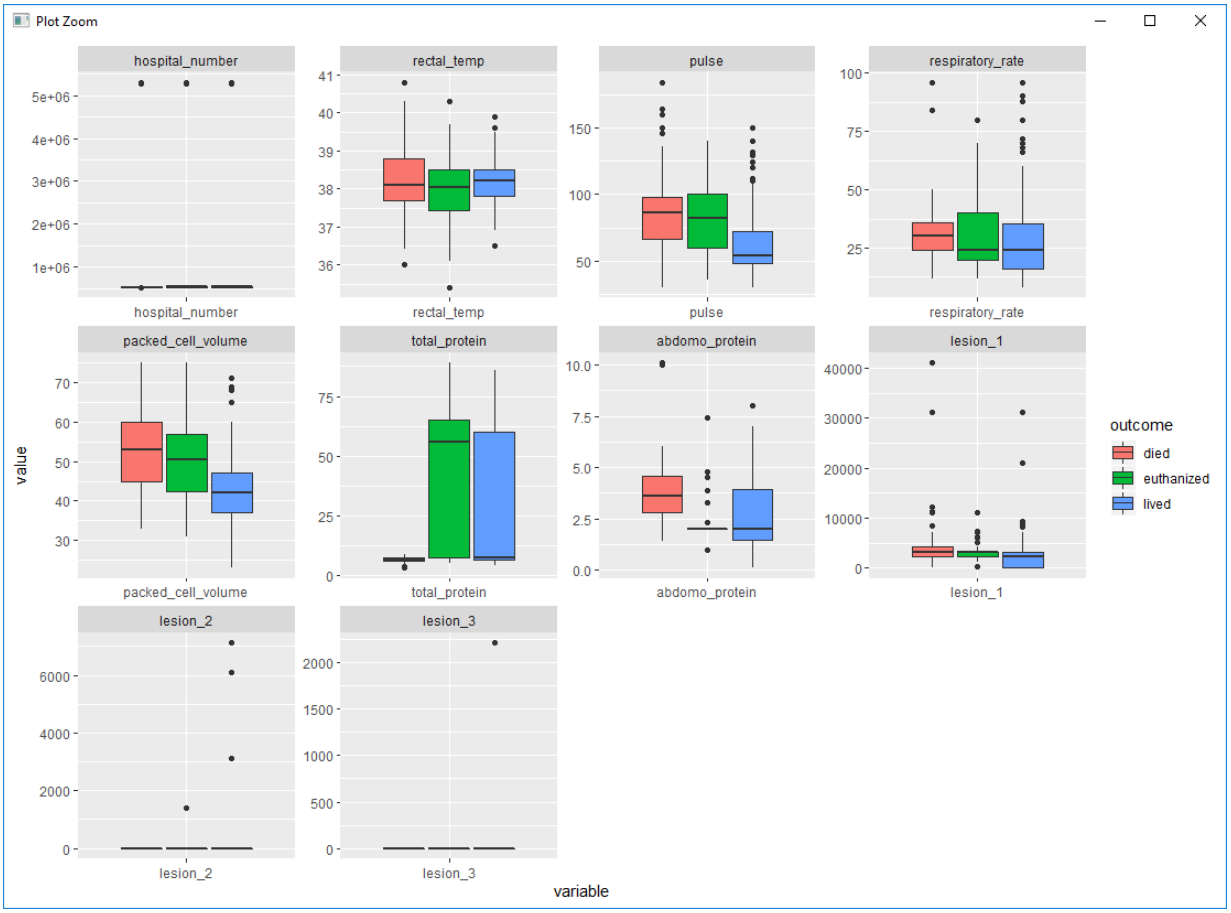

Figura 1: *Box Plot* dos atributos.

Após o tratamento de *outliers*, é possível perceber uma grande diferença nos gráficos, mostrando dados mais próximos uns dos outros e, portanto, melhores para o estudo. Os atributos *hospital_number*, *rectal_temp*, *respiratory_rate*, *abdomo_protein* e *lesion_1* foram tratados, como observado no gráfico a seguir.

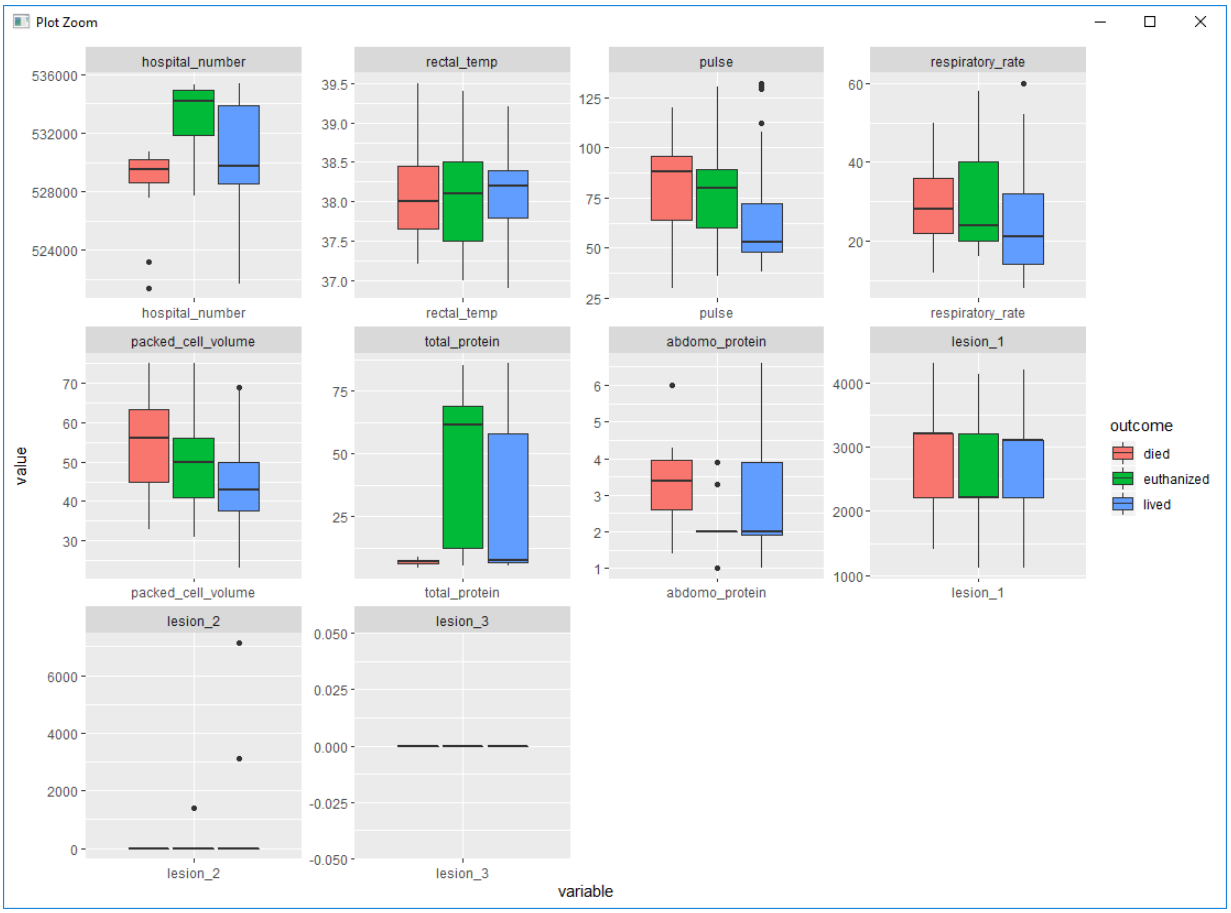

Figura 2: *Box Plot* após o tratamento de *outliers*.

Ao utilizar o comando *summary*, percebi que existem muitos atributos com *missing values* e talvez seja necessário retirar alguns destes atributos, pois estes não contribuem muito para a classificação final.

Um atributo retirado foi o *nasogastric_reflux_ph*, pois possuía 246 *missing values* de 299 observações. Além do atributo *nasogastric_reflux_ph*, os atributos *cp_data* e *lesion_3* também foram removidos, pois o pdf de explicação da base de dados indica que o primeiro não é um atributo relevante para o caso e o segundo contém somente zeros após o tratamento de *outliers*.

```
Console  Terminal  Jobs
C:/Users/GabrielTaranto/Desktop/Trabalho_DM/
> summary(horses_train)
 surgery      age       hospital_number   rectal_temp       pulse         respiratory_rate
 no : 40   adult:146   Min.   :521399   Min.   :36.9   Min.   : 30.00   Min.   : 8.00
 yes:107   young:  1   1st Qu.:528625   1st Qu.:37.9   1st Qu.: 48.00   1st Qu.:20.00
                       Median :529812   Median :38.1   Median : 66.00   Median :24.00
                       Mean   :530654   Mean   :38.1   Mean   : 70.75   Mean   :26.33
                       3rd Qu.:533908   3rd Qu.:38.3   3rd Qu.: 88.00   3rd Qu.:31.00
                       Max.   :535392   Max.   :39.5   Max.   :132.00   Max.   :60.00
 temp_of_extremities  peripheral_pulse      mucous_membrane capillary_refill_time
 cold  :16           absent   : 7      bright_pink  : 9     3         :  1
 cool  :89           increased: 2      bright_red   :14     less_3_sec:104
 normal:30           normal   :50      dark_cyanotic:12     more_3_sec: 42
 warm  :12           reduced  :88      normal_pink  :27
                                       pale_cyanotic:30
                                       pale_pink    :55
          pain         peristalsis abdominal_distention    nasogastric_tube
 alert       :12   absent     :48   moderate:66          none       : 33
 depressed   :23   hypermotile: 9   none    :25          significant: 14
 extreme_pain:29   hypomotile :84   severe  :25          slight     :100
 mild_pain   :59   normal     : 6   slight  :31
 severe_pain :24

    nasogastric_reflux nasogastric_reflux_ph rectal_exam_feces          abdomen
 less_1_liter:20       Min.   :1.000         absent   :102     distend_large:94
 more_1_liter:29       1st Qu.:4.300         decreased: 24     distend_small:33
 none        :98       Median :4.300         increased:  5     firm         : 3
                       Mean   :4.274         normal   : 16     normal       : 6
                       3rd Qu.:4.300                           other        :11
                       Max.   :7.500
 packed_cell_volume total_protein       abdomo_appearance abdomo_protein
 Min.   :23.00      Min.   : 4.50   clear        : 14      Min.   :1.000
 1st Qu.:41.00      1st Qu.: 6.70   cloudy       :109      1st Qu.:2.000
 Median :46.00      Median : 7.50   serosanguious: 24      Median :2.000
 Mean   :48.16      Mean   :22.35                          Mean   :2.231
 3rd Qu.:55.00      3rd Qu.:31.50                          3rd Qu.:2.000
 Max.   :75.00      Max.   :86.00                          Max.   :6.600
      outcome   surgical_lesion    lesion_1       lesion_2       lesion_3 cp_data
 died      :42   no : 33         Min.   :1111   Min.   :   0.00   Min.   :0   no :103
 euthanized:27   yes:114         1st Qu.:2208   1st Qu.:   0.00   1st Qu.:0   yes: 44
 lived     :78                   Median :3111   Median :   0.00   Median :0
                                 Mean   :2796   Mean   :  79.06   Mean   :0
                                 3rd Qu.:3205   3rd Qu.:   0.00   3rd Qu.:0
                                 Max.   :4300   Max.   :7111.00   Max.   :0
```

Figura 3: *Dataset* com os atributos *nasogastric_ reflux_ ph* e *cp_ data*.

```
Console  Terminal  Jobs
C:/Users/GabrielTaranto/Desktop/Trabalho_DM/
> drop <- c("nasogastric_reflux_ph", "cp_data", "lesion_3")
> horses_train = horses_train[,!(names(horses_train) %in% drop)]
> summary(horses_train)
 surgery      age       hospital_number   rectal_temp       pulse         respiratory_rate
 no : 40   adult:146   Min.   :521399   Min.   :36.9   Min.   : 30.00   Min.   : 8.00
 yes:107   young:  1   1st Qu.:528625   1st Qu.:37.9   1st Qu.: 48.00   1st Qu.:20.00
                       Median :529812   Median :38.1   Median : 66.00   Median :24.00
                       Mean   :530654   Mean   :38.1   Mean   : 70.75   Mean   :26.33
                       3rd Qu.:533908   3rd Qu.:38.3   3rd Qu.: 88.00   3rd Qu.:31.00
                       Max.   :535392   Max.   :39.5   Max.   :132.00   Max.   :60.00
 temp_of_extremities  peripheral_pulse      mucous_membrane capillary_refill_time
 cold  :16           absent   : 7      bright_pink  : 9     3         :  1
 cool  :89           increased: 2      bright_red   :14     less_3_sec:104
 normal:30           normal   :50      dark_cyanotic:12     more_3_sec: 42
 warm  :12           reduced  :88      normal_pink  :27
                                       pale_cyanotic:30
                                       pale_pink    :55
          pain         peristalsis abdominal_distention    nasogastric_tube
 alert       :12   absent     :48   moderate:66          none       : 33
 depressed   :23   hypermotile: 9   none    :25          significant: 14
 extreme_pain:29   hypomotile :84   severe  :25          slight     :100
 mild_pain   :59   normal     : 6   slight  :31
 severe_pain :24

    nasogastric_reflux rectal_exam_feces          abdomen   packed_cell_volume
 less_1_liter:20       absent   :102     distend_large:94   Min.   :23.00
 more_1_liter:29       decreased: 24     distend_small:33   1st Qu.:41.00
 none        :98       increased:  5     firm         : 3   Median :46.00
                       normal   : 16     normal       : 6   Mean   :48.16
                                         other        :11   3rd Qu.:55.00
                                                            Max.   :75.00
 total_protein       abdomo_appearance abdomo_protein       outcome   surgical_lesion
 Min.   : 4.50   clear        : 14     Min.   :1.000   died      :42   no : 33
 1st Qu.: 6.70   cloudy       :109     1st Qu.:2.000   euthanized:27   yes:114
 Median : 7.50   serosanguious: 24     Median :2.000   lived     :78
 Mean   :22.35                         Mean   :2.231
 3rd Qu.:31.50                         3rd Qu.:2.000
 Max.   :86.00                         Max.   :6.600
    lesion_1       lesion_2
 Min.   :1111   Min.   :   0.00
 1st Qu.:2208   1st Qu.:   0.00
 Median :3111   Median :   0.00
 Mean   :2796   Mean   :  79.06
 3rd Qu.:3205   3rd Qu.:   0.00
 Max.   :4300   Max.   :7111.00
```

Figura 4: *Dataset* sem os atributos *nasogastric_ reflux_ ph*, *cp_ data* e *lesion_ 3*.

Os *missing values* dos atributos numéricos foram substituídos pela mediana dos valores do atributo da base de treino, tanto na base de treino quanto na base de teste, enquanto que os atributos categóricos foram substituídos pelo mais frequente.

```
47  #substituir o mais frequente nos atributos categóricos
48  for (i in list_na){
49    horses_train[i] = na.replace(horses_train[i],names(which.max(table(horses_train[i]))
50  }
51
52  for (j in list_na){
53    horses_test[j] = na.replace(horses_test[j],names(which.max(table(horses_train[j]))))
54  }
55
56  summary(horses_train)
57  summary(horses_test)
58
```

Figura 5: Código da substituição pelo valor mais frequente.

3 Balanceamento dos Dados e processo para gerar as duas bases tratadas

Esta seção apresenta a normalização dos dados e a substituição dos valores *euthanized* do atributo *outcome* para *died*, pois é necessário fazer essa modificação caso utilize qualquer modelo de aprendizado que suporte apenas duas classes, como o SVM. A normalização foi realizada com o método *range* da função *preProcess*, ajustando todos os valores da base de treino e teste.

```
84  #normalizar os dados
85  normalizedTrain <- preProcess(horses_train, method = "range")
86  horses_train_normal <- predict(normalizedTrain, horses_train)
87
88  horses_test_normal <- predict(normalizedTrain, horses_test)
89
90  summary(horses_train_normal)
91  summary(horses_test_normal)
```

Figura 6: Normalização da base de treino e teste.

```
> summary(horses_train_normal)
 surgery      age        hospital_number   rectal_temp         pulse
 no : 40   adult:146   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
 yes:107   young:  1   1st Qu.:0.5164   1st Qu.:0.3846   1st Qu.:0.1765
                       Median :0.6012   Median :0.4615   Median :0.3529
                       Mean   :0.6614   Mean   :0.4626   Mean   :0.3995
                       3rd Qu.:0.8939   3rd Qu.:0.5385   3rd Qu.:0.5686
                       Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
 respiratory_rate temp_of_extremities  peripheral_pulse      mucous_membrane
 Min.   :0.0000   cold  :16           absent   : 7       bright_pink  : 9
 1st Qu.:0.2308   cool  :89           increased: 2       bright_red   :14
 Median :0.3077   normal:30           normal   :50       dark_cyanotic:12
 Mean   :0.3524   warm  :12           reduced  :88       normal_pink  :27
 3rd Qu.:0.4423                                         pale_cyanotic:30
 Max.   :1.0000                                         pale_pink    :55
 capillary_refill_time          pain         peristalsis abdominal_distention
 3         :  1        alert       :12   absent     :48   moderate:66
 less_3_sec:104        depressed   :23   hypermotile: 9   none    :25
 more_3_sec: 42        extreme_pain:29   hypomotile :84   severe  :25
                       mild_pain   :59   normal     : 6   slight  :31
                       severe_pain :24

    nasogastric_tube    nasogastric_reflux rectal_exam_feces          abdomen
 none       : 33     less_1_liter:20       absent   :102     distend_large:94
 significant: 14     more_1_liter:29       decreased: 24     distend_small:33
 slight     :100     none        :98       increased:  5     firm         : 3
                                           normal   : 16     normal       : 6
                                                             other        :11

 packed_cell_volume total_protein         abdomo_appearance abdomo_protein    outcome
 Min.   :0.0000     Min.   :0.00000   clear        : 14     Min.   :0.0000   died :69
 1st Qu.:0.3462     1st Qu.:0.02699   cloudy       :109     1st Qu.:0.1786   lived:78
 Median :0.4423     Median :0.03681   serosanguious: 24     Median :0.1786
 Mean   :0.4838     Mean   :0.21906                         Mean   :0.2199
 3rd Qu.:0.6154     3rd Qu.:0.33129                         3rd Qu.:0.1786
 Max.   :1.0000     Max.   :1.00000                         Max.   :1.0000
 surgical_lesion    lesion_1         lesion_2
 no : 33         Min.   :0.0000   Min.   :0.00000
 yes:114         1st Qu.:0.3438   1st Qu.:0.00000
                 Median :0.6272   Median :0.00000
                 Mean   :0.5285   Mean   :0.01112
                 3rd Qu.:0.6566   3rd Qu.:0.00000
                 Max.   :1.0000   Max.   :1.00000
```

Figura 7: Substituição dos valores *euthanized* por *died*.

4 Qui Quadrado e PCA

Nesta seção, utilizo Qui Quadrado e PCA para rankear os atributos e descobrir quais outros atributos podem ser removidos das bases de treino e teste. Os atributos 1 a 10 são, respectivamente, *hospital_number*, *rectal_temp*, *pulse*, *respiratory_rate*, *packed_cell_volume*, *total_protein*, *abdomo_protein*, *lesion_1*, *lesion_2* e *outcome*. Aplicando Qui Quadrado, percebo que existem muitos atributos com *p-value* maior que 0.05, o que indica não influência na saída (*outcome*).

```
Não influencia na saída! Verificar o atributo  1
Não influencia na saída! Verificar o atributo  2
Influencia na saída!
Não influencia na saída! Verificar o atributo  4
Não influencia na saída! Verificar o atributo  5
Não influencia na saída! Verificar o atributo  6
Não influencia na saída! Verificar o atributo  7
Influencia na saída!
Não influencia na saída! Verificar o atributo  9
Influencia na saída!
```

Figura 8: Qui Quadrado aplicado na base de treino.

Aplicando PCA, percebo que alguns atributos que não inlfuenciam na saída no teste de Qui Quadrado, também não influenciam na saída pelo PCA. Portanto, existem grandes chances de que os atributos *lesion_1* e *lesion_2* não influenciem no *outcome*. Logo, foram excluídos da base de treino e teste.

```
> summary(horses_train_num_normal.pca)
Importance of components:
                          PC1    PC2    PC3    PC4     PC5     PC6     PC7     PC8
Standard deviation     1.4150 1.3825 1.0451 0.9998 0.93133 0.89303 0.82248 0.63177
Proportion of Variance 0.2225 0.2124 0.1214 0.1111 0.09637 0.08861 0.07516 0.04435
Cumulative Proportion  0.2225 0.4348 0.5562 0.6673 0.76362 0.85223 0.92740 0.97174
                           PC9
Standard deviation     0.50428
Proportion of Variance 0.02826
Cumulative Proportion  1.00000
```

Figura 9: PCA aplicado na base de treino.

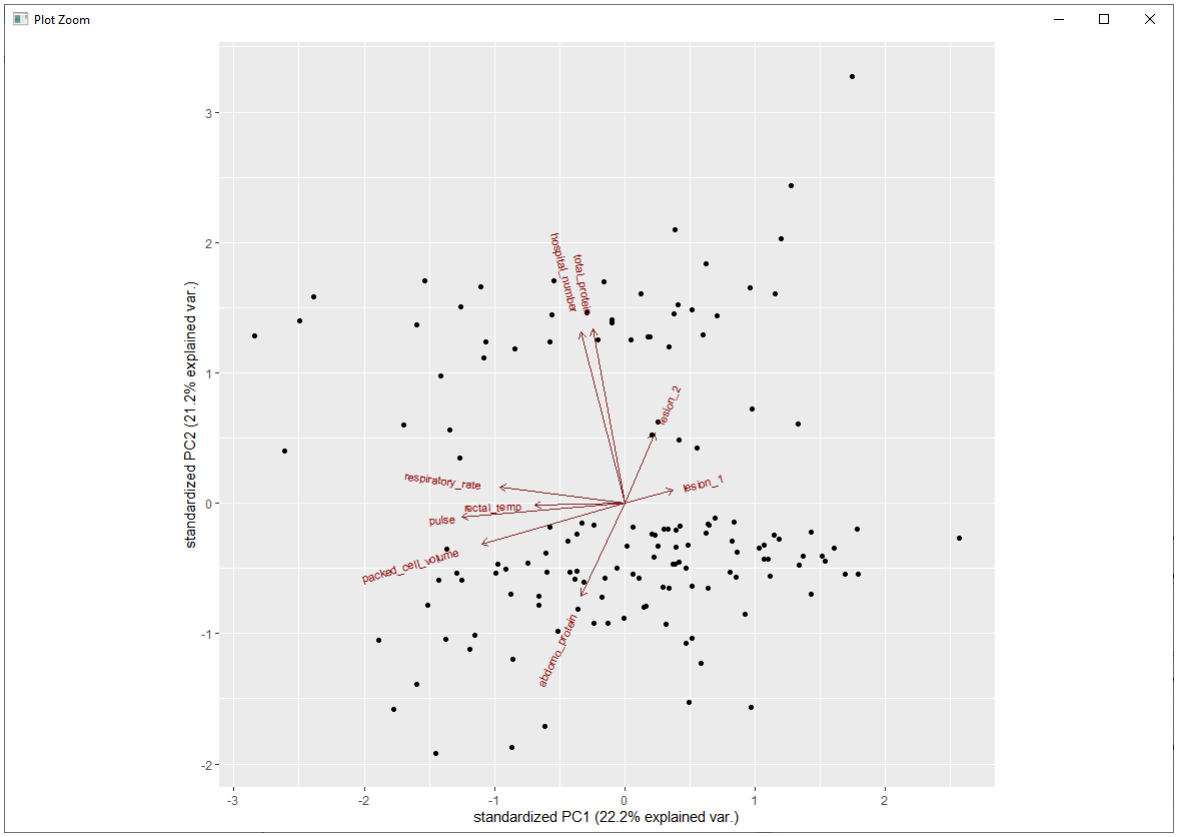

Figura 10: Gráfico dos atributos.

5 Testes de diferentes modelos matemáticos

5.1 Decision Tree

Ao aplicarmos o modelo de *Decision Tree* em nosso problema, obtivemos uma acurácia de 77,5% aproximadamente. Este valor não é muito bom, o que nos faz concluir que talvez o modelo de *Decision Tree* não seja o ideal para a classificação do nosso problema. Das 23 variáveis possíveis, somente 9 foram usadas, conforme figura a seguir.

```
> predictionsDTree <- predict(tree_model, horses_test_normal, type="class")
> table(predictionsDTree, horses_test_normal$outcome)

predictionsDTree died lived
           died    19     3
           lived   17    50
> accuracy = 1 - mean(predictionsDTree != horses_test_normal$outcome)
> accuracy
[1] 0.7752809
> summary(tree_model)

Classification tree:
tree(formula = outcome ~ ., data = horses_train_normal)
Variables actually used in tree construction:
[1] "pain"                  "temp_of_extremities"   "surgery"
[4] "abdominal_distention"  "capillary_refill_time" "hospital_number"
[7] "packed_cell_volume"    "rectal_temp"           "abdomo_appearance"
Number of terminal nodes:  16
Residual mean deviance:  0.5792 = 75.88 / 131
Misclassification error rate: 0.1361 = 20 / 147
```

Figura 11: Decision Tree.

Através da visualização da árvore, podemos identificar os atributos que contribuem para que o cavalo viva ou morra.

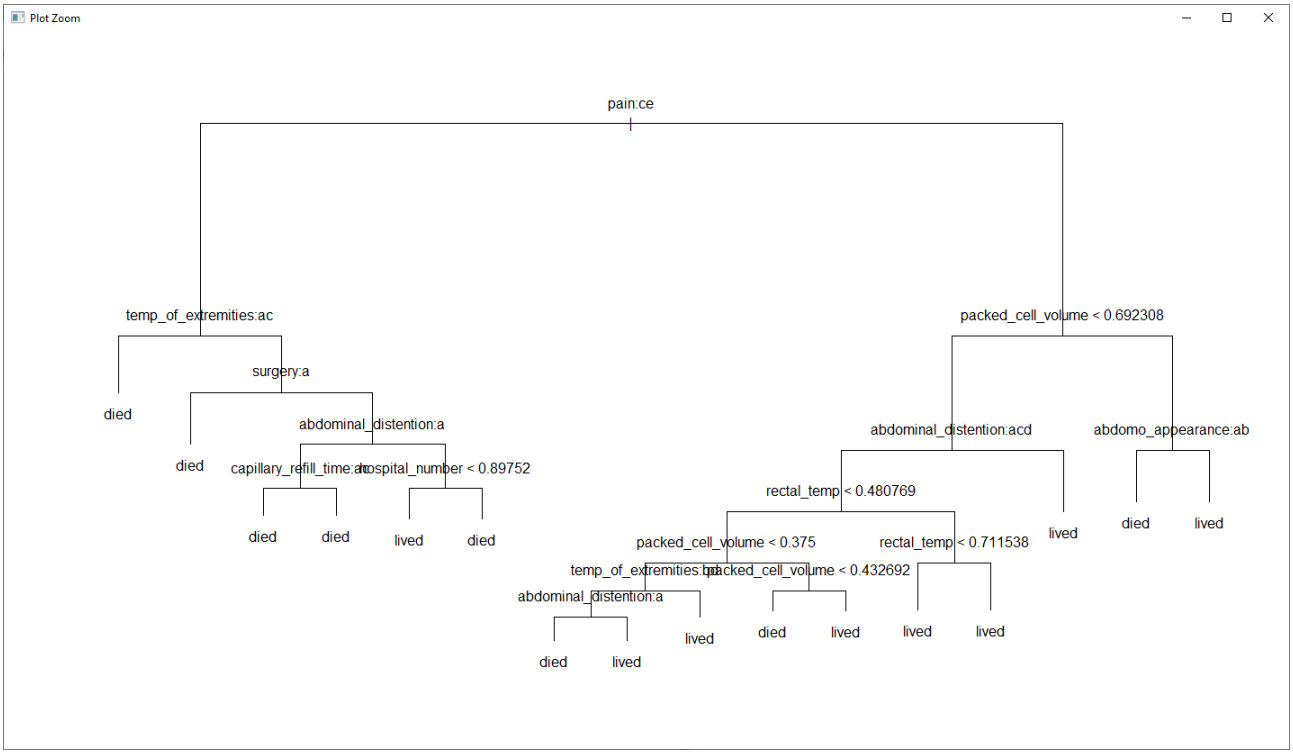

Figura 12: Estrutura da Decision Tree.

A interpretação dessa árvore é feita pelos gráficos de barras como este a seguir. Por exemplo, os valores do atributo *capillary_refill_time* são nomeados como *a*, *b* e *c* para *3*, *less_3_sec* e *more_3_sec*. Ou seja, todo o cavalo que possui valor desse atributo *3* vai morrer.

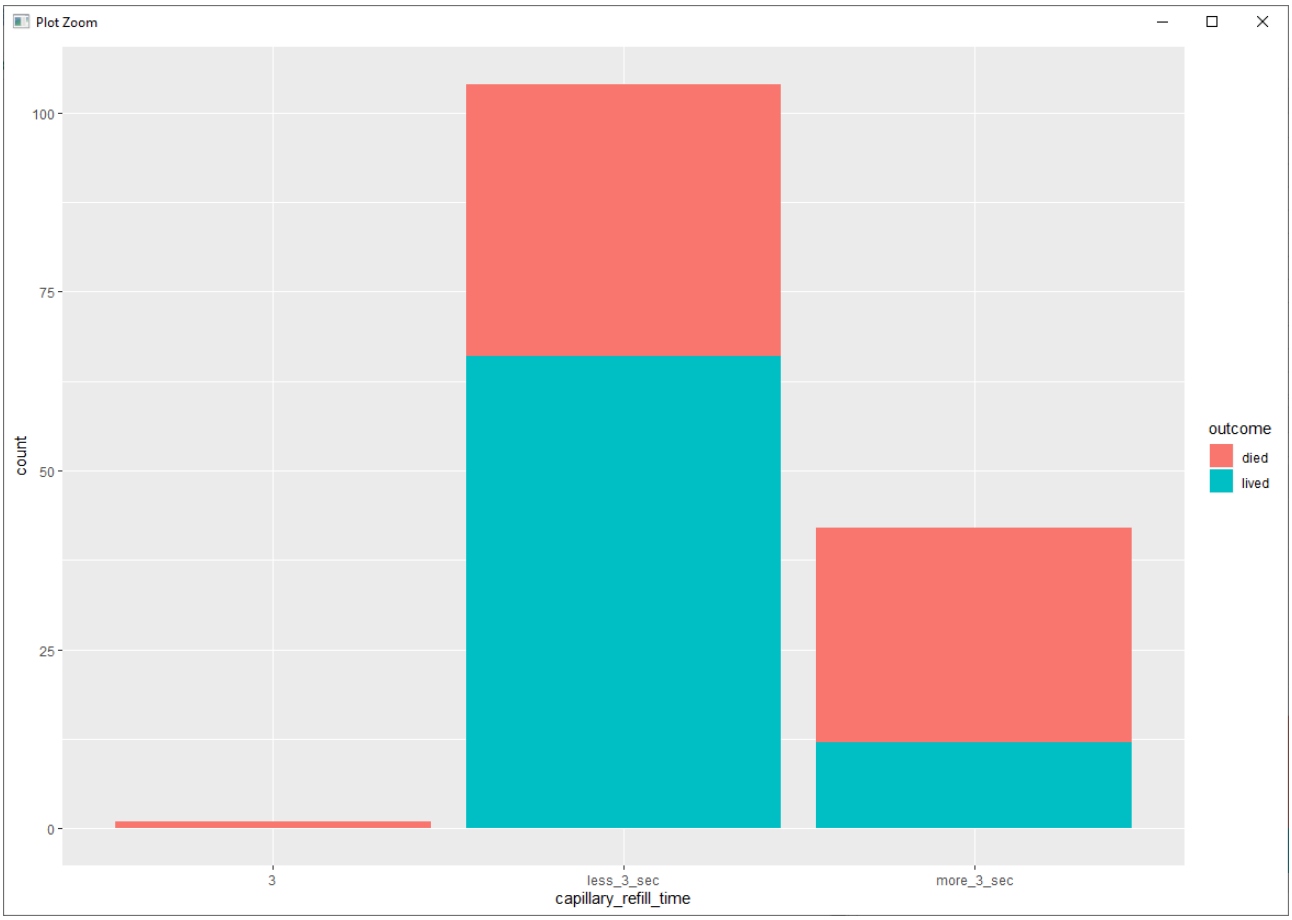

Figura 13: Gráfico de barras do atributo *capillary_refill_time* com *outcome*.

5.2 SVM

Ao aplicarmos o modelo *SVM* em nosso problema, obtivemos uma acurácia de quase 60%, o que não é bom. Este valor indica que o modelo talvez não seja apropriado para nosso problema e que talvez alguma etapa de pré-processamento da base do problema tenha sido feita errada.

```
> table(predictionsSVM, horses_test_normal$outcome)

predictionsSVM died lived
         died     0     0
         lived   36    53
> accuracy = 1 - mean(predictionsSVM != horses_test_normal$outcome)
> accuracy
[1] 0.5955056
> summary(svm_model)

Call:
svm(formula = outcome ~ ., data = horses_train_normal, probability = T)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1

Number of Support Vectors:  118

 ( 59 59 )


Number of Classes:  2

Levels:
 died lived
```

Figura 14: Modelo SVM.

5.3 Random Forest

Ao aplicarmos o modelo *Random Forest* em nosso problema, obtivemos uma acurácia de 87,6%. Este valor é bom, mas ainda não escolheria este modelo para teste em alguma base, uma vez que não é tão confiável que ele faça uma previsão adequada de quase todos os dados.

```
> #Random Forest
> system.time(forest_model <- randomForest(outcome ~., data = horses_train_normal,
+                                          importance = TRUE, do.trace = 100))
ntree      OOB      1      2
  100:  25.17% 27.54% 23.08%
  200:  26.53% 30.43% 23.08%
  300:  26.53% 30.43% 23.08%
  400:  25.17% 28.99% 21.79%
  500:  25.85% 31.88% 20.51%
  usuário   sistema decorrido
     0.11      0.00      0.11
> predictionsForest = predict(forest_model, horses_test_normal)
> table(predictionsForest, horses_test_normal$outcome)

predictionsForest died lived
            died    28     3
            lived    8    50
> accuracy = 1 - mean(predictionsForest != horses_test_normal$outcome)
> accuracy
[1] 0.8764045
> plot(forest_model)
> forest_model

Call:
 randomForest(formula = outcome ~ ., data = horses_train_normal,      importance = TRUE, do.
trace = 100)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 4

        OOB estimate of  error rate: 25.85%
Confusion matrix:
      died lived class.error
died    47    22   0.3188406
lived   16    62   0.2051282
```

Figura 15: Modelo *Random Forest*.

É possível visualizar também a importância de cada atributo na classificação final, como na figura a seguir. Os atributos mais acima da figura são os mais importantes. No nosso caso, os atributos *pain*, *packed_ cell_ volume* e *pulse* são os 3 atributos mais importantes pelos métodos de *Mean Decrease Accuracy* e *Mean Decrease Gini*.

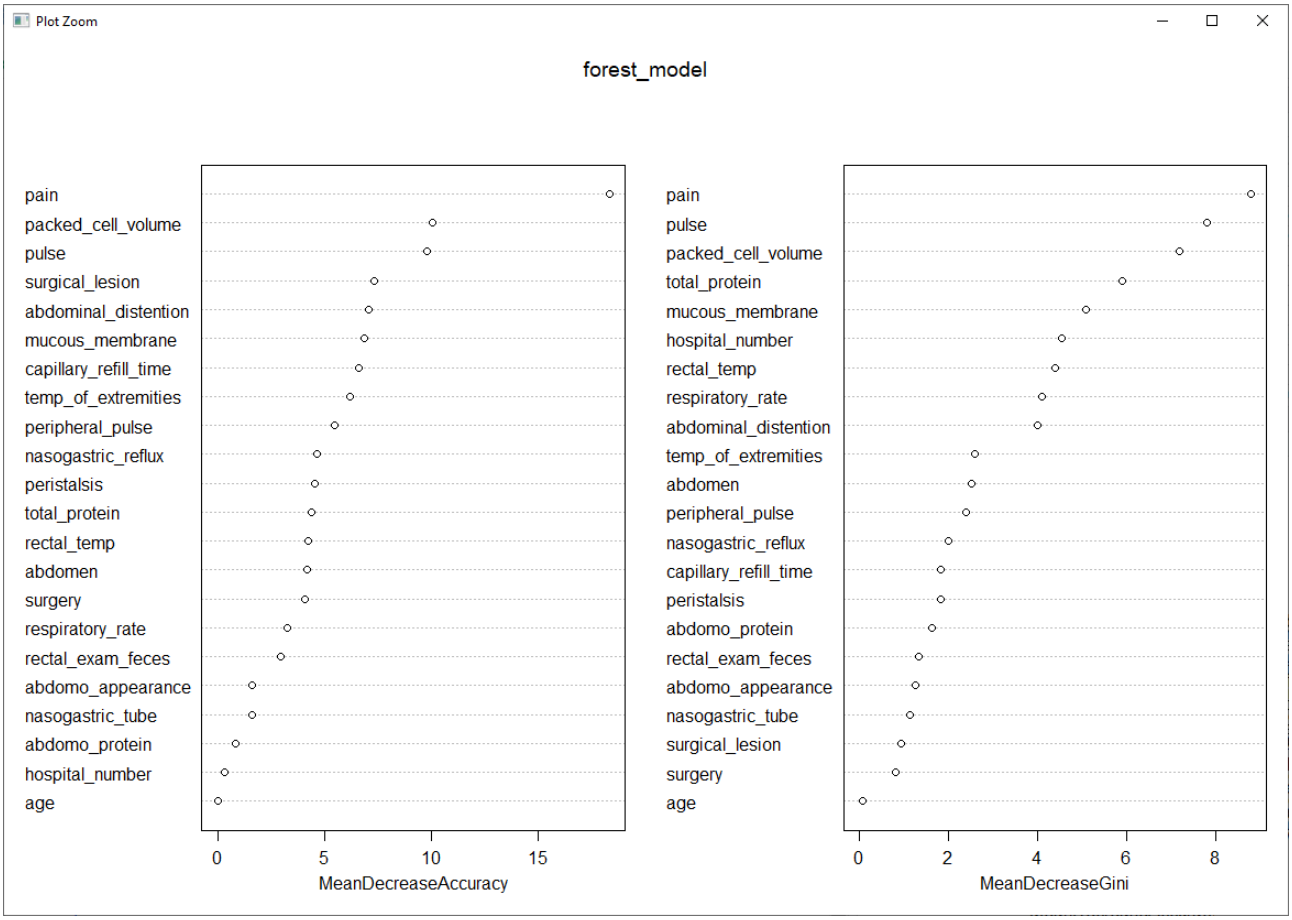

Figura 16: Importância dos atributos.

Podemos visualizar também a taxa de erros de acordo com os diferentes números de árvores até atingirmos a marca de 500 árvores, o *default* do modelo. A acurácia pode ser aumentada alterando-se o número de árvores do modelo. No meu caso, as 500 árvores foram tão boas quanto outros valores, então mative o *default*.

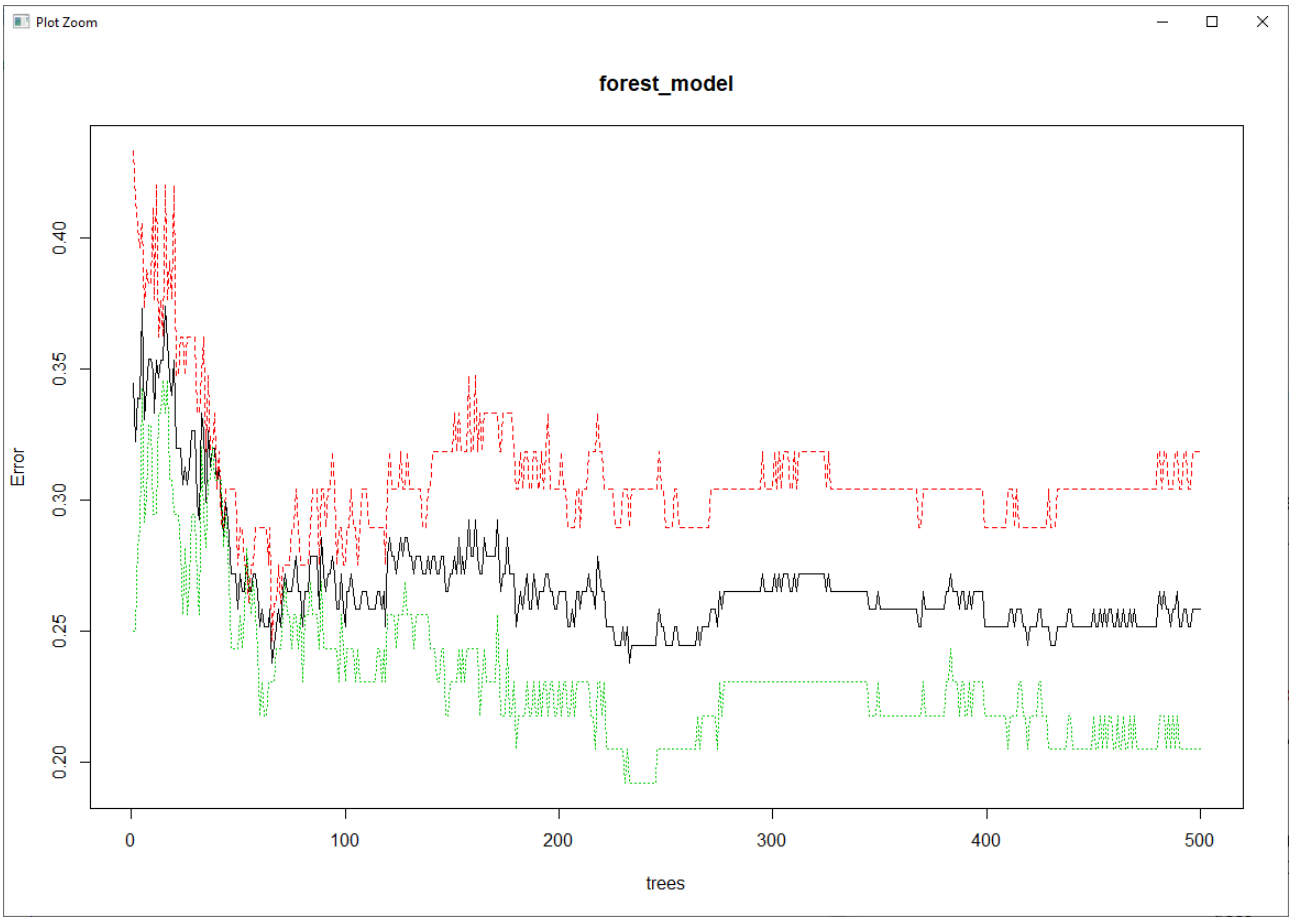

Figura 17: Gráfico

5.4 KNN

Ao aplicarmos o modelo *KNN* com os dados categóricos já transformados em numéricos, obtivemos uma acurácia de 65,2% aproximadamente.

```
#KNN
drop <- c("outcome")
horses_train_knn = horses_train_normal[,!(names(horses_train_normal) %in% drop)]
horses_test_knn = horses_test_normal[,!(names(horses_test_normal) %in% drop)]

#transformando os atributos categóricos em numéricos
classesTrain = lapply(horses_train_knn, class)
for (n in seq_along(classesTrain)){
  if (classesTrain[[n]] == "factor"){
    horses_train_knn[n] =  as.numeric(unlist(horses_train_knn[n]))
  }
}

classesTest = lapply(horses_test_knn, class)
for (n in seq_along(classesTest)){
  if (classesTest[[n]] == "factor"){
    horses_test_knn[n] =  as.numeric(unlist(horses_test_knn[n]))
  }
}


system.time(knn_model <- knn(horses_train_knn,
                             horses_test_knn,
                  cl = horses_train_normal$outcome, k = 5))

table(knn_model, horses_test_normal$outcome)
accuracy = 1 - mean(knn_model != horses_test_normal$outcome)
accuracy
```

Figura 18: Código do modelo *KNN*.

As duas acurácias muito baixas dos modelos *KNN* e *SVM* indicam que realmente algo deve ser modificado no pré-processamento da base de dados.

```
> table(knn_model, horses_test_normal$outcome)

knn_model died lived
    died    20    15
    lived   16    38
> accuracy = 1 - mean(knn_model != horses_test_normal$outcome)
> accuracy
[1] 0.6516854
>
```

Figura 19: Modelo *KNN*.

Portanto, resolvi não aplicar a limpeza de *outliers*, uma vez que parece que muitos dados foram removidos e os modelos parecem ter dificuldade na classificação do problema.

5.5 Novo teste com modelos otimizados

Ao rodar os dados de treinamento e teste já normalizados e com preenchimento de *missing values*, mas sem a remoção de *outliers*, obtivemos uma melhora significativa em todos os modelos, o que comprova o que já suspeitávamos: a remoção de *outliers* foi responsável por criar uma tendência errada e atrapalhar na classificação dos modelos devido a grande quantidade de remoção de valores.

5.5.1 Decision Tree otimizado

Sem a remoção de *outliers*, a acurácia do *Decision Tree* passou a ser de 87,6%. O novo valor é muito melhor que o anterior de 77,5%.

```
> #Decision Tree
> system.time(tree_model <- tree(outcome ~., horses_train_normal))
  usuário   sistema decorrido
        0         0         0
> predictionsDTree <- predict(tree_model, horses_test_normal, type="class")
> table(predictionsDTree, horses_test_normal$outcome)

predictionsDTree died lived
           died    31     6
           lived    5    47
> accuracy = 1 - mean(predictionsDTree != horses_test_normal$outcome)
> accuracy
[1] 0.8764045
```

Figura 20: *Decision Tree* sem remoção de *outliers*.

5.5.2 SVM otimizado

Sem a remoção de *outliers*, a acurácia do *SVM* passou a ser de 82%. O novo valor é muito melhor que o anterior de 60%.

```
> #SVM
> system.time(svm_model <- svm(outcome ~., horses_train_normal, probability = T))
  usuário   sistema decorrido
     0.36      0.00      0.36
> predictionsSVM <- predict(svm_model, horses_test_normal, probability = T)
> table(predictionsSVM, horses_test_normal$outcome)

predictionsSVM died lived
         died    27     7
         lived    9    46
> accuracy = 1 - mean(predictionsSVM != horses_test_normal$outcome)
> accuracy
[1] 0.8202247
```

Figura 21: *SVM* sem remoção de *outliers*.

5.5.3 Random Forest otimizado

Sem a remoção de *outliers*, a acurácia do *Random Forest* passou a ser de 100%. O novo valor é muito melhor que o anterior de 87,6%.

```
> #Random Forest
> system.time(forest_model <- randomForest(outcome ~., data = horses_train_normal,
+                                          mtry = 8, importance = TRUE,
+                                          do.trace = 100))
ntree      OOB      1      2
  100:  27.76% 37.19% 21.35%
  200:  25.42% 33.88% 19.66%
  300:  24.41% 33.88% 17.98%
  400:  24.75% 34.71% 17.98%
  500:  24.08% 34.71% 16.85%
  usuário   sistema decorrido
     0.26      0.00      0.27
> predictionsForest = predict(forest_model, horses_test_normal)
> table(predictionsForest, horses_test_normal$outcome)

predictionsForest died lived
            died    36     0
            lived    0    53
> accuracy = 1 - mean(predictionsForest != horses_test_normal$outcome)
> accuracy
[1] 1
```

Figura 22: *Random Forest* sem remoção de *outliers*.

5.5.4 KNN otimizado

Sem a remoção de *outliers*, a acurácia do *KNN* passou a ser de 83,14%. O novo valor é muito melhor que o anterior de 65,2%.

```
> table(knn_model, horses_test_normal$outcome)

knn_model died lived
    died    22     1
    lived   14    52
> accuracy = 1 - mean(knn_model != horses_test_normal$outcome)
> accuracy
[1] 0.8314607
```

Figura 23: *KNN* sem remoção de *outliers*.

6 Conclusão

O modelo de maior acurácia foi o *Random Forest*, com acurácia de 100%. Portanto, para nosso problema, o melhor modelo seria o *Random Forest*, que por acaso deu o melhor resultado possível e pode sem dúvidas ser utilizado para futuras classificações de novos dados no mesmo formato.