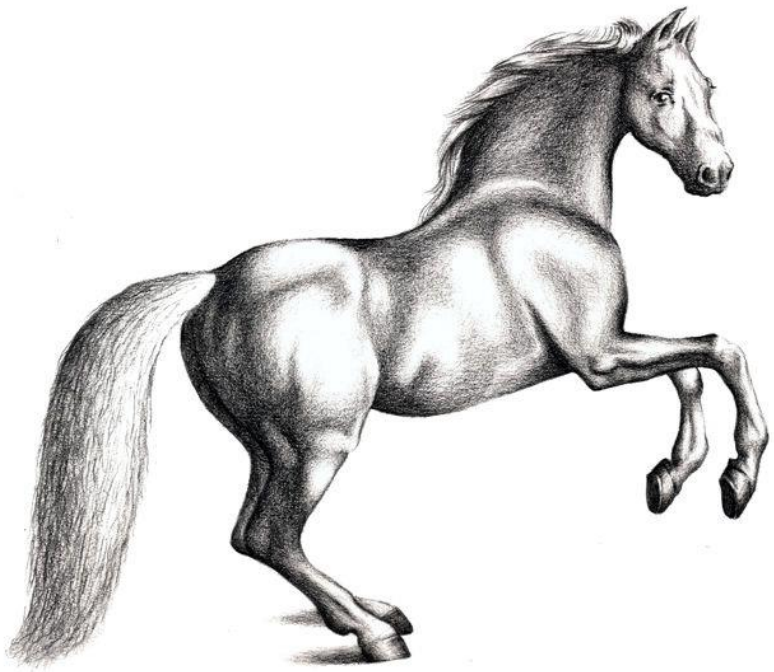
Trabalho Final
Data Mining



Integrantes:

Letícia Aranha
lsf.aranha@gmail.com

Matheus Rangel Cardoso
matheus.rangel.cardoso@gmail.com

Sumário

1. Identificação do Problema:

O projeto tem como objetivo produzir um modelo capaz de prever o diagnóstico de cavalos em três classes distintas de saída: vivo, morto e eutanásia. Os atributos contidos na base de dados utilizada para a criação do modelo são:

Atributo Original na Base	Tradução
surgery	Cirurgia
age	Idade
hospital_number	ID do hospital
rectal_temp	Temperatura retal
pulse	pulso
respiratory_rate	taxa respiratória
temp_of_extremities	Temperatura das Extremidades
peripheral_pulse	Pulso Periférico
mucous_membrane	Membrana Mucosa
capillary_refill_time	Tempo de Reposição Capilar
pain	Dor
peristalsis	Movimento Peristáltico
abdominal_distention	Distensão abdominal
nasogastric_tube	Tubo Nasogástrico
nasogastric_reflux	Refluxo Nasogástrico
nasogastric_reflux_ph	Ph do Refluxo Nasogástrico
rectal_exam_feces	Exame de Fezes
abdomen	Abdomen
packed_cell_volume	Volume Celular
total_protein	Proteína Total
abdomo_appearance	Aparência Abdominal
abdomo_protein	Proteína Abdominal
outcome	Diagnóstico
surgical_lesion	Lesão Cirúrgica
lesion_1	Lesão 1
lesion_2	Lesão 2
lesion_3	Lesão 3
cp_data	Patologia

Figura 1 - Tabela de atributos

2. Ferramenta Utilizada:

Visto que o R Studio é uma ferramenta Open Sourse, todos os processos abaixo foram desenvolvidos usando a ferramenta:

- Tratamento das bases de Treino e Teste
- Criação de gráficos e processos de análise de resultados
- Treinamento dos modelos

Ao final do código, basta executar o modelo treinado para prever o diagnóstico de um cavalo (vivo, morto ou submetido a eutanásia).

3. Análise Exploratória e Pré-Processamento

Premissas:

- Para realizar o treinamento de um modelo capaz de prever o diagnóstico de cavalos, foi preciso tratar a base de dados original.

- Os dados encontram-se separados em duas bases distintas (base de *"treino"* e base de "*teste"*), para que seja possível analisar o desempenho do modelo treinado em aferir corretamente sobre dados ainda não explorados. Importante para se evitar o "overfitting[1]" do modelo.

- Quando iniciado a etapa de tratamento de dados, alguns dos ajustes realizados na base de treino deverão ser replicados na base de teste.

3.1. Atributos Excluídos:

3.1.1. Atributos que representam ID's:

Alguns atributos presentes na base representam ID's. Ou seja, tratam-se de códigos que possibilitam a identificação de um registro ("uma linha") em uma tabela além de serem utilizados para fazer a correlação entre informações entre duas tabelas. Para o exercício proposto, tais entradas não representam nenhum dado relevante que auxilie o modelo a prever o diagnóstico do animal, tendo inclusive um resultado inverso, prejudicando a capacidade preditiva do modelo, já que o modelo processa o ID como um conjunto numérico.

[1] Quando o modelo treinado se ajusta perfeitamente bem à base de teste, obtendo um alto nível de precisão, mas se mostra ineficaz quando exposto à uma nova base, apresentando um alto índice de erros.

São estes:

- hospital_number
- lesion_1
- lesion_2
- lesion_3

Premissas:

Hospital Number:

O atributo “hospital_number” representa o ID no hospital ao qual o animal foi registrado em período de tratamento.

Lesion_n:

Os registros referentes aos atributos de lesão (lesion_1, lesion_2 e lesion_3) representam diferentes tipos de lesões de acordo com a concatenação de seus valores. Cada entrada é composta por 4 números (4 ID’s), cujos valores representam, respectivamente, o local da lesão, seu tipo, subtipo e um código específico, como é possível observar na documentação abaixo:

```
25, 26, 27: type of lesion
- first number is site of lesion
1 = gastric
2 = sm intestine
3 = lg colon
4 = lg colon and cecum
5 = cecum
6 = transverse colon
7 = retum/descending colon
8 = uterus
9 = bladder
11 = all intestinal sites

00 = none
- second number is type
1 = simple
2 = strangulation
3 = inflammation
4 = other
- third number is subtype
1 = mechanical
2 = paralytic
0 = n/a
- fourth number is specific code
1 = obturation
2 = intrinsic
3 = extrinsic
4 = adynamic
5 = volvulus/torsion
6 = intussuption
7 = thromboembolic
8 = hernia
9 = lipoma/slenic incarceration
10 = displacement
0 = n/a
```

Figura 2 - Documentação atributos de lesão

3.1.2. Cp_data

O atributo “cp_data” foi excluído do dataframe, pois de acordo com a documentação, este atributo é “irrelevante” já que os dados referentes à “patologia” não estão presentes na base:

```
28: cp_data
- is pathology data present for this case?
1 = Yes
2 = No
- this variable is of no significance since pathology data is not included or collected for these
cases
```

Figura 3 - Documentação sobre atributos “cp_data”

3.2. Valores Nulos:

A metodologia para tratar os valores nulos consiste na verificação da porcentagem de valores nulos em cada atributo.

1) Se a porcentagem de valores nulos for abaixo de 50%, imputar um valor:
    a. Variável Contínua ou Discreta → imputar a mediana.
    b. Variável for Categórica → imputar a moda.

2) Se a porcentagem de valores nulos for maior do que 50%:
    a. Excluir o atributo do modelo.

3.2.1. Valores Nulos:

Foi elaborado um código que retorna a contagem de valores nulos e a porcentagem dos valores nulos presentes nos atributos na base de Treino, conforme ilustra a figura abaixo:

	contagem_nulos_treino	nulos_treino_porcentagem
surgery	0	0.000000
age	0	0.000000
hospital_number	0	0.000000
rectal_temp	60	20.066890
pulse	24	8.026756
respiratory_rate	58	19.397993
temp_of_extremities	56	18.729097
peripheral_pulse	69	23.076923
mucous_membrane	47	15.719064
capillary_refill_time	32	10.702341
pain	55	18.394649
peristalsis	44	14.715719
abdominal_distention	56	18.729097
nasogastric_tube	104	34.782609
nasogastric_reflux	106	35.451505
nasogastric_reflux_ph	246	82.274247
rectal_exam_feces	102	34.113712
abdomen	118	39.464883
packed_cell_volume	29	9.698997
total_protein	33	11.036789
abdomo_appearance	165	55.183946
abdomo_protein	198	66.220736
outcome	0	0.000000
surgical_lesion	0	0.000000
lesion_1	0	0.000000
lesion_2	0	0.000000
lesion_3	0	0.000000
cp_data	0	0.000000

Figura 4 - Contagem de nulos por atributo

3.2.1.1. Valores Nulos excluídos do modelo:

Os atributos excluídos do modelo devido à sua alta incidência de valores nulos são:

- nasogastric_reflux_ph - 246 registos nulos - 82.27% Nulos
- abdomo_appearance - 165 registros nulos - 55.18% Nulos
- abdomo_protein - 198 registros nulos - 66.22% Nulos

A substituição de uma alta ocorrência de “NAs” pela mediana ou pela moda pode acarretar em um viés nos dados, prejudicando a capacidade preditiva do modelo, pois a reincidência dos mesmos valores (principalmente no que se refere à atributos de variáveis contínuas ou discretas) tornaria a variância[2] do atributo próxima de 0.

[2] Em um conjunto de dados, a variância é uma medida de dispersão estatística responsável por mostrar o quão distante cada registro está do valor central (média). De um modo geral, quanto menor é a variância (quanto mais próxima de 0), mais irrelevante o atributo se torna na hora de treinar o modelo preditivo. Fonte: Link.

3.2.1.2. Decisão: Imputar Registros

Para os demais atributos, foi realizada uma análise gráfica para julgar a viabilidade de substituir os valores nulos ("null") de acordo com as seguintes instruções:

3.2.1.2.1. Variáveis Contínuas e Discretas:

Para os demais atributos com variáveis contínuas ou discretas, optou-se por substituir os valores nulos pela mediana, e não pela média, para que a existência de outliers ainda não tratados afetasse o novo valor imputado.

Abaixo seguem os atributos que foram tratados seguindo esse preceito:

- rectal_temp - 60 Nulos (20.06%)
- Pulse - 24 Nulos (8.02%)
- respiratory_rate - 58 Nulos (19.39%)
- packed_cell_volume - 29 Nulos (9.69%)
- total_protein - 33 Nulos (11.03%)

3.2.1.2.2. Variáveis Categóricas

Para as variáveis categóricas, foi necessário analisar a incidência de cada categoria e identificar aquelas que tiveram maior ocorrência. Para tal, optou-se por gerar gráficos do tipo barra para cada um dos atributos, conforme as imagens abaixo:

- **temp_of_extremities - 56 Nulos (18.72%)**

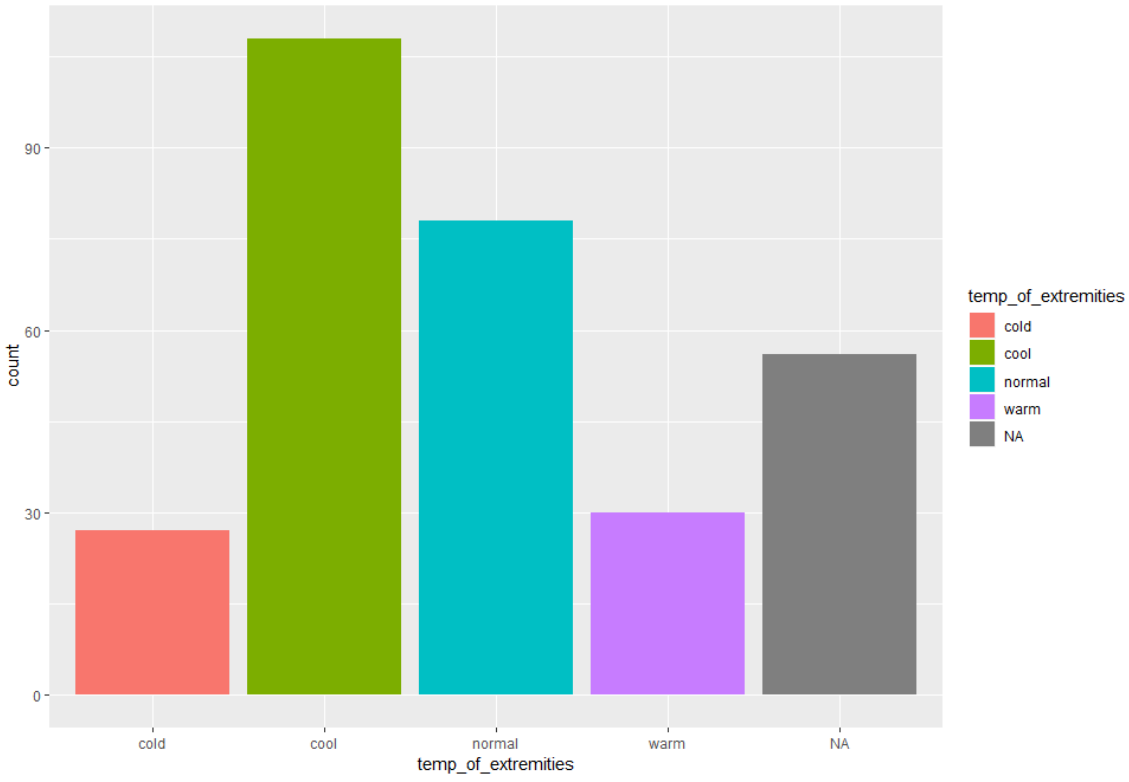


Figura 5 - Gráfico temp_of_extremities

- **peripheral_pulse - 69 Nulos (23.07%)**

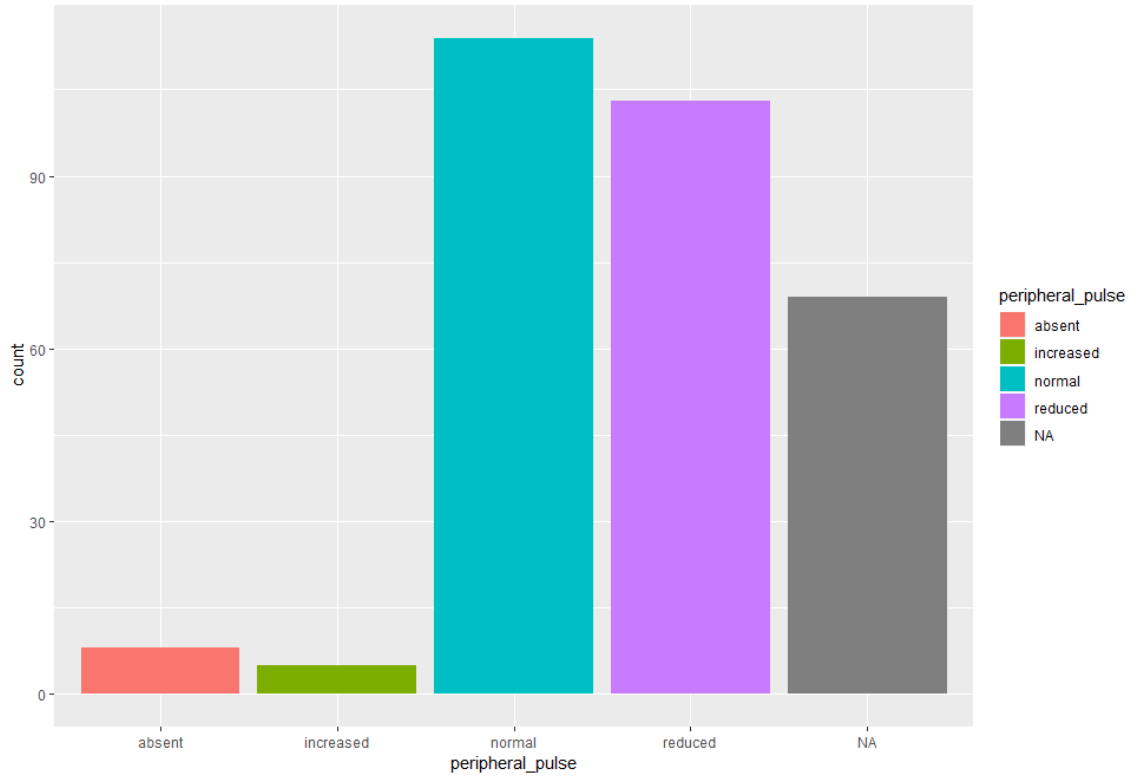


Figura 6 - Gráfico peripheral_pulse

- **mucous_membrane - 47 Nulos (15.71%)**

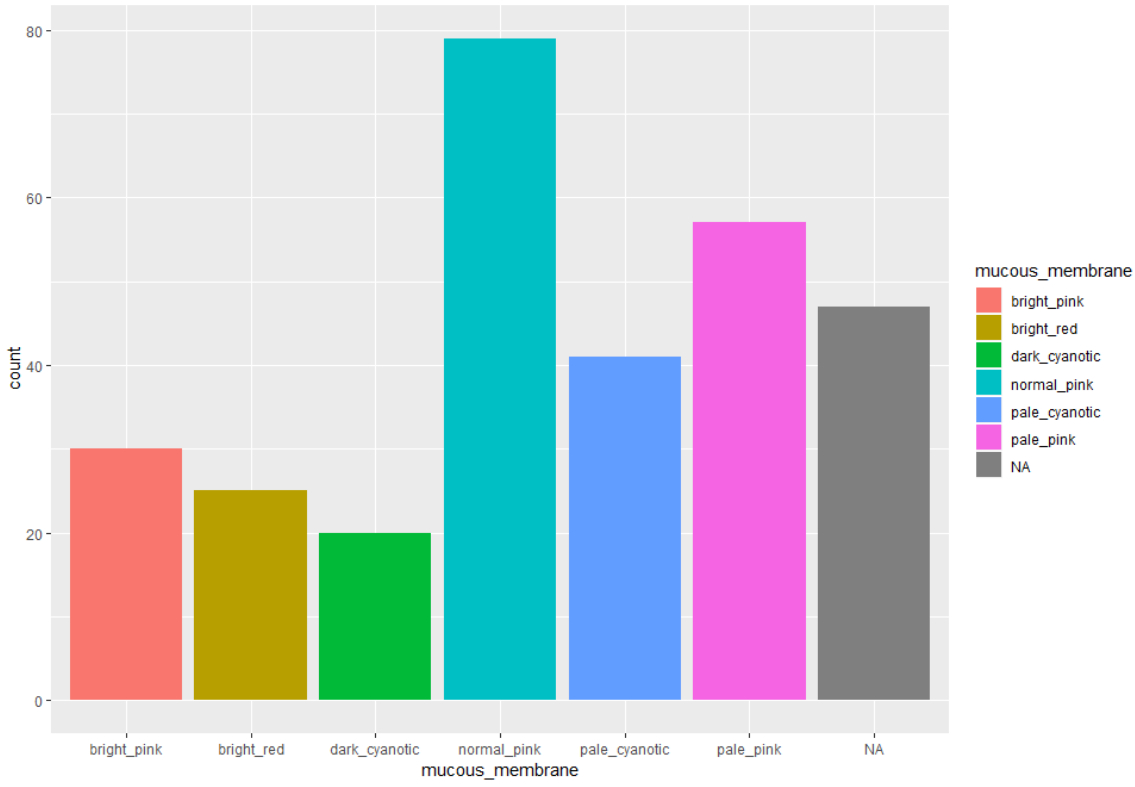


Figura 7 - Gráfico mucous_membrane

- **capillary_refill_time - 32 Nulos (10.70%)**

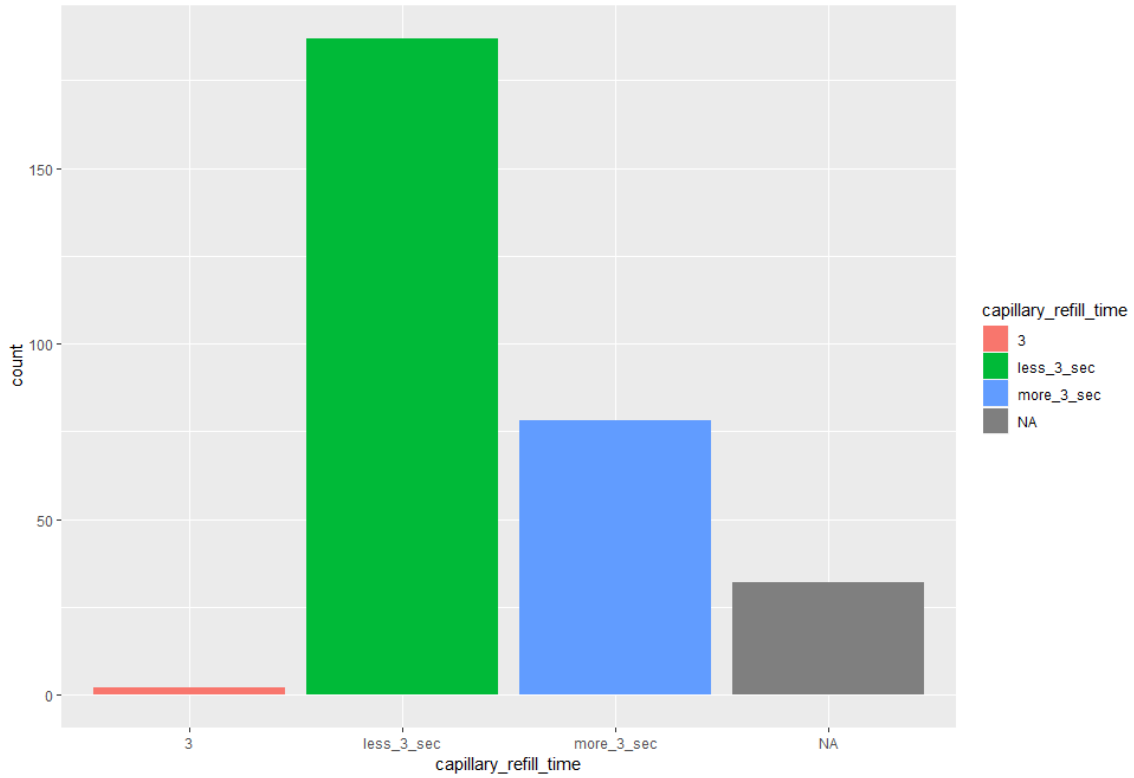


Figura 8 - Gráfico capillary_refill_time

- **pain - 55 Nulos (18.39%)**

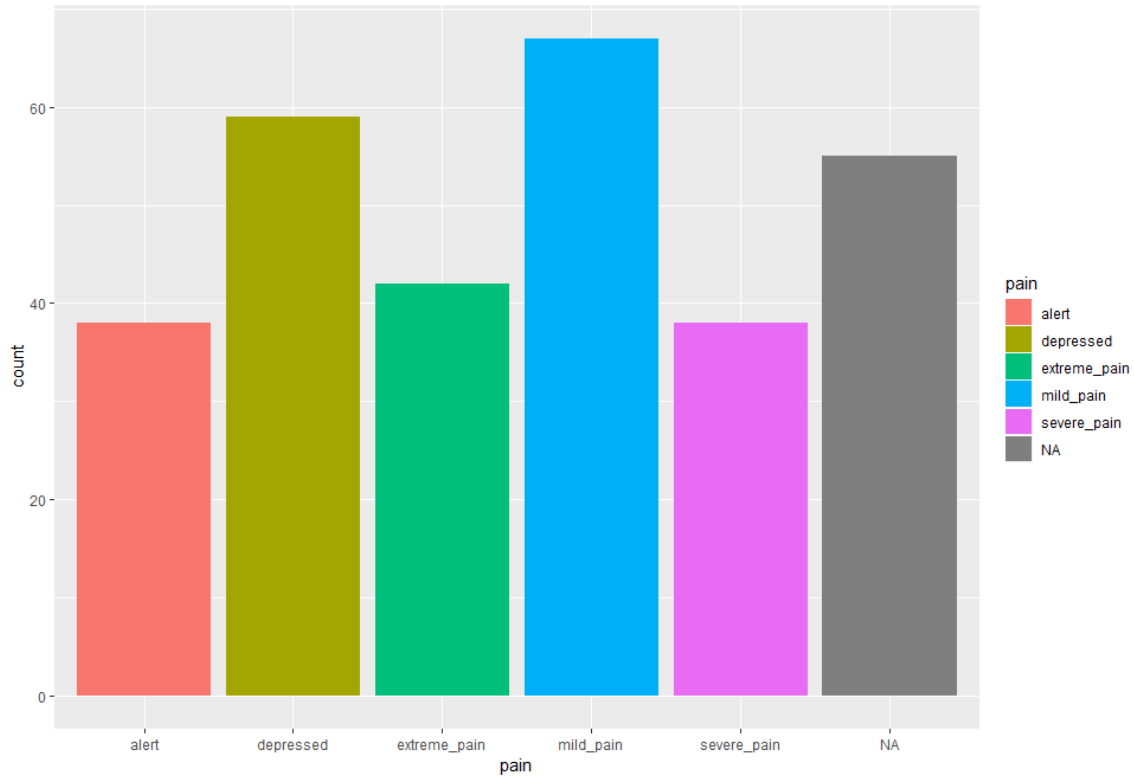


Figura 9 - Gráfico pain

- **peristalsis - 44 Nulos (14.71%)**

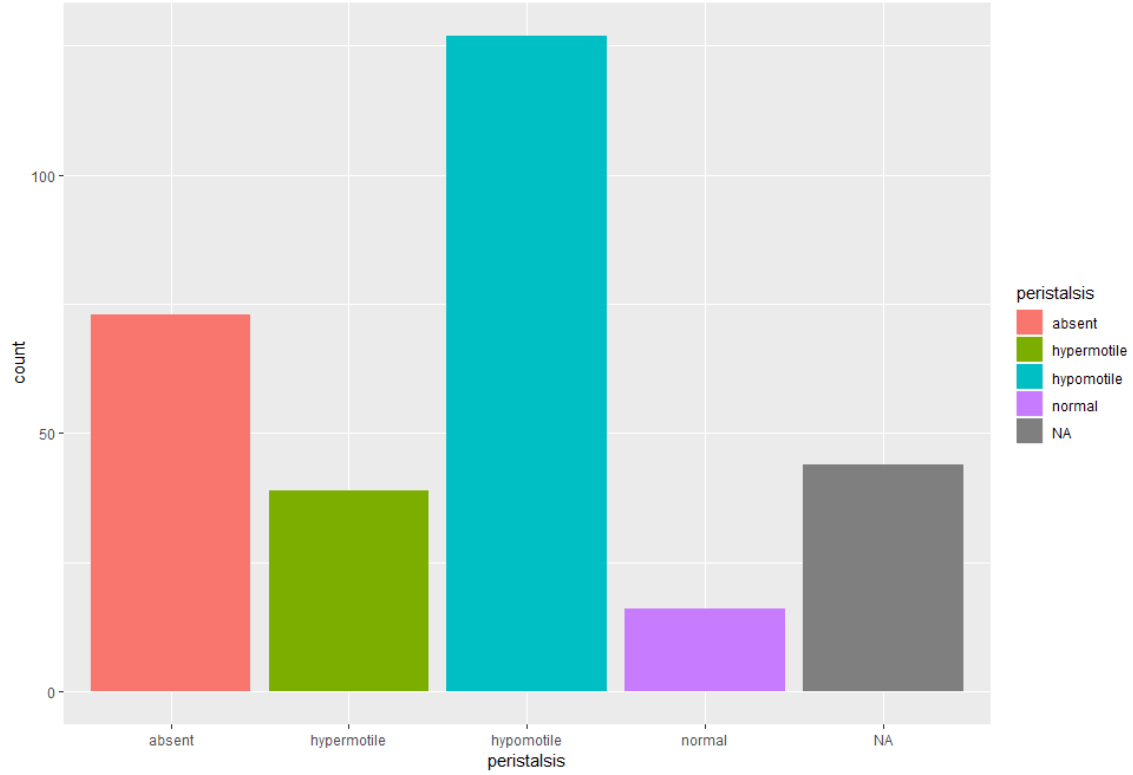


Figura 10 - Gráfico peristalsis

- **abdominal_distention - 56 Nulos (18.72%)**

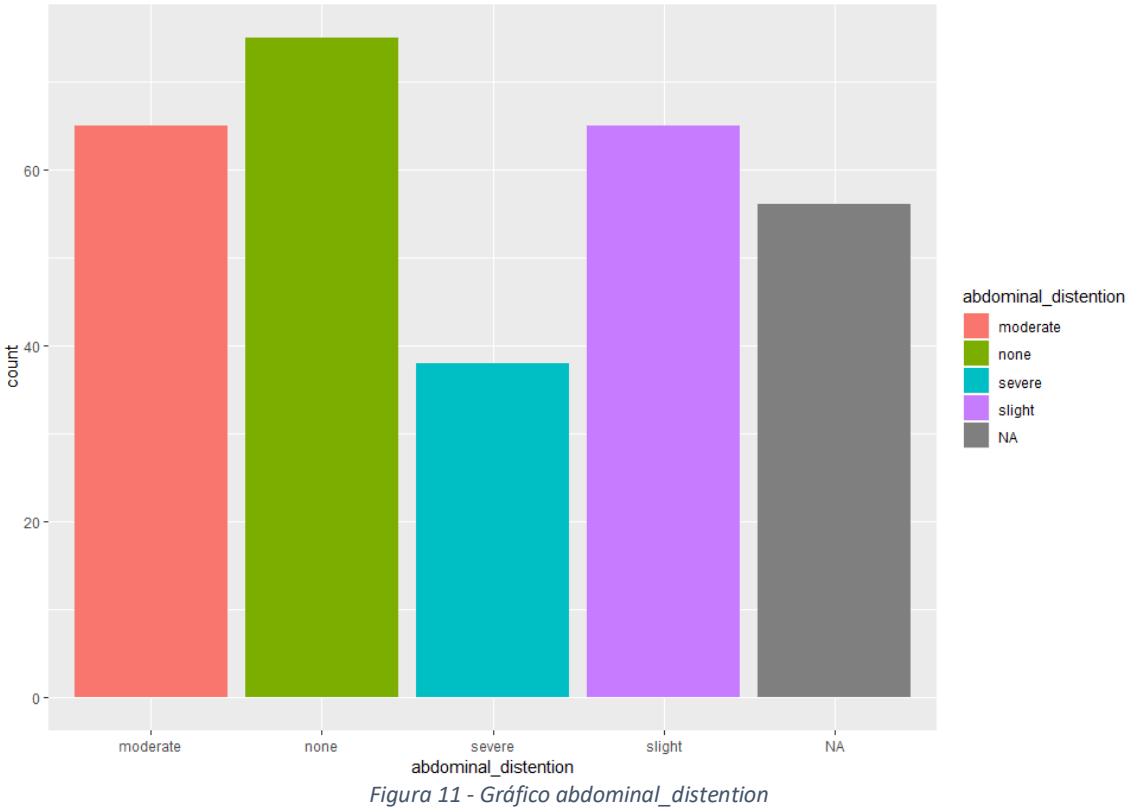


Figura 11 - Gráfico abdominal_distention

- **nasogastric_tube - 104 Nulos (34.78%)**

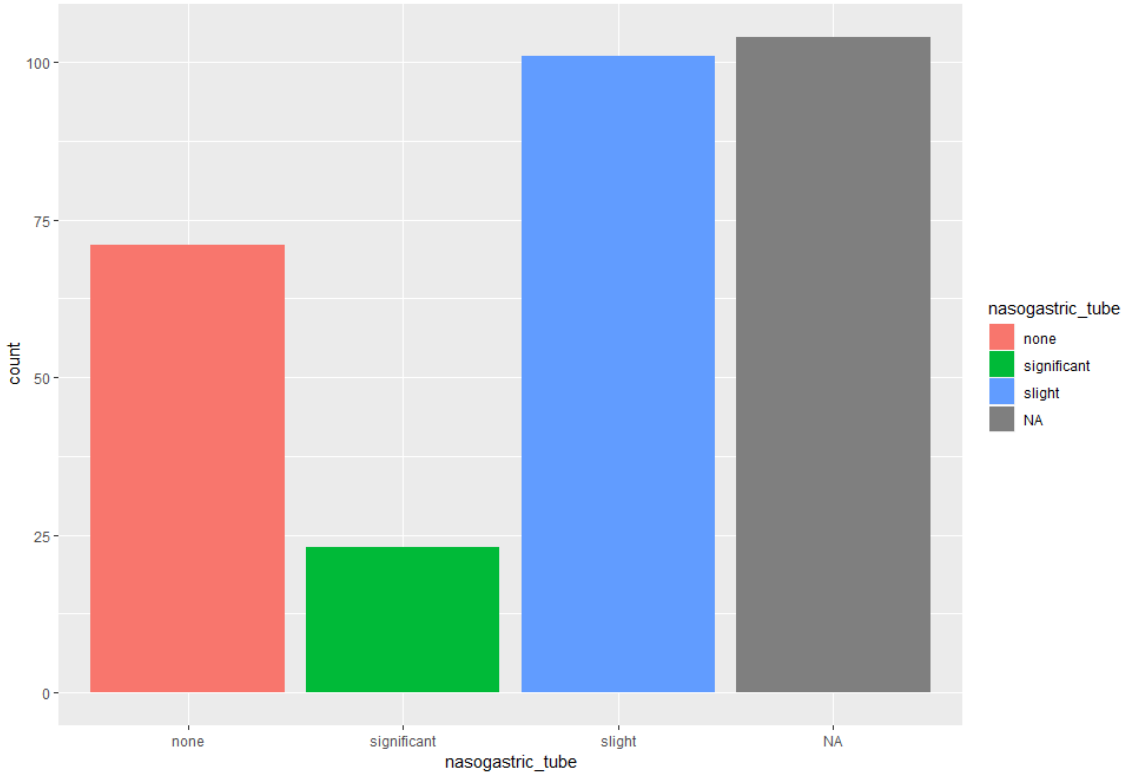


Figura 12 - Gráfico nasogastric_tube

- **nasogastric_reflux - 106 Nulos (35.45%)**

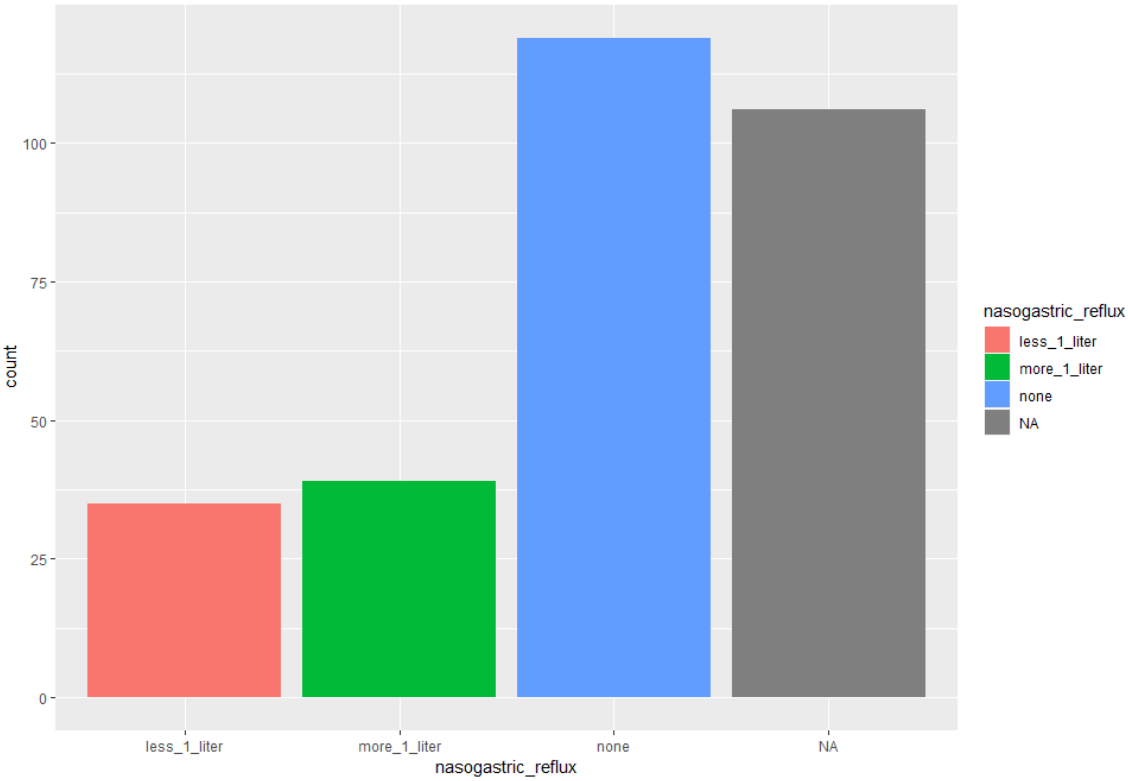


Figura 13 - Gráfico nasogastric_reflux

- **abdomen - 118 Nulos (39,46%)**

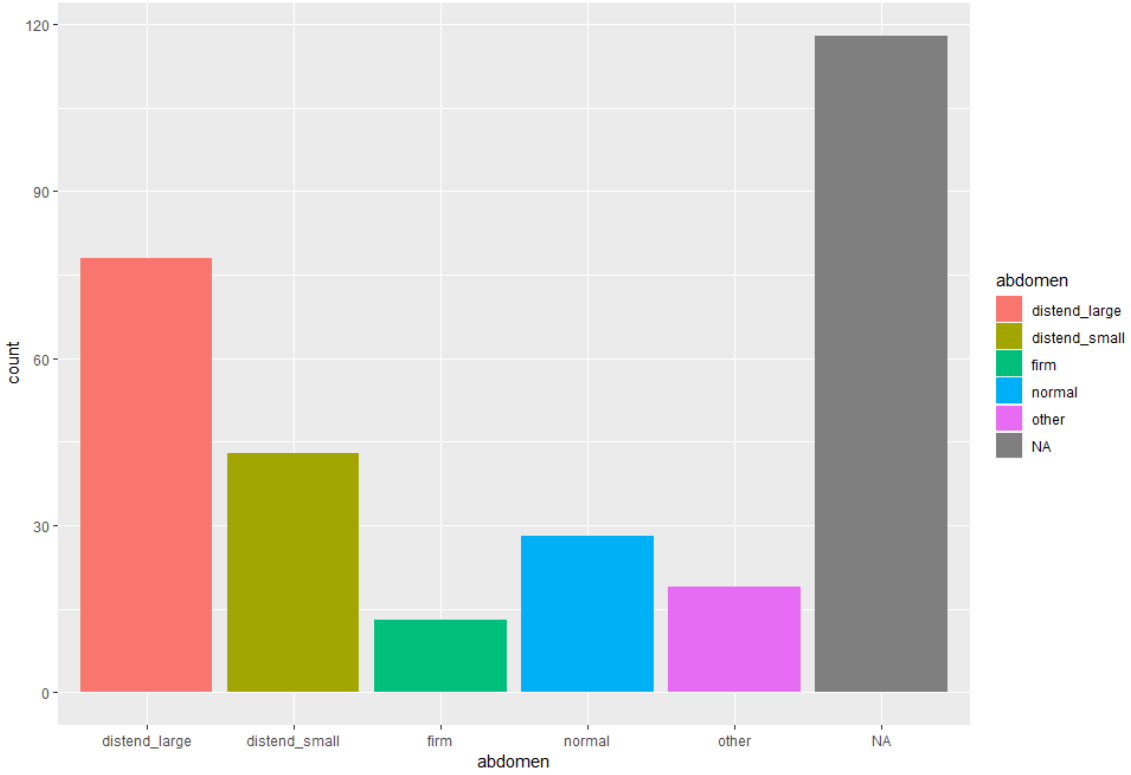


Figura 14 - Gráfico abdomen

- **rectal_exam_feces - 102 Nulos (34.11%)**

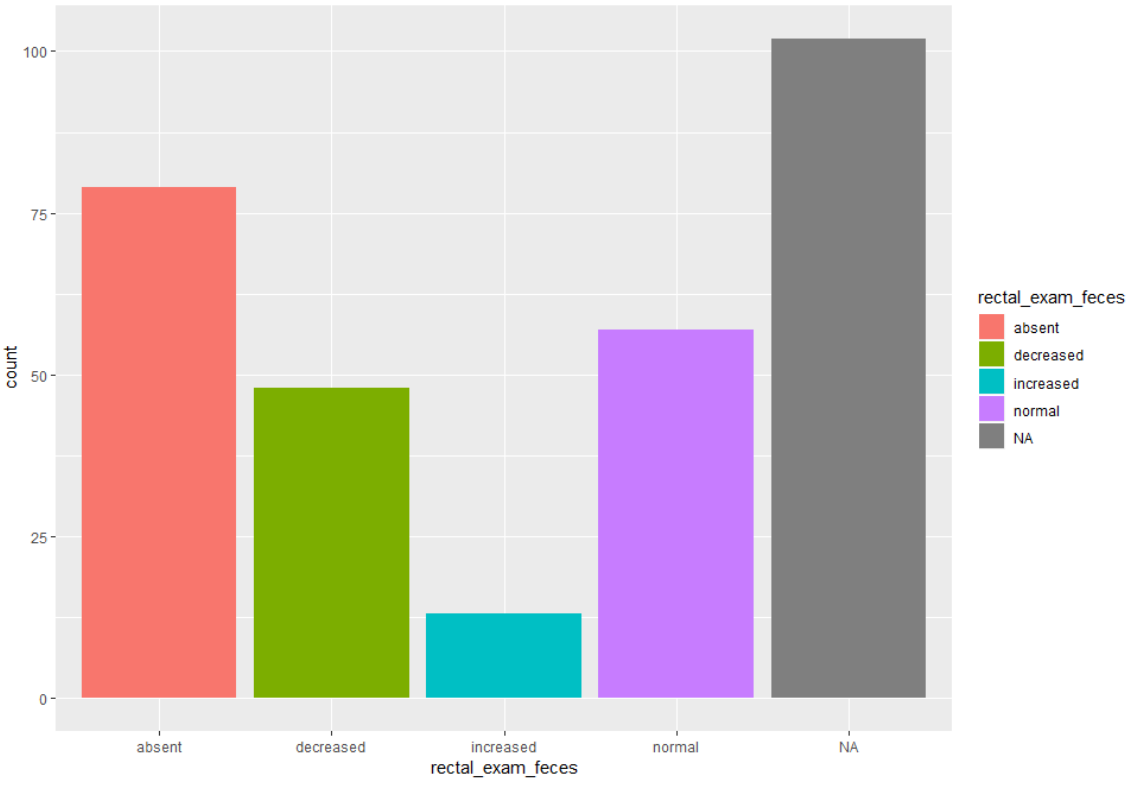


Figura 15 - Gráfico rectal_exam_feces

Para todos os atributos listados acima, os valores nulos foram substituídos pela moda, ou seja, pela classe de maior ocorrência. Foram:

- Classe “cool” em temp_of_extremities;
- Classe “normal” em peripheral_pulse;
- Classe “normal_pink” em mucous_membrane;
- Classe “< 3 seconds” em capillary_refill_time;
- Classe “mild_pain” em pain;
- Classe “hypomotile” em peristalsis;
- Classe “none” em abdominal_distension;
- Classe “slight” em nasogastric_tube;
- Classe “none” em nasogastric_reflux;
- Classe “absent” em rectal_exam_feces;
- Classe “distend_large” em abdomen.

3.3. Outliers

Outliers são valores extremos presentes em uma base de dados, que podem ser considerados como atípicos, provenientes muitas vezes de erros de digitação ou equívocos na coleta dos dados. Valores muito discrepantes podem causar um viés nos dados, o que prejudica a capacidade preditiva do modelo.

Para a análise dos Outliers nas variáveis discretas do modelo, optou-se pela visualização dos dados em um gráfico do tipo *box-plot*. Esse modelo de representação gráfica possibilita visualizar diversos aspectos da distribuição dos dados, tais como posição, variabilidade e assimetria, além da ocorrência de valores atípicos, que são representados como pontos fora do desvio padrão.

Abaixo seguem os gráficos obtidos para as variáveis analisadas:

- **rectal_temp**

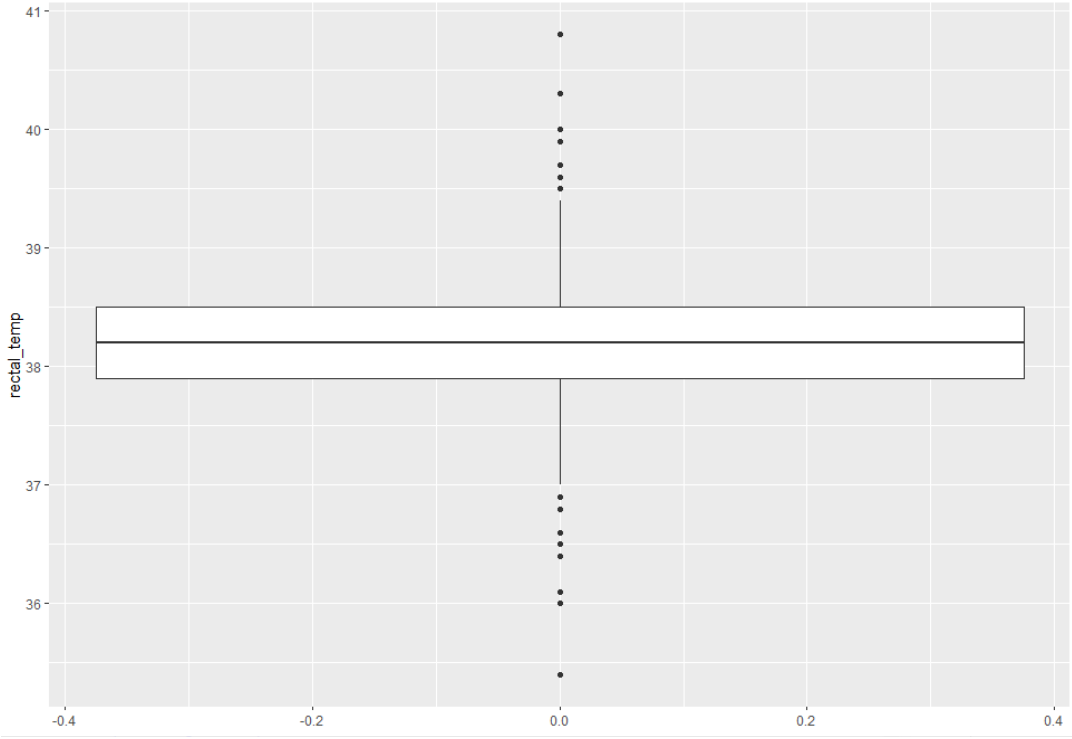


Figura 16 - Box-plot rectal_temp

- **pulse**

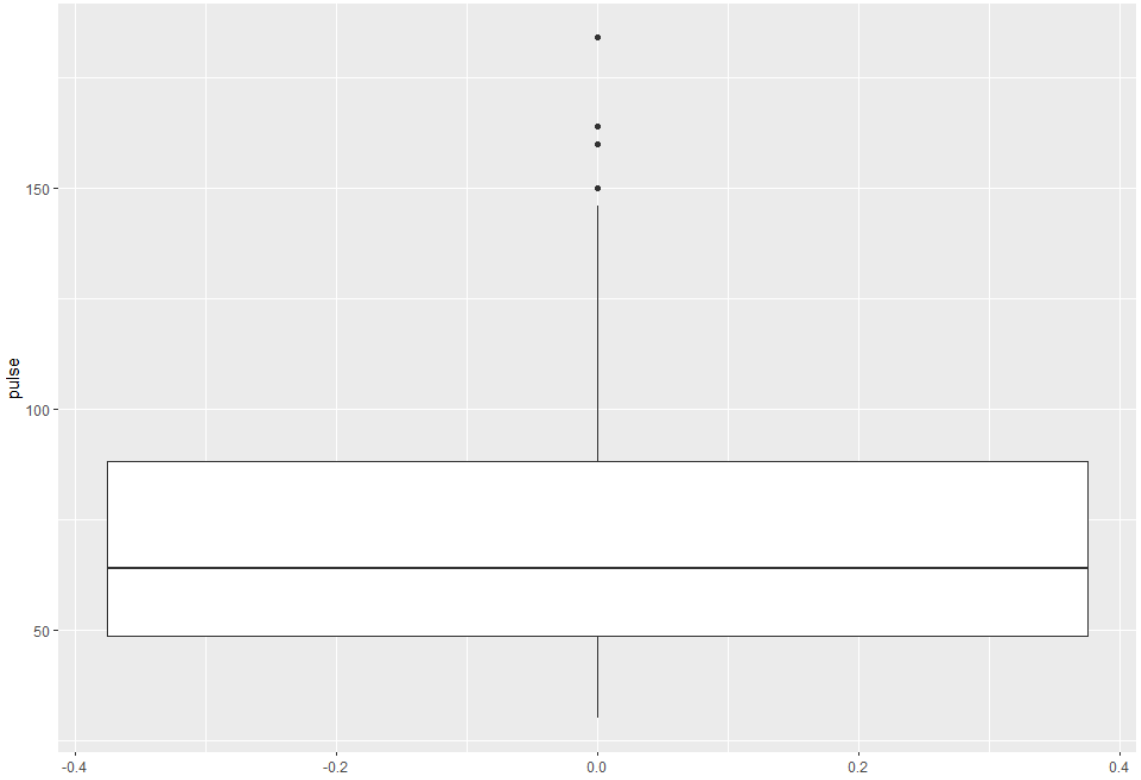


Figura 17 - Box-plot pulse

- **respiratory_rate**

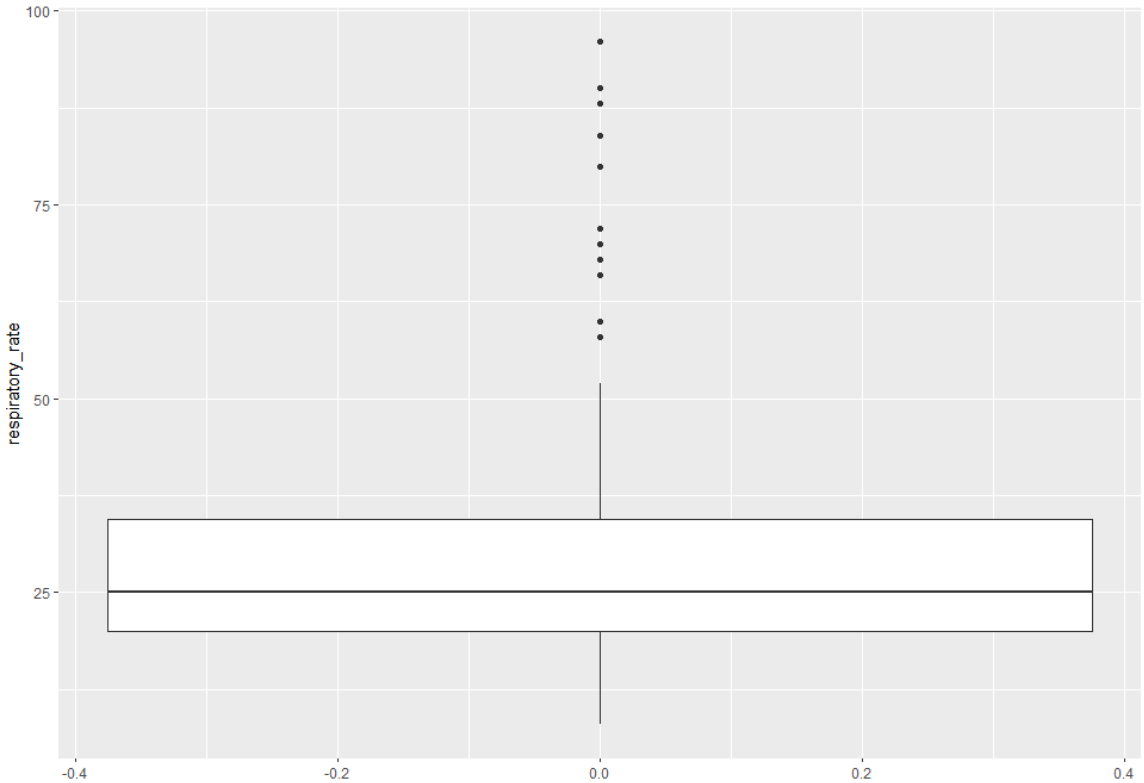


Figura 18 - Box-plot respiratory_rate

- **packed_cell_volume**

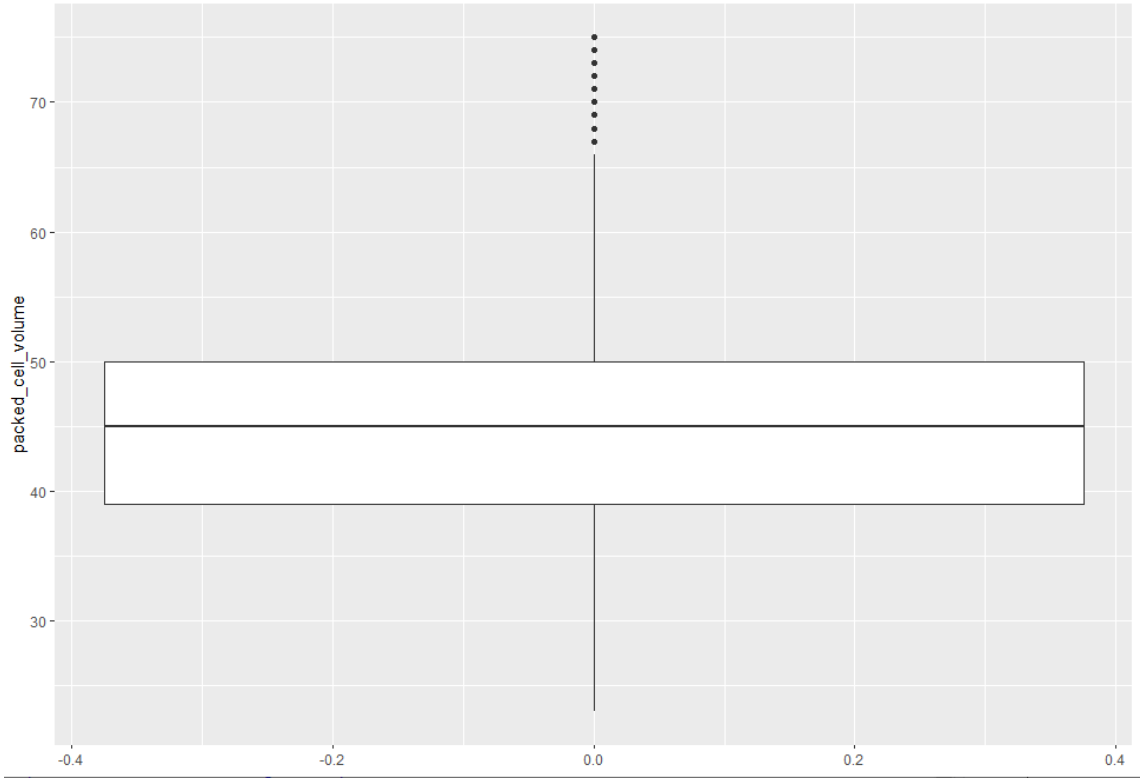


Figura 19 - Box-plot packed_cell_volume

- **total_protein**

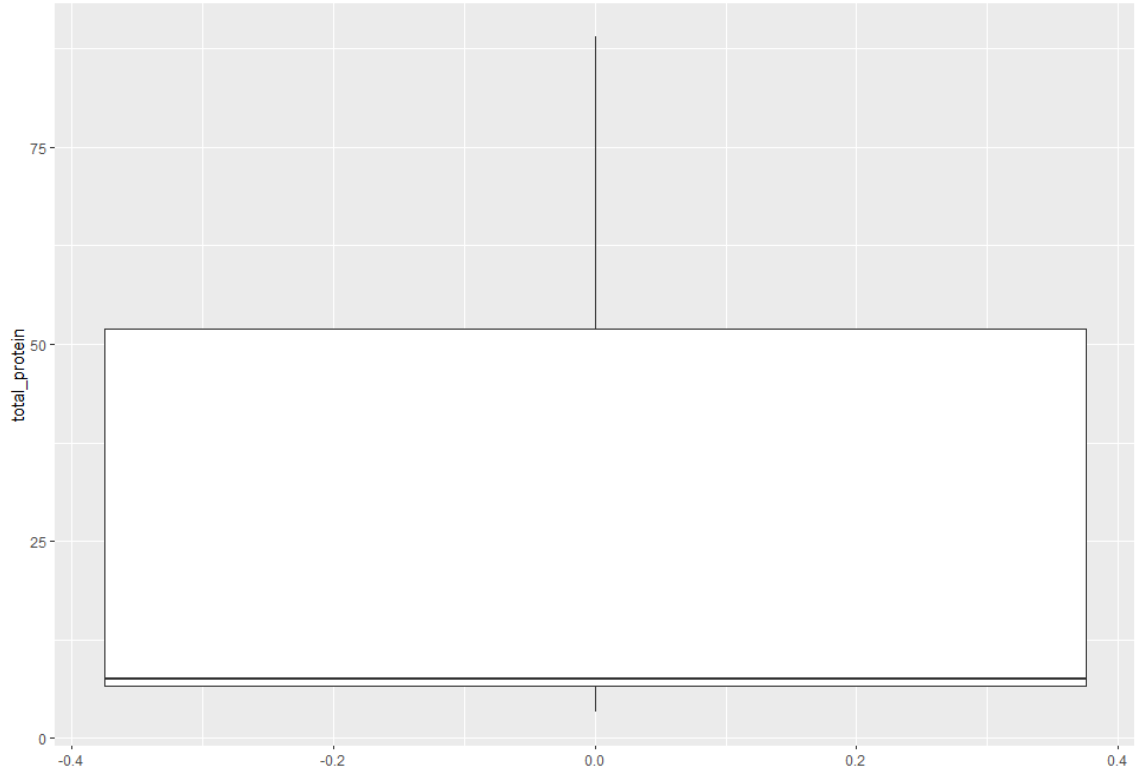


Figura 20 - Box-plot total_protein

Critério de identificação e substituição dos outliers:

Para identificar quais valores são caracterizados como outliers, calculou-se a Amplitude Inter Quartil (IRQ), que é obtida com a subtração do 3º Quartil pelo 1º Quartil.

Com esse valor foi possível definir o limite máximo e o limite mínimo conforme as fórmulas abaixo:

Limite Máximo
L_Máx = 3º Quartil + (IRQ × 1,5)

Qualquer registro em um determinado atributo que encontra-se acima do 3º quartil somada à uma vez e meia amplitude inter quartil, será considerado como outlier, e o valor deste registro será substituído pelo limite máximo.

Limite mínimo
L_Min = 1ºQuartil - (IRQ × 1,5)

Qualquer registro em determinado atributo que encontre-se abaixo do 1º quartil subtraído pela amplitude inter quartil multiplicada em uma vez e meia, será considerado um outlier, e valor deste registro será substituído pelo limite mínimo.

3.4. Normalização

Certos algoritmos de machine learning, tais como Support Vector Machine (SVM) e K Nearest Neighbor(KNN) são sensíveis a escala de dados. Para estes algoritmos, a distância entre os pontos dos dados é de extrema importância. Para que o modelo não seja influenciado pela ordem de grandeza dos dados, foi implementado no R o parâmetro *"scaler"* (presente na função *"preprocess"*) que normaliza os atributos discretos presentes nas bases de Treino e Teste em uma escala de 0 e 1, conforme exemplifica a equação matemática abaixo:

$$\frac{xi - \min(x)}{\max(x) - \min(x)}$$

Onde:
xi = registro
min(x) = o valor mínimo de um atributo
max(x) = o valor máximo de um atributo

3.5. Pré-Processamento na Base de Teste

Todos os parâmetros de pré-processamento descritos acima (substituição de "missing values", tratamento de "outliers", exclusão de atributos que não agregam ao modelo e normalização) também foram aplicados na base de teste, em processos separados, para que ambos mantivessem a mesma formatação, permitindo assim sua compatibilidade na etapa de treinamento do modelo preditivo.

3.6. Balanceamento

Após a análise dos gráficos abaixo, foi identificado um desbalanceamento na base de treino. O atributo “outcome” possui mais registros com valor “lived” do que “died” e “euthanized”. Portanto, será necessário realizar o balanceamento da base de treino para que o modelo preditivo não seja tendencioso em classificar os dados.

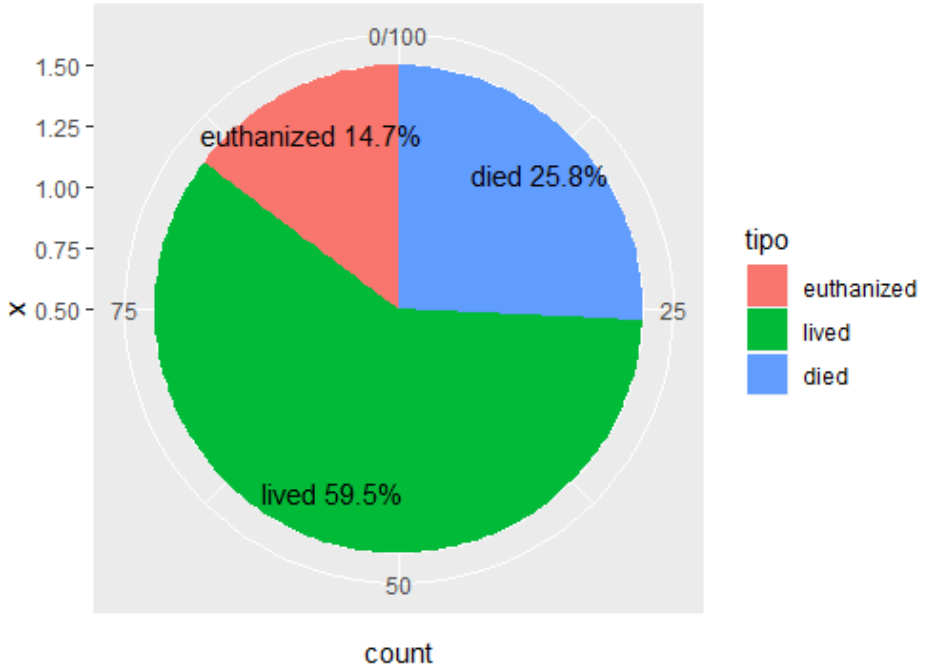


Figura 21 - Gráfico de Pizza outcome original

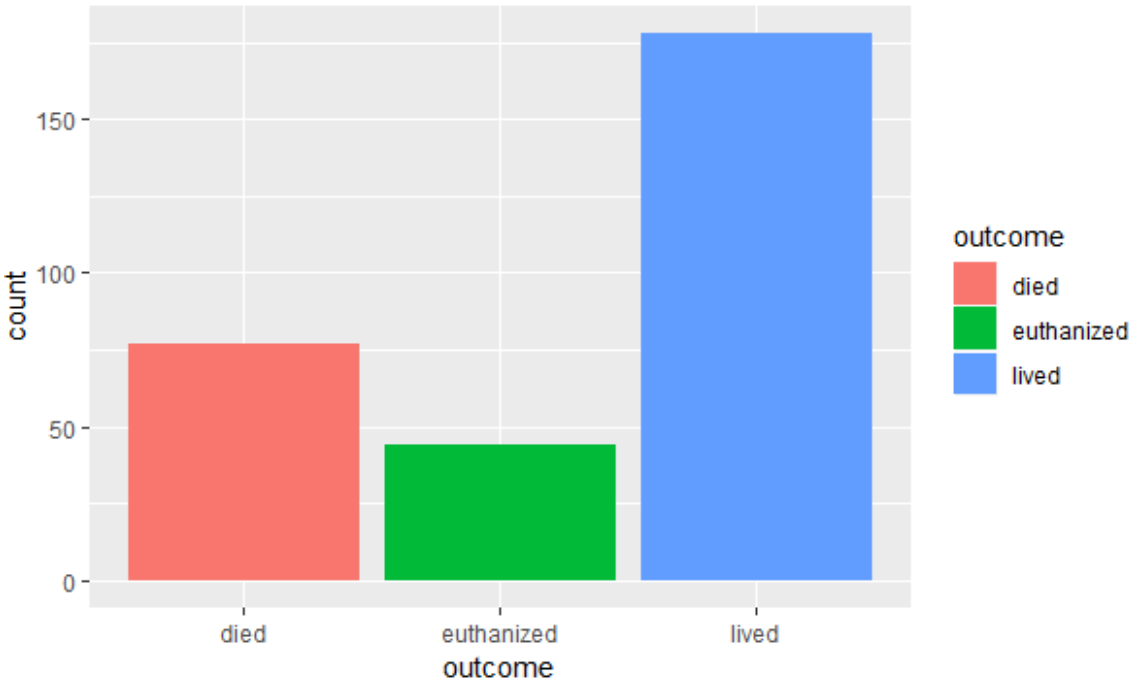


Figura 22 - Gráfico de Barras outcome original

A técnica escolhida para realizar o balanceamento foi o SMOTE. Técnica na qual amostras sintéticas da classe rara são geradas ao perturbar um atributo por vez por um valor dentro da diferença entre os k vizinhos mais próximos.

Base de Treino após a aplicação do SMOTE:

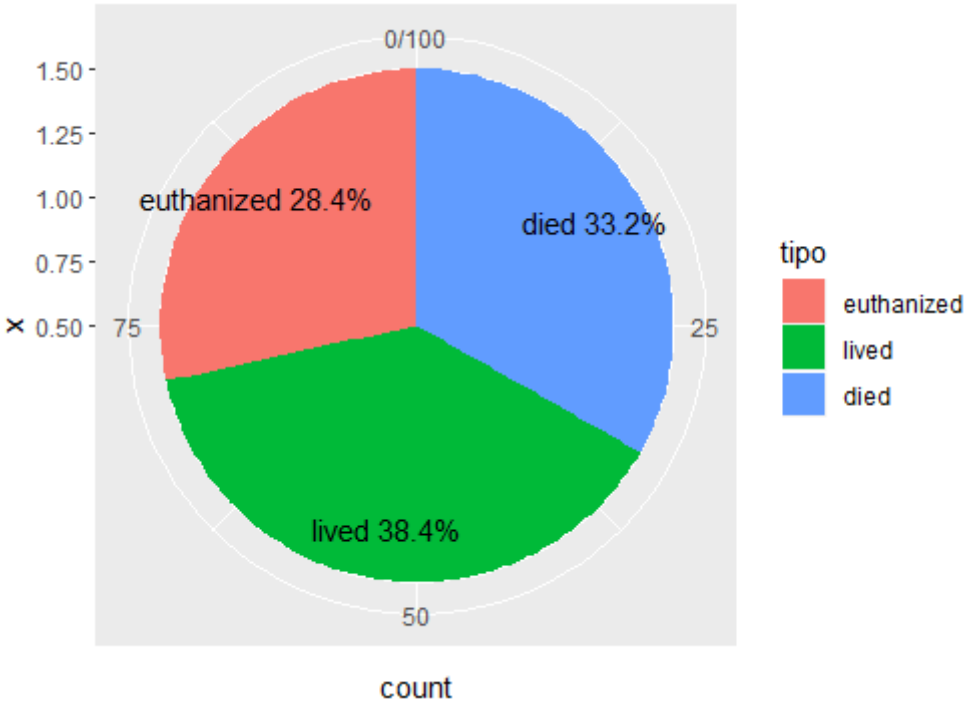


Figura 23 - Gráfico de Pizza outcome Balanceado

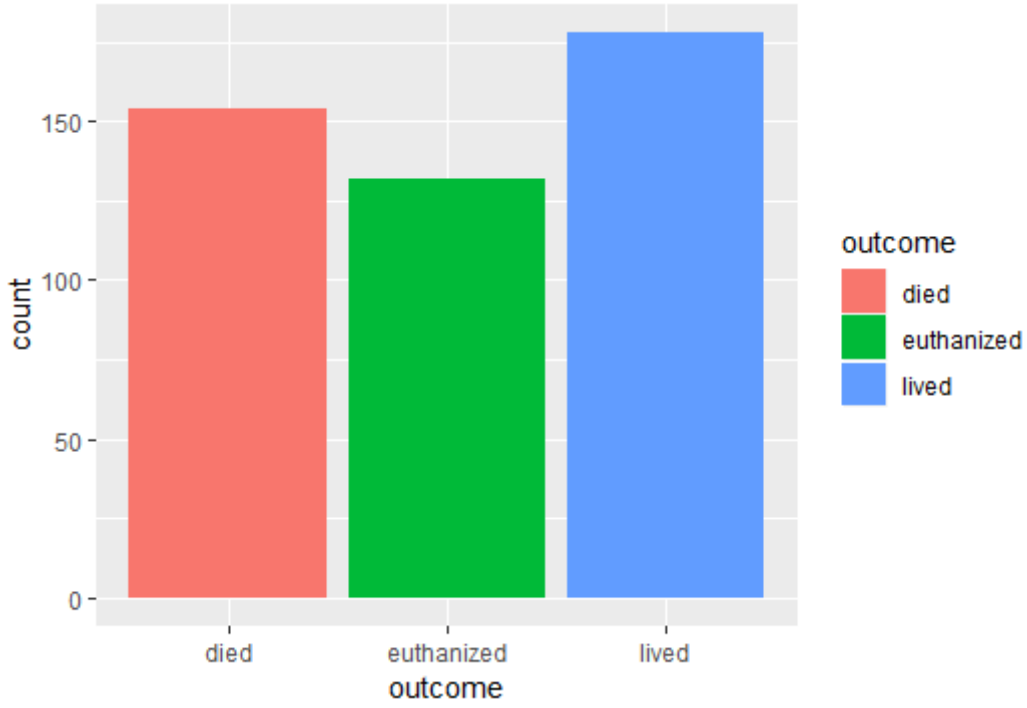


Figura 24 - Gráfico de Barras outcome balanceado

Observação:

Para avaliar o impacto da realização do SMOTE na base de treino nas previsões dos modelos, foi gravado em memória uma versão da base de treino sem a aplicação do SMOTE (variável: *"treino_sem_balanceamento"*) e outra base com a aplicação do SMOTE (variável: "*treino"*).

3.7. Redução da dimensionalidade

O método utilizado para a redução da dimensionalidade foi a Análise de Componentes Principais (*"Principal Component Analysis - PCA"*).

Premissas do PCA:

- Busca explicar o máximo possível da variância total.
- As variáveis precisam ser numéricas. Como o PCA calcula a variância total dos dados, ele só será aplicado para variáveis numéricas e não categóricas
- Analisa e agrupa toda a variância dentro das variáveis selecionadas em "Componentes Principais"
- Cada componente principal é uma combinação linear de todas as variáveis originais.
- Todos os componentes são ortogonais entre si. Portanto, não apresentam informações redundantes.
- Cada componente calculado representa uma nova variância a qual não havia sido explicada pelo componente anterior.
- Como os componentes principais criados pelo PCA são uma abstração do conjunto total de todos os dados, eles não são fáceis de interpretar ou mapear.

Metodologia:

O PCA reduz o número de atributos presentes em um conjunto de dados ao agrupar toda a variância presente em todas as variáveis do conjunto de dados em componentes (chamados de Componentes Principais ou *"Principal Components" - PC*). Como cada novo componente calculado, possui cada vez menos variância, que não havia sido captada anteriormente por um Componente anterior, a significância dos componentes finais se torna mínima.

A chave para a redução das variáveis, que é o objetivo do PCA, é determinar em que ponto do conjunto de componentes criado há variância suficiente captada, que não seja mais necessário considerar mais nenhum componente.

Portanto, o limite de corte ("thresh") estabelecido foi de 95% da variância captada nos componentes principais. Desse modo, os componentes finals que resentam 5% da variância não captada pelos Componentes Principais foram descartadas do dataframe.

Dummy Coding

Antes de realizar o PCA, foi necessário realizar o Dummy Coding nas bases de Treino e Teste. Esse procedimento foi realizado em cópias da base de Treino com o SMOTE e na cópia da base de teste.

Após a realização do Dummy Coding, a base de dados apresentou 43 variáveis, sendo 42 numéricas e 1 categórica (outcome).

Resultado do PCA:

```
PCA needed 32 components to capture 95 percent of the variance
```

Figura 25 - Resultado PCA

O PCA conseguiu captar 95% da variância total de todos os atributos do conjunto de dados com apenas 32 componentes principais, conforme explicitou a Figura 25.

	V1
rectal_temp	0.008782486
pulse	0.016836313
respiratory_rate	0.025667161
packed_cell_volume	0.033599974
total_protein	0.047756581
surgery_yes	0.089091948
age_young	0.102818956
temp_of_extremities_cool	0.144843721
temp_of_extremities_normal	0.175275366
temp_of_extremities_warm	0.187780296
peripheral_pulse_increased	0.191724347
peripheral_pulse_normal	0.233411141
peripheral_pulse_reduced	0.273163506
mucous_membrane_bright_red	0.289547584
mucous_membrane_dark_cyanotic	0.304174529

mucous_membrane_normal_pink	0.343116396
mucous_membrane_pale_cyanotic	0.366456188
mucous_membrane_pale_pink	0.392186315
capillary_refill_time_less_3_sec	0.428797210
capillary_refill_time_more_3_sec	0.465129497
pain_depressed	0.494559893
pain_extreme_pain	0.517651945
pain_mild_pain	0.556263756
pain_severe_pain	0.579603548
peristalsis_hypermotile	0.595698729
peristalsis_hypomotile	0.638166734
peristalsis_normal	0.646853462
abdominal_distention_none	0.687554836
abdominal_distention_severe	0.711628350
abdominal_distention_slight	0.739374431
nasogastric_tube_significant	0.754887066
nasogastric_tube_slight	0.793935786
nasogastric_reflux_more_1_liter	0.817275578
nasogastric_reflux_none	0.853886474
rectal_exam_feces_decreased	0.877472423
rectal_exam_feces_increased	0.886486040
rectal_exam_feces_normal	0.911040787
abdomen_distend_small	0.932354336
abdomen_firm	0.940050895
abdomen_normal	0.953777903
abdomen_other	0.963116828
surgical_lesion_yes	1.000000000

Figura 26 - Variância acumulada

A imagem acima demonstra a variância acumulada por atributo no conjunto de dados. Tendo sido o corte do PCA foi estabelecido em 95%, as variáveis abaixo foram desconsideradas do modelo:

- surgical_lesion_yes
- abdomen_other
- abdomen_normal

4. Treinando modelos

Premissas:

Todos os testes foram realizados com a “seed” igual a “1” para que não houvesse uma variação nos resultados.

Os modelos foram treinados em três bases distintas:

- Base 1 - “treino_sem_balanceamento”
  Base de treino sem a aplicação do SMOTE

- Base 2 - “treino”
  Base de treino com a aplicação do SMOTE

- Base 3 - “treinopca”

Base de treino com a aplicação do SMOTE e PCA

Será feita uma avaliação por meio de duas medidas estatísticas: acurácia e kappa para analisar qual das bases de dados obteve melhor resultado nos modelos treinados.

O objetivo final é que um modelo consiga prever o maior número de ocorrências na base de teste da forma mais assertiva possível. Dessa maneira, o modelo que obtiver o resultado mais próximo de: Died = 23, Euthanized = 13 e Lived = 53 será considerado o melhor modelo.

```
table(teste$outcome)

    died euthanized      lived
      23         13         53
```

Figura 27 - Contagem dos valores dos atributos de “outcome” na base de teste

4.1. Modelos na Base 1:

4.1.1. Árvore de Decisão:

Base 1: Base de treino sem a aplicação do SMOTE

Matriz de Confusão:

predictionsDtree_normal	died	euthanized	lived
died	15	1	1
euthanized	2	10	4
lived	6	2	48

Figura 28 - Matriz de confusão Árvore de Decisão - Base 1

Interpretação dos Resultados:

Died:

- 15 registros foram previstos corretamente.
- 1 registro foi previsto como a categoria “Died”, mas que na realidade é categorizado como “Euthanized”
- 1 registro foi previsto como a categoria “Died”, mas que na realidade é categorizado “Lived”

Euthanized:

- 10 registros foram previstos corretamente.
- 2 registros foram previstos como a categoria “Euthanized”, mas que na realidade são categorizados como “Died”.
- 4 registros foram previstos como a categoria “Euthanized”, mas que na realidade são categorizados como “Lived”.

Lived:

- 48 registros foram previstos corretamente.
- 2 registros foram previstos como a categoria “Lived”, mas que na realidade são categorizados como “Euthanized”.
- 6 registros foram previstos como a categoria “Lived”, mas que na realidade são categorizados como “Died”.

Avaliação:

Acuracia Tree Normal	Kappa Tree Normal
0.8202247	0.6729444

Figura 29 - Resultados modelo Árvore de Decisão - Base 1

4.1.2. SVM:

Matriz de Confusão:

predictionsSVM_normal	died	euthanized	lived
died	16	3	3
euthanized	0	4	0
lived	7	6	50

Figura 30 - Matriz de confusão SVM - Base 1

Interpretação dos Resultados:

Died:

- 16 registros foram previstos corretamente.
- 3 registros foram previstos como a categoria “Died”, mas que na realidade são categorizados como “Euthanized”.
- 3 registros foram previstos como a categoria “Died”, mas que na realidade são categorizados como “Lived”.

Euthanized:

- 4 registros foram previstos corretamente.

Lived:

- 50 registros foram previstos corretamente.
- 6 registros foram previstos como a categoria “Lived”, mas que na realidade são categorizados como “Euthanized”.
- 7 registros foram previstos como a categoria “Lived”, mas que na realidade são categorizados como “Died”.

Avaliação:

Acuracia SVM Normal	Kappa SVM Normal
0.7865169	0.5797714

Figura 31 - Resultados modelo SVM - Base 1

4.1.3. Random:

Matriz de Confusão:

predictionsForest_normal	died	euthanized	lived
died	23	0	0
euthanized	0	13	0
lived	0	0	53

Figura 32 - Matriz de confusão Random Forest - Base 1

A matriz de confusão presente na imagem acima atingiu o resultado esperado abaixo:

- 23 registros previstos corretamente como Died
- 13 registros previstos corretamente como Euthanized
- 53 registros previstos corretamente como Lived

Avaliação:

Acuracia FOREST Normal	Kappa FOREST Normal
1	1

Figura 33 - Resultados modelo Random Forest - Base 1

Gráfico Erro Out-of-Bag (OOB):

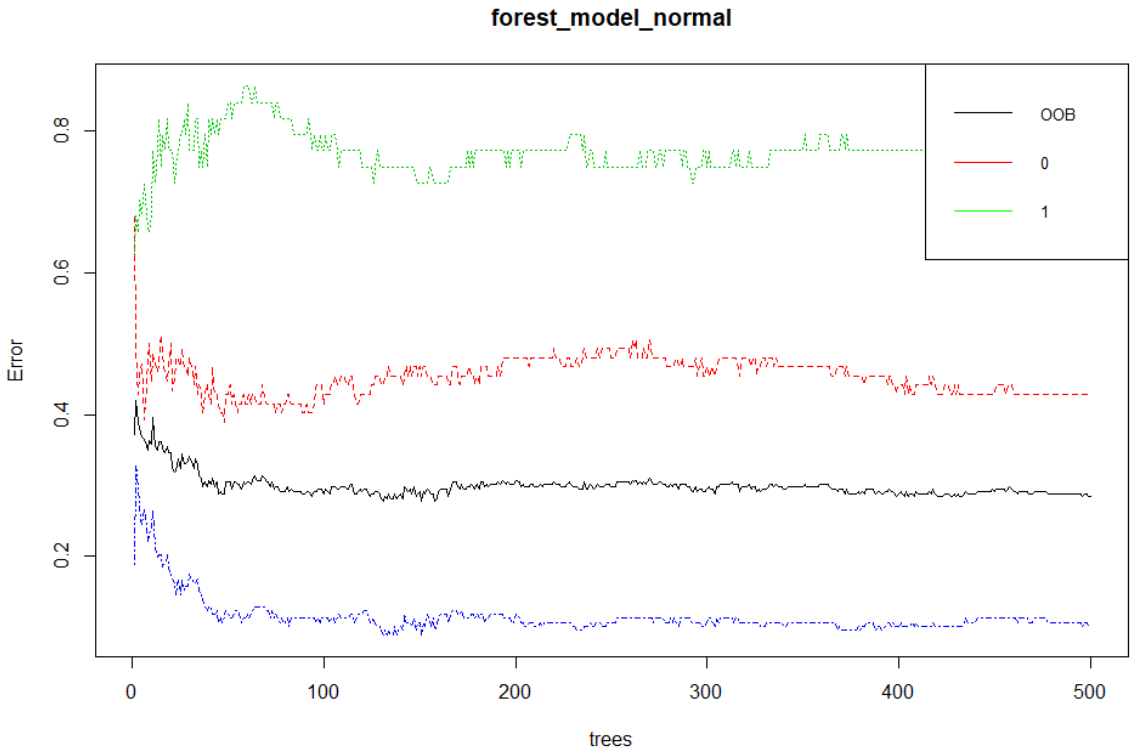


Figura 34 - Gráfico Erro OOB Random Forest - Base 1

Existe um "tradeoff" presente no modelo de Random Forest, onde quanto mais árvores são criadas pelo modelo, mais chances o modelo tem de prever corretamente o resultado. Porém, mais recursos de máquina ele irá exigir.

Por padrão, foram produzidas 500 árvores de decisão para a geração deste modelo de floresta. Entretanto, como o modelo de floresta exige um grande poder computacional, recomenda-se a análise do gráfico de erro OOB acima para mensurar o número de árvores necessárias para que o modelo seja capaz de acertar as previsões. Para o modelo em questão, ao analisar o gráfico OOB, percebe-se que entre 0 a 100 árvores a linha preta, que representa a função do erro OOB, ainda permanece instável. Apenas a partir de 200 árvores a linha começa a se estabilizar. Para obter um limite de segurança, o ideal seria treinar o modelo com 300 árvores.

Variáveis mais significativas para o modelo:

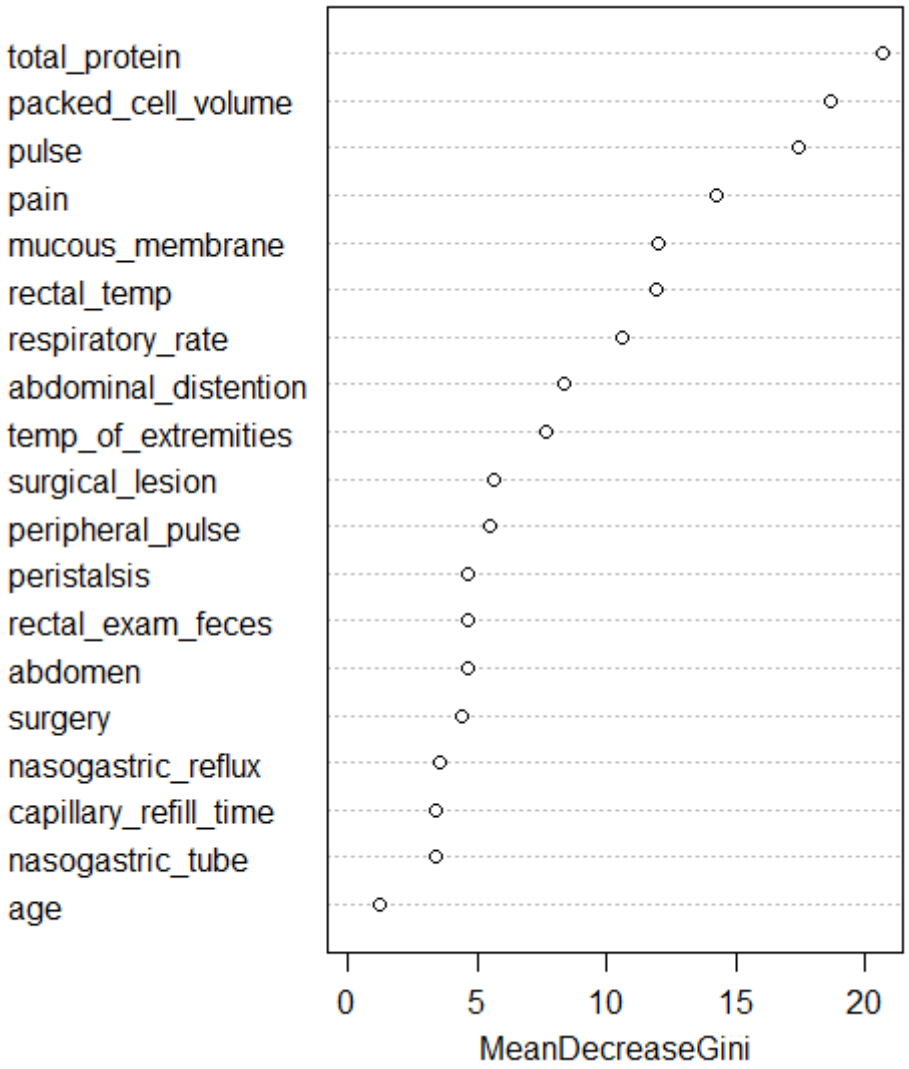


Figura 35 - Índice de Gini Random Forest - Base 1

Ao observar o Índice de Gini presente na imagem acima, constata-se que as cinco variáveis mais significativas para o modelo são:

- total_protein
- packed_cell_volume
- pulse
- pain
- mucous_mebrane

4.1.4. Conclusão modelos:

De todos os modelos treinados com a base 1, o modelo que apresentou os melhores resultados foi o Random Forest.

4.2. Modelos na Base 2:

Base 2: Base de treino com a aplicação do SMOTE

4.2.1. Árvore de Decisão:

Matriz de Confusão:

predictionsDtree_smote	died	euthanized	lived
died	17	0	4
euthanized	3	12	6
lived	3	1	43

Figura 36 - Matriz de confusão Árvore de Decisão - Base 2

Interpretação dos Resultados:

Died:

- 17 registros foram previstos corretamente.
- 4 registros foram previstos como a categoria “Died”, mas que na realidade são categorizados como “Lived”.

Euthanized:

- 12 registros foram previstos corretamente.
- 3 registros foram previstos como a categoria “Euthanized”, mas que na realidade são categorizados como “Died”.
- 6 registros foram previstos como a categoria “Euthanized”, mas que na realidade são categorizados como “Lived”.

Lived:

- 43 registros foram previstos corretamente.
- 1 registro foi previsto como a categoria “Lived”, mas que na realidade é categorizados como “Euthanized”.
- 3 registros foram previstos como a categoria “Lived”, mas que na realidade são categorizados como “Died”.

Avaliação:

Acuracia TREE Smote	Kappa TREE Smote
0.8089888	0.6762944

Figura 37 - Resultados modelo Árvore de Decisão - Base 2

4.2.2. SVM:

Matriz de Confusão:

predictionsSVM_smote	died	euthanized	lived
died	21	1	6
euthanized	0	8	4
lived	2	4	43

Figura 38 - Matriz de confusão SVM - Base 2

Interpretação dos Resultados:

Died:

- 21 registros foram previstos corretamente.
- 1 registro foi previsto como a categoria “Died”, mas que na realidade é categorizados como “Euthanized”.
- 6 registros foram previstos como a categoria “Died”, mas que na realidade são categorizados como “Lived”.

Euthanized:

- 8 registros foram previstos corretamente.
- 4 registros foram previstos como a categoria “Euthanized”, mas que na realidade são categorizados como “Lived”

Lived:

- 43 registros foram previstos corretamente.
- 4 registros foram previstos como a categoria “Lived”, mas que na realidade são categorizados como “Euthanized”.
- 2 registros foram previstos como a categoria “Lived”, mas que na realidade são categorizados como “Died”.

Avaliação:

Acuracia SVM Smote	Kappa SVM Smote
0.8089888	0.6655615

Figura 39 - Resultados modelo SVM - Base 2

4.2.3. Random Forest:

Matriz de Confusão:

predictionsForest_smote	died	euthanized	lived
died	23	0	0
euthanized	0	13	0
lived	0	0	53

Figura 40 - Matriz de confusão Random Forest - Base 2

A matriz de confusão presente na imagem acima atingiu o resultado esperado abaixo:

- 23 registros previstos corretamente como Died
- 13 registros previstos corretamente como Euthanized
- 53 registros previstos corretamente como Lived

Avaliação:

Acuracia Forest Smote	Kappa Forest Smote
1	1

Figura 41 - Resultados modelo Random Forest - Base 2

Gráfico Erro Out-of-Bag (OOB):

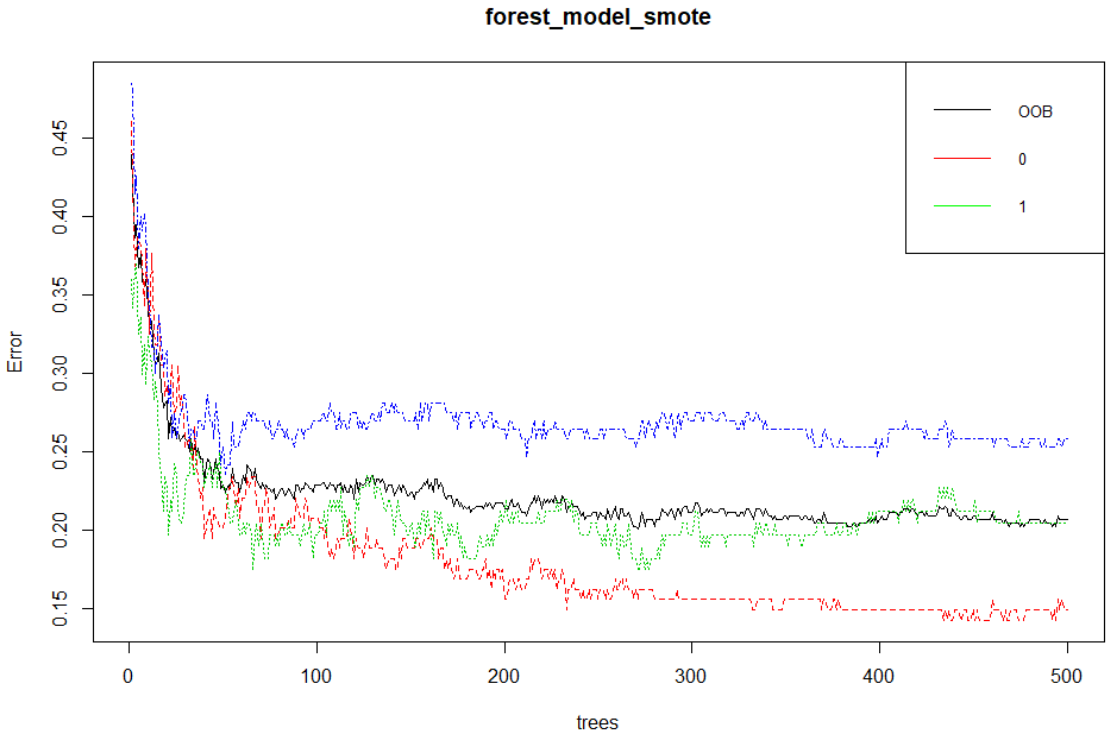


Figura 42 - Gráfico Erro OOB Random Forest - Base 2

Ao analisar o gráfico do erro OOB acima, o número ótimo de árvores são 400 árvores.

Variáveis mais significativas para o modelo:

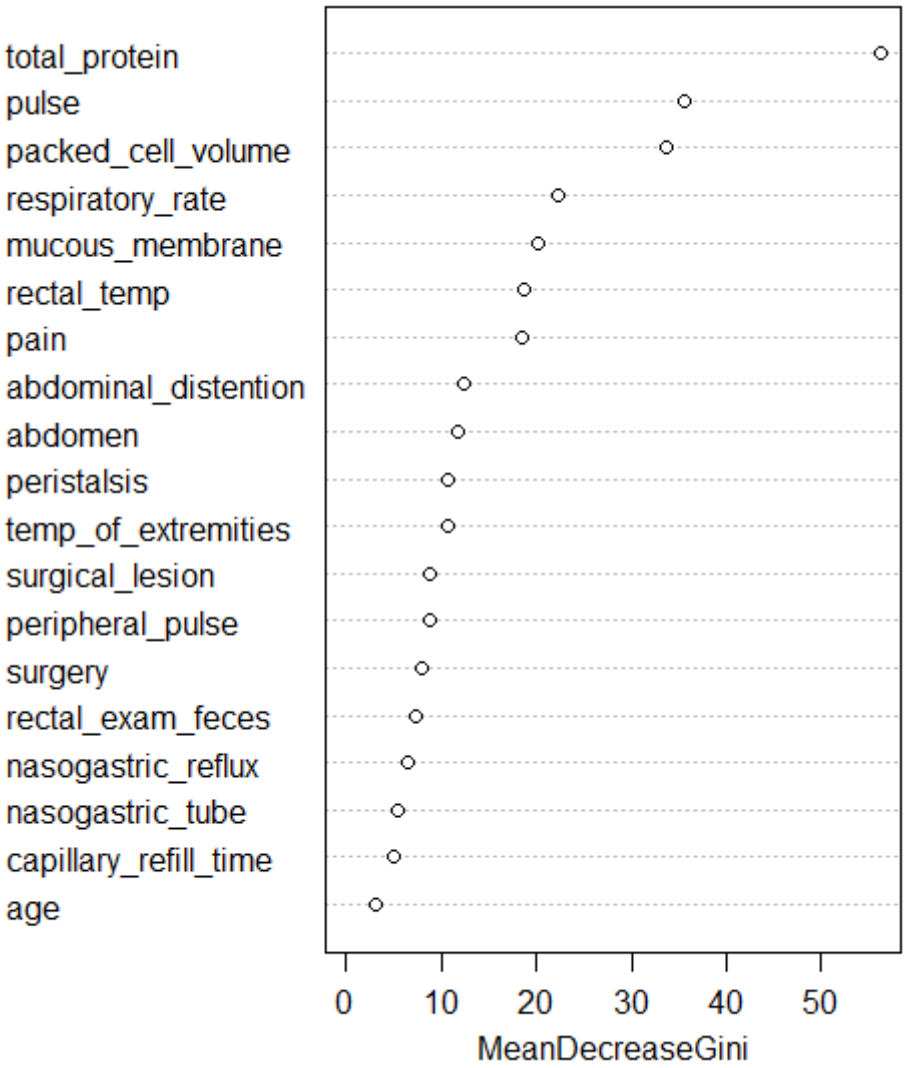


Figura 43 - Índice de Gini Random Forest - Base 2

Ao observar o Índice de Gini presente na imagem acima, constata-se que as três variáveis mais significativas para o modelo são:

- total_protein
- pulse
- packed_cell_volume

4.2.4. Conclusão modelos:

De todos os modelos treinados com a Base 2, o modelo que apresentou os melhores resultados foi o Random Forest.

4.3. Modelos na Base 3:

Base 3: Base de treino com a aplicação do SMOTE e PCA

4.3.1. Árvore de Decisão:

Matriz de Confusão:

```
predictionsDtree_PCA died euthanized lived
           died         17           3     6
           euthanized    1           9     3
           lived         5           1    44
>
```

Figura 44 - Matriz de confusão Árvore de Decisão - Base 3

Interpretação dos Resultados:

Died:

- 17 registros foram previstos corretamente.
- 3 registros foram previstos como a categoria “Died”, mas que na realidade são categorizados como “Euthanized”.
- 6 registros foram previstos como a categoria “Died”, mas que na realidade são categorizados como “Lived”.

Euthanized:

- 9 registros foram previstos corretamente.
- 1 registros foram previstos como a categoria “Euthanized”, mas que na realidade são categorizados como “Died”.
- 3 registros foram previstos como a categoria “Euthanized”, mas que na realidade são categorizados como “Lived”.

Lived:

- 44 registros foram previstos corretamente.
- 1 registro foi previsto como a categoria “Lived”, mas que na realidade é categorizados como “Euthanized”.
- 5 registros foram previstos como a categoria “Lived”, mas que na realidade são categorizados como “Died”.

Avaliação:

Acuracia TREE PCA	Kappa TREE PCA
0.7865169	0.624556

Figura 45 - Resultados modelo Árvore de Decisão - Base 3

4.3.2. SVM:

Matriz de Confusão:

```
predictionsSVM_PCA died euthanized lived
          died        22          0     0
          euthanized   0         12     3
          lived        1          1    50
>
```

Figura 46 - Matriz de confusão SVM - Base 3

Interpretação dos Resultados:

Died:

- 22 registros foram previstos corretamente.

Euthanized:

- 12 registros foram previstos corretamente.
- 3 registros foram previstos como a categoria “Euthanized”, mas que na realidade são categorizados como “Lived”

Lived:

- 50 registros foram previstos corretamente.
- 1 registro foi previsto como a categoria “Lived”, mas que na realidade é categorizados como “Euthanized”.
- 1 registro foi previsto como a categoria “Lived”, mas que na realidade é categorizados como “Died”.

Avaliação:

Acuracia SVM PCA	Kappa SVM PCA
0.9438202	0.9003136

Figura 47 - Resultados modelo SVM - Base 3

4.3.3. Random Forest:

Matriz de Confusão:

```
predictionsForest_PCA died euthanized lived
          died         23          0     0
          euthanized    0         13     0
          lived         0          0    53
```

Figura 48 - Matriz de confusão Random Forest - Base 3

A matriz de confusão presente na imagem acima atingiu o resultado esperado abaixo:

- 23 registros previstos corretamente como Died
- 13 registros previstos corretamente como Euthanized
- 53 registros previstos corretamente como Lived

Avaliação:

Acuracia FOREST PCA	Kappa FOREST PCA
1	1

Figura 49 - Resultados modelo Random Forest - Base 3

Gráfico Erro Out-of-Bag (OOB):

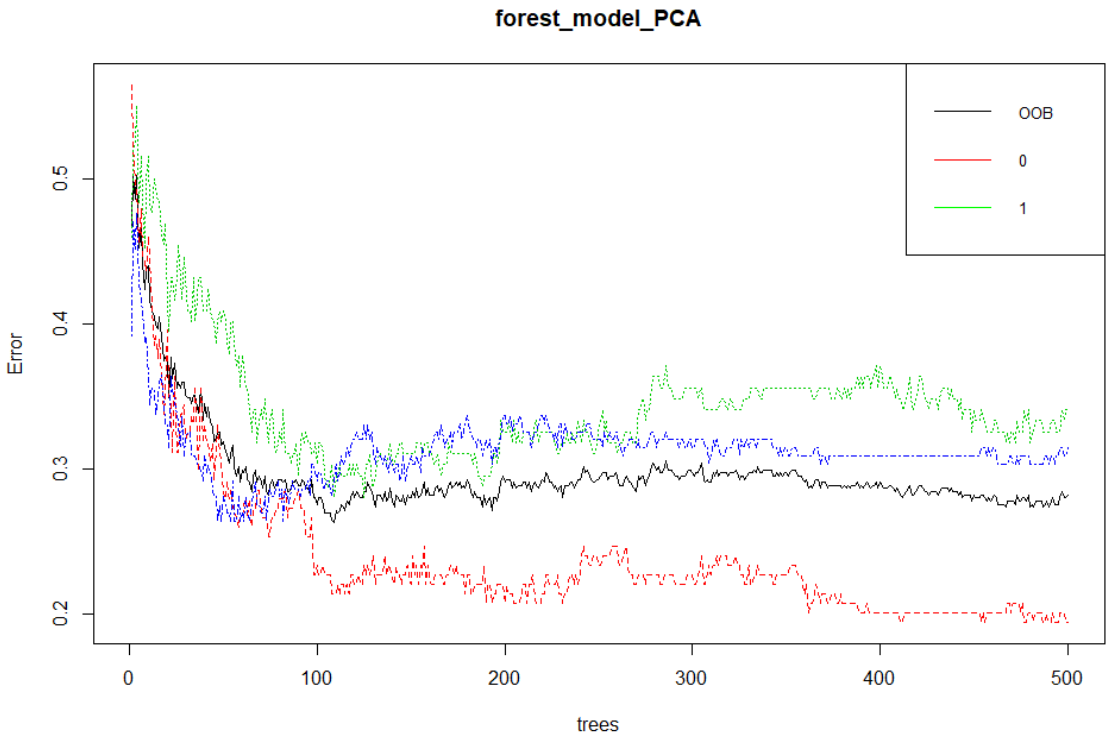


Figura 50 - Gráfico Erro OOB Random Forest - Base 3

Ao analisar o gráfico do erro OOB acima, o número ótimo de árvores para se treinar o modelo são de 400 árvores.

Variáveis mais significativas para o modelo:

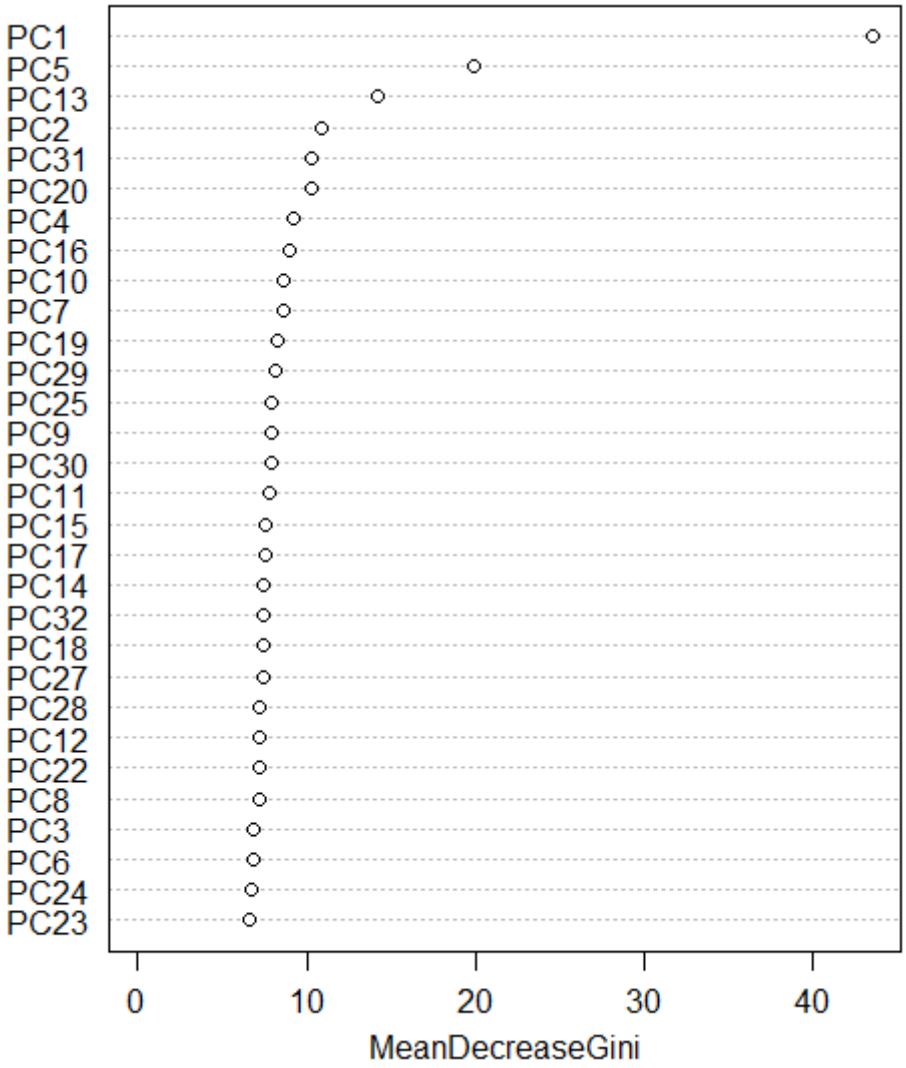


Figura 51 - Índice de Gini Random Forest - Base 3

Como mencionado anteriormente no ponto “3.6 - Redução da Dimensionalidade”, os componentes principais criados pelo método de PCA são uma abstração do conjunto total de todos os dados. Logo, torna-se difícil interpretar quais variáveis são mais significativas para o modelo.

4.3.4. Conclusão modelos:

De todos os modelos treinados com a base 3, o modelo que apresentou os melhores resultados foi o Random Forest.

5. Escolha do Melhor modelo

O modelo de Random Forest foi escolhido para prever o diagnóstico de cavalos, pois este modelo obteve os melhores resultados, tanto de acurácia quanto de kappa, nas três bases nas quais ele foi testado.

Em relação às Bases, como todas obtiveram um resultado de 100% no modelo de Random Forest, o grupo optou por escolher a base de treino com o SMOTE como modelo final, pois a base de treino sem o balanceamento, apesar de não provocar nenhum viés na capacidade preditiva do modelo de Random Forest, não é uma boa prática trabalhar com bases desbalanceadas. Enquanto o modelo de PCA, não permite uma visualização das variáveis mais significativas para o modelo.

Além disso, optou-se por treinar o modelo utilizando 400 árvores, como já justificado acima.

6. Anexos

Versão em R do script:
Projeto_DM_Script_Leticia_Aranha_Matheus_Rangel.R

Versão em .txt script:
Projeto_DM_Script_Leticia_Aranha_Matheus_Rangel_txt.txt