

UNIVERSIDADE DE SÃO PAULO – USP
ESCOLA DE ENGENHARIA DE LORENA

THIAGO GONÇALVES GUIMARÃES LOPES

**Análise comparativa de modelos de *machine learning* na
predição de câncer de pele**

Lorena – SP

2020

THIAGO GONÇALVES GUIMARÃES LOPES

**Análise comparativa de modelos de *machine learning* na
predição de câncer de pele**

Trabalho de conclusão de curso apresentado à Escola de Engenharia de Lorena – Universidade de São Paulo como requisito parcial para conclusão da Graduação do curso de Engenharia Física.

Orientadora: Prof. Dr. Fabiano Fernandes Bargos.

Lorena – SP

Dezembro de 2020

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE

Ficha catalográfica elaborada pelo Sistema Automatizado
da Escola de Engenharia de Lorena,
com os dados fornecidos pelo(a) autor(a)

Lopes, Thiago Gonçalves Guimarães
Análise comparativa de modelos de machine
learning na predição de câncer de pele / Thiago
Gonçalves Guimarães Lopes; orientador Fabiano
Fernandes Bargos. - Lorena, 2020.
121 p.

Monografia apresentada como requisito parcial
para a conclusão de Graduação do Curso de Engenharia
Física - Escola de Engenharia de Lorena da
Universidade de São Paulo. 2020

1. Machine learning. 2. Câncer de pele. 3. Teste
binário e multiclasse. I. Título. II. Bargos, Fabiano
Fernandes, orient.

Dedico este trabalho a minha família
e a Taynara, minha namorada,
que me apoiaram durante toda a
graduação e me motivam a cada dia a ser melhor.

Agradecimentos

O número de pessoas o qual gostaria de mencionar e agradecer a esta altura é maior do que eu possa contar. Então, dedico este agradecimento do trabalho de graduação a todos aqueles que contribuíram para minha formação.

Primeiramente, agradeço aos meus pais, Ivone e Bráulio e minha irmã Ana Beatriz. Sempre me motivaram, ouviram minhas reclamações, dramas sobre a graduação e mesmo assim continuaram me incentivando a continuar. Se me tornei a pessoa que sou hoje, eles são os responsáveis, meu alicerce e meu bem mais precioso.

Durante minha infância, adolescência e até hoje fiz inúmeras e valiosas amizades. Devido à distância e à falta de contato, acabamos nos afastando, mas guardo todos em meu peito. Todavia, gostaria de mencionar algumas destas parcerias: aos meus antigos amigos de república, Carlos, Antônio, Murilo, Brunello, Soares, Diego e Hugo. Agradeço pelos momentos de diversão e cafés ruins compartilhados.

Aos meus amigos de sala, João Felipe, Ronaldo, Rennan, Nakashima, William, Mikael, Ygor, Pedro, Cainan e tantas outras amizades ocasionadas pela faculdade. Vocês além de me proporcionarem momentos de risadas e felicidades, também me acompanharam e passaram ao meu lado nos momentos de surto pré/ pós/ durante prova, madrugando estudando; por isso que sou grato a vocês.

Tenho enorme gratidão por todos os professores que contribuíram para meu aprendizado e formação, enaltecendo especialmente meu orientador Fabiano, Luiz Eleno e Durval. Também dedico a todos os funcionários da EEL, que com todo empenho e dedicação tornam o ambiente mais acolhedor.

Por fim, agradeço mais que especialmente minha namorada Taynara. Sem o apoio, carinho e incentivo dela, eu não estaria alcançando meus objetivos hoje. Ela que me motiva, incentiva, inspira e acredita em mim. Independente das ideias malucas e devaneios na madrugada, ela sempre me deu suporte e confiança.

*"Success is not final.
Failure is not fatal.
It is the courage to
continue that counts."*

Wiston Churchil (1874 - 1965).

Resumo

LOPES, T. G. G. **Análise comparativa de modelos de *machine learning* na predição de câncer de pele.** 2020. 121 p. Monografia (Trabalho de Graduação de Engenharia Física) – Escola de Engenharia de Lorena, Universidade de São Paulo, Lorena, 2020.

Existem diversas aplicações de Inteligência Artificial na área da medicina, especialmente na predição de doenças. Uma das comorbidades que mais acometem a sociedade atualmente é o câncer, sendo o de pele o tipo mais comum. Esse trabalho tem como objetivo empregar diferentes técnicas de *Machine Learning* para predição de câncer de pele em imagens de lesões. Dados coletados por instituições médicas e dermatológicas foram utilizados para extração de características que, após tratamentos de normalização e SMOTE (*Synthetic Minority Oversampling Technique*), são usados para treinamento dos modelos de aprendizado de máquina supervisionados. Os hiperparâmetros de cada um foram obtidos pelo processo de validação cruzada. Os algoritmos de *machine learning* avaliados nesta monografia são: Regressão Logística; *Naive Bayes*; Árvore de Decisões; Floresta Aleatória; Árvores Extremamente Aleatória; KNN (*K-Nearest Neighbors*); RNN (*Radius Nearest Neighbors*); *Adaptive Boosting*; *Gradient Boosting*; SVM (*Support Vector Machine*) e RNA-MLP (Redes Neurais Artificiais *Multi Layer Perceptron*). Para avaliar o desempenho dos modelos os dados foram divididos na proporção 80% para treino e 20% para testes. Os resultados para cada um variavam em acurácia, sendo os algoritmos com melhores resultados o *Gradient Boosting*, SVM com kernel linear e polinomial, Floresta Aleatória, Regressão Logística, Árvores Extremamente Aleatórias e RNA-MLP. Os valores de acurácia variaram de 82,44% até 95% para testes binários e multiclasse.

Palavras-chave: *Machine learning*. Câncer de pele. Teste Binário e Multiclassee.

Abstract

LOPES, T. G. G. **Comparative analysis of machine learning models in skin cancer prediction.** 2020. 121 p. Monograph (Undergraduate Thesis in Engineering Physics) – Engineering School of Lorena, University of São Paulo, Lorena, 2020.

There are several applications of Artificial Intelligence in the field of medicine, especially in disease prediction. One of the comorbidities that most affect society today is cancer, with skin being the most common type. This work aims to employ different Machine Learning techniques to predict skin cancer in lesion images. Data collected by medical and dermatological institutions were used to extract characteristics that, after normalization and SMOTE (Synthetic Minority Oversampling Technique) treatments, are used to train supervised machine learning models. The hyperparameters of each were obtained by the cross-validation process. The machine learning algorithms evaluated in this monograph are: Logistic Regression; Naive Bayes; Decision Tree; Random Forest; Extremely Random Trees; KNN (K-Nearest Neighbors); RNN (Radius Nearest Neighbors); Adaptative Boosting; Gradient Boosting; SVM (Support Vector Machine) and ANN-MLP (Artificial Neural Networks Multi Layer Perceptron). To evaluate the performance of the models, the data were divided in the proportion 80% for training and 20% for tests. The results for each one varied in accuracy, with the algorithms with the best results being Gradient Boosting, SVM with linear and polynomial kernel, Random Forest, Logistic Regression, Extremely Random Trees and RNA-MLP. Accuracy values ranged from 82.44% to 95% for binary and multiclass tests.

Keywords: Machine learning. Skin cancer. Binary and Multiclass tests.

Lista de abreviaturas e siglas

ML	<i>Machine Learning</i>
SMOTE	<i>Synthetic Minority Oversampling Technique</i>
IBM	<i>International Business Machines Corporation</i>
INCA	Instituto Nacional do Câncer
GBM	Grupo Brasileiro de Melanoma
CIE	<i>Commission International de l'Éclairage</i>
SVM	<i>Support Vector Machine</i>
RNA	Redes Neurais Artificiais
MLP	<i>Multi Layer Perceptron</i>
BGD	<i>Batch Gradient Descent</i>
SGD	<i>Stochastic Gradient Descent</i>
NB	<i>Naive Bayes</i>
NBG	<i>Naive Bayes Gaussian</i>
NBB	<i>Naive Bayes Bernoulli</i>
MAP	<i>Maximum a posteriori</i>
AdaBoost	<i>Adaptative Boosting</i>
GB	<i>Gradiente Boosting</i>
KNN	<i>K - Nearest Neighbors</i>
RNN	<i>Radius Nearest Neighbors</i>
VP	Verdadeiro Positivo
VN	Verdadeiro Negativo
FP	Falso Positivo
FN	Falso Negativo

ROC	<i>Receiving Operating Characteristics</i>
MCC	Coeficiente de Correlação de Mathews
AUC	<i>Area Under Curve</i>
PCA	<i>Principal Component Analysis</i>

List of symbols

I	Intensity of pixel
f	Unknown function
C_i	Class i of sample
\hat{f}	Estimated function
\mathbf{x}	Vector of characteristics
y_i	Value of the label assigned to sample \mathbf{x}_i
\mathbf{w}	Vector of weights associated with the vector of characteristics
η	Learning rate
$\sigma(\cdot)$	Sigmoid function
J	Cost function
P	Probability function
H	Shannon entropy
G	[Gini index]
r_j	Error rate associated with weight
h	Estimated function (similar to \hat{f})
M	Number of estimators
l	Loss function
$d(\mathbf{x}_i, \mathbf{x}_j)$	Euclidean distance between points \mathbf{x}_i and \mathbf{x}_j
$\ \mathbf{w}\ $	Norm of the weight vector
ζ	Margin variable of the SVM model
\mathfrak{I}	New space with linearly separable attributes (SVM model)
Φ	Mapping function between spaces X and \mathfrak{I}
K	Kernel

f_a	Função de ativação
δ_l	Erro entre as camadas de uma rede neural perceptron
Ac	Acurácia
Pr	Precisão
Rec	Revocação
F_1	Medida F para $\beta = 1$
F_β	Medida F
B_S	Acurácia Balanceada
ROC_S	Pontuação ROC
MCC	Coeficiente de Correlação de Matthews
F_M	Métrica Final

Listas de ilustrações

Figura 1 – Composição celular da pele humana	4
Figura 2 – Exemplo de sinais na pele como a) Nevo comum e b) Nevo atípico e c) Melanoma	5
Figura 3 – Características para identificação de lesões malignas na pele	6
Figura 4 – Exemplo de análise de bordas	7
Figura 5 – Exemplo de bordas incomuns em nevos e melanomas	8
Figura 6 – Exemplo de coloração em pintas e melanomas na pele: a) Marrom Claro; b) Marrom Escuro; c) Preto; d) Branco; e) Cinza; f) Vermelho; g) Vários tons;	8
Figura 7 – Medição de simetria de uma lesão	9
Figura 8 – Exemplo de nevos simétricos	9
Figura 9 – Exemplo de nevos ou melanomas assimétricos	9
Figura 10 – Esquema de estruturas dermatoscópicas	10
Figura 11 – Exemplo de tipicidade da rede de pigmentos a) típico e b) atípico . . .	10
Figura 12 – Exemplo de áreas amorfas regulares	11
Figura 13 – Exemplo de áreas amorfas irregulares	11
Figura 14 – Exemplo de pontos e glóbulos em melanomas	11
Figura 15 – Exemplo de estrias em melanomas e carcinomas	12
Figura 16 – Resumo da metodologia <i>radiomics</i>	12
Figura 17 – Exemplos de atributos morfológicos	13
Figura 18 – Posicionamento dos eixos em imagens digitais	14
Figura 19 – Representação de uma imagem em escala RGB como canais independentes	15
Figura 20 – Diagrama do Sistema CIE XYZ	16
Figura 21 – Diferentes percepções da mesma imagem ao analisar canais de cores diferentes	17
Figura 22 – Representação do sistema RGB como cubo unitário 3D	18
Figura 23 – Sistema RGB como cubo unitário e definição de uma cor a partir da combinação do sistema RGB como cubo unitário 3D	18
Figura 24 – Exemplo de sobreajuste	20
Figura 25 – SMOTE - Synthetic Minority Oversampling Technique	21
Figura 26 – <i>K-Fold Cross Validation</i>	23
Figura 27 – Procedimento usado para treinamento e avaliação dos modelos de ML utilizando a Validação Cruzada e <i>Grid Search</i>	23
Figura 28 – Função Logística	25
Figura 29 – <i>Gradient Descent</i>	26
Figura 30 – <i>Falhas no Gradient Descent</i>	26

Figura 31 – Diferença na minimização do a) Batch e b) Stochastic Gradient Descent	27
Figura 32 – Exemplo de árvore de decisão para tarefa de classificação proposta	30
Figura 33 – Classificação da árvore de decisões com diferentes valores de profundidade	32
Figura 34 – Exemplo de um treinado do <i>Ensemble Learning</i>	33
Figura 35 – Exemplo de um treinamento usando <i>AdaBoost</i>	34
Figura 36 – Diferença de resultados no modelo do <i>AdaBoost</i> variando a taxa de aprendizado	35
Figura 37 – Impacto do valor de K no Exemplo de KNN	38
Figura 38 – Impacto do valor de R no Exemplo de KNN	39
Figura 39 – Exemplo de classificação utilizando o modelo SVM linear com margens rígidas	41
Figura 40 – Exemplo de classificação utilizando o modelo SVM linear com margens suaves	41
Figura 41 – Neurônio Biológico	45
Figura 42 – Neurônio Artificial	45
Figura 43 – Exemplos de funções de ativação	46
Figura 44 – Unidade Linear com função de ativação <i>Step</i>	47
Figura 45 – Componentes de uma RNA-MLP	48
Figura 46 – RNA-MLP de 1 camada oculta	49
Figura 47 – Matriz de confusão para um problema binário	50
Figura 48 – Gráfico ROC	53
Figura 49 – Fluxo de trabalho desenvolvido para esta monografia	56
Figura 50 – Exemplo de lesão e máscara presente no conjunto de dados	57
Figura 51 – Diferença de histogramas ao utilizar ou não as máscaras para delimitar a região de interesse	60
Figura 52 – Exemplo de um <i>boxplot</i>	62
Figura 53 – Diferença na distribuição dos dados utilizando: a) sem a adaptação b) com adaptação	63
Figura 54 – Contorno da imagem binarizada	64
Figura 55 – Circunferência em torno da mancha.	65
Figura 56 – Divisão do contorno.	66
Figura 57 – Elipse e contorno da pinta	66
Figura 58 – Elipse e contorno da pinta	67
Figura 59 – Cálculo da amplitude inferior e superior	68
Figura 60 – Resumo da metodologia utilizada na etapa de treinamento dos modelos	71
Figura 61 – Variáveis categóricas do conjunto PH2: a) antes da transformação boleana; b) após transformação boleana	72
Figura 62 – Diferenças nas amostras com a utilização do SMOTE	72
Figura 63 – Distribuição das imagens do conjunto PH2 por Critério	76

Figura 64 – Distribuição das imagens do conjunto HAM10000 por gênero	76
Figura 65 – Distribuição das imagens do conjunto HAM10000 por localização	77
Figura 66 – Distribuição das imagens do conjunto HAM10000 por idade	77
Figura 67 – Distribuição das variáveis morfológicas para o conjunto das imagens do conjunto PH2	78
Figura 68 – Análise no espaço bidimensional das variáveis morfológicas normalizadas do conjunto PH2	78
Figura 69 – Distribuição das variáveis dos Canais RGB e Cinza para o conjunto das imagens do conjunto PH2	79
Figura 70 – Análise no espaço bidimensional das variáveis de coloração dos canais RGB e Cinza normalizadas do conjunto PH2	80
Figura 71 – Distribuição de algumas variáveis das representações de cores HSV, HSL, CIE L*a*b e CIE Lu'v'	80
Figura 72 – Análise no espaço bidimensional das variáveis de coloração dos canais HSV normalizadas do conjunto PH2	81
Figura 73 – Análise no espaço bidimensional das variáveis de coloração dos canais HSL normalizadas do conjunto PH2	81
Figura 74 – Análise no espaço bidimensional das variáveis de coloração dos canais CIE Lu'v' normalizadas do conjunto PH2	82
Figura 75 – Análise no espaço bidimensional das variáveis de coloração dos canais CIE L*a*b normalizadas do conjunto PH2	82
Figura 76 – Análise no espaço bidimensional de todas as variáveis utilizadas neste projeto	82
Figura 77 – Matriz de confusão e curva ROC obtida para o o modelo <i>Gradient Boosting</i> amostra 1 teste binário	90
Figura 78 – Matriz de confusão obtida para o o modelo SVM - Linear para amostra 1 teste multiclasse	91
Figura 79 – Matriz de confusão e curva ROC obtida para o o modelo SVM - Linear amostra 2 teste binário	92
Figura 80 – Matriz de confusão obtida para o o modelo <i>Random Forest</i> para amostra 2 teste multiclasse	92
Figura 81 – Matriz de confusão e curva ROC obtida para o o modelo Regressão Logística amostra 3 teste binário	93
Figura 82 – Matriz de confusão obtida para o o modelo Árvores Extremamente Aleatórias para amostra 3 teste multiclasse	94
Figura 83 – Matriz de confusão e curva ROC obtida para o o modelo RNA-MLP com 3 camadas ocultas amostra 4 teste binário	94
Figura 84 – Matriz de confusão obtida para o o modelo RNA-MLP com 2 camadas ocultas para amostra 4 teste multiclasse	95

Lista de tabelas

Tabela 1 – Estimativa de novos casos de câncer no Brasil em 2020	3
Tabela 2 – Mortes por câncer de Pele no Brasil em 2018	3
Tabela 3 – Tipos de Kernels avaliados na otimização das SVM	43
Tabela 4 – Tabela de métricas para problemas multiclasse	54
Tabela 5 – Variáveis semânticas do conjunto de dados PH2	58
Tabela 6 – Variáveis do conjunto de dados HAM10000	59
Tabela 7 – Hiperparâmetros e seus valores analisados em cada modelo de ML . . .	70
Tabela 8 – Resumo dos resultados obtidos para o modelo SVM	88
Tabela 9 – Resumo dos resultados obtidos para os RNA-MLP	89
Tabela 10 – Resultados obtidos para amostra 1 teste binário	104
Tabela 11 – Resultados obtidos para amostra 1 teste multiclasse	106
Tabela 12 – Resultados obtidos para amostra 2 teste binário	108
Tabela 13 – Resultados obtidos para amostra 2 teste multiclasse	110
Tabela 14 – Resultados obtidos para amostra 3 teste binário	113
Tabela 15 – Resultados obtidos para amostra 3 teste multiclasse	115
Tabela 16 – Resultados obtidos para amostra 4 teste binário	117
Tabela 17 – Resultados obtidos para amostra 4 teste multiclasse	119

Sumário

1	INTRODUÇÃO	1
1.1	Objetivos	2
2	FUNDAMENTAÇÃO TEÓRICA	3
2.1	Câncer de pele	3
2.1.1	Estatísticas sobre a doença no Brasil	3
2.1.2	Tipos de cânceres de pele	4
2.1.3	Atributos analisados para identificação da doença	6
2.1.3.1	Diâmetro	7
2.1.3.2	Borda	7
2.1.3.3	Coloração	7
2.1.3.4	Simetria	8
2.1.3.5	Estruturas dermatoscópicas	10
2.2	Radiomics	12
2.3	Visão Computacional	14
2.3.1	Imagens Digitais	14
2.3.2	Sistemas de Representação de Cores	15
2.4	Aprendizado de Máquina	18
2.4.1	Uma visão geral	18
2.4.2	Modelos	24
2.4.2.1	Régressão Logística	24
2.4.2.2	<i>Gaussian e Bernoulli Naive Bayes</i>	27
2.4.2.3	Árvore de Decisões	29
2.4.2.4	Floresta Aleatória e Árvores Extremamente Aleatórias	32
2.4.2.5	<i>Adaptive Boosting e Gradient Boosting</i>	34
2.4.2.6	KNN e RNN	37
2.4.2.7	SVM	40
2.4.2.8	Redes Neurais Artificiais	44
2.4.3	Métricas de avaliação	50
3	METODOLOGIA	56
3.1	Fluxo de trabalho	56
3.2	Conjunto de dados	57
3.3	Extração das características	60
3.3.1	Variáveis de Coloração	63
3.3.2	Variáveis Morfológicas	64

3.4	Escolha dos modelos de ML	68
3.5	Obtenção dos hiperparâmetros	69
3.6	Treinamento dos modelos	71
3.7	Teste e avaliação de desempenho dos modelos	73
4	RESULTADOS	75
4.1	Apresentação dos dados dos conjuntos	75
4.2	Avaliação de desempenho dos modelos	83
4.2.1	Rregressão Logística	83
4.2.2	<i>Gaussian e Bernoulli Naive Bayes</i>	84
4.2.3	Árvore de Decisões	84
4.2.4	Floresta Aleatória	85
4.2.5	Árvores Extremamente Aleatória	85
4.2.6	KNN e RNN	86
4.2.7	<i>Adaptative Boosting</i>	86
4.2.8	<i>Gradient Boosting</i>	87
4.2.9	SVM	87
4.2.10	Redes Neurais Artificiais	88
4.3	Discussão geral dos modelos	89
5	CONCLUSÃO	97
	REFERÊNCIAS	99
	ANEXOS	103
	ANEXO A – RESULTADOS PARA CADA MODELO	104

1 Introdução

A técnicas de Aprendizado de Máquina, *Machine Learning* (ML) do inglês, tem inúmeras utilizações, uma delas é na Medicina. Ferramentas computacionais baseadas nestes conceitos são utilizadas para controle de epidemias, monitoramento de pacientes e auxílio na predição de doenças (FACELI et al., 2011).

Todavia, como ressalta Faceli et al. (2011), as resoluções destes problemas não se baseiam em um modelo de "caixa-preta". Diagnosticar uma pessoa saudável como doente é melhor do que diagnosticar um doente como saudável. Tendo em vista que o resultado deste procedimento impacta diretamente a vida humana.

A área da saúde é muito beneficiada pelo desenvolvimento de técnicas computacionais, as quais são responsáveis pela ampliação da capacidade e da qualidade dos diagnósticos de diversas doenças, através da análise de imagens digitais (BOTELHO, 2007). Estas imagens aliadas à implementação de algoritmos de ML, proporcionam novas técnicas para diagnosticar uma doença.

Nessa concepção, esta monografia se resume em realizar um estudo sobre predição de câncer de pele através de imagens digitais e avaliar modelos de *machine learning* na classificação desta doença. Esta análise contempla dois conjuntos de dados fornecidos por Mendonça et al. (2013) e Tschandl (2018), de domínio público, contendo uma série de imagens de lesões dermatológicas, máscaras de interesse e um arquivo com suas descrições. O primeiro contém 200 imagens divididas em 3 classes: nevo comum, nevo atípico e melanoma. Já o segundo, apresenta 10015 imagens divididas em 7 classes: lesões tipo ceratose benigna, dermatofibroma, nevos melanocíticos, lesões vasculares, carcinoma epidermoide, carcinoma basocelular e melanoma . Utilizando conceitos médicos abordados na Seção 2.1. Importante informações estatísticas serão retiradas a partir das imagens. Em seguida, através de uma análise dos dados obtidos, características pertinentes ao problema serão selecionadas e estas serão utilizadas para treinamento e validação dos modelos de ML.

Além da escolha de melhores atributos para serem utilizados como parâmetros no treinamento dos modelos, buscou-se analisar a performance dos resultados de cada algoritmo para diferentes tratamentos de dados. Introduziu-se neste processo a Validação Cruzada, Padronização dos Dados e *Synthetic Minority Oversampling Technique* (SMOTE) para determinação dos melhores hiperparâmetros em cada caso e identificação de mudanças significativas nos resultados gerados.

Por fim, esse trabalho considera os seguintes modelos de aprendizado de máquina: Regressão Logística; *Naive Bayes*; Árvore de Decisões; Floresta Aleatória; Árvores Ex-

tremamente Aleatória; KNN (*K-Nearest Neighbors*); RNN (*Radius Nearest Neighbors*); *Adaptive Boosting*; *Gradient Boosting*; SVM *Support Vector Machine* e Rede Neural Artificial. Estes modelos foram avaliados e comparados através da utilização da validação cruzada e das métricas exploradas na Seção 2.4.3.

1.1 Objetivos

Este trabalho tem por objetivo principal realizar um estudo sobre a capacidade de modelos de *machine learning* em classificar imagens médicas relacionadas à câncer de pele. Tendo como objetivos específicos:

- 1) Extrair características de seleção dos conjuntos de dados
- 2) Analisar a importância destes atributos para o problema
- 3) Treinar modelos de *Machine Learning* na predição de câncer de pele
- 4) Avaliar o desempenho destes modelos

2 Fundamentação Teórica

2.1 Câncer de pele

2.1.1 Estatísticas sobre a doença no Brasil

Câncer é o nome dado ao conjunto de doenças cuja principal característica é o crescimento autônomo e desordenado das células. Estas doenças invadem os tecidos e órgãos. Existem mais de 100 tipos de cânceres distintos e dentre eles, o câncer de pele é o tipo mais frequente na população brasileira ([INCA, 2020](#)).

Apesar do câncer ser uma doença muito temida pelas pessoas, já existem diversos tratamentos e em muitos casos, a cura é possível. Para que isso aconteça, deve-se realizar o diagnóstico nas fases iniciais da doença ([BOTELHO, 2007](#)).

No Brasil, o câncer de pele corresponde a cerca de 30% de todos os tumores malignos registrados no país. Desta categoria da doença, existem duas subdivisões: Melanoma e Não-Melanoma. Sendo este primeiro o mais grave, representando apenas 5% das neoplasias malignas do órgão, devido à sua alta possibilidade de provocar metástase (disseminação do câncer para outros órgãos) ([INCA, 2020](#)).

Na Seção [2.1.2](#) será realizado uma breve apresentação dos tipos de câncer de pele e suas principais diferenças.

Tabela 1 – Estimativa de novos casos de câncer de pele no Brasil em 2020

Tipo	Homem	Mulher	Total
Não-Melanoma	83.770	93.160	176.930
Melanoma	4.200	4.250	8.450

Fonte: ([INCA, 2020](#)).

Tabela 2 – Mortes por câncer de Pele no Brasil em 2018

Tipo	Homem	Mulher	Total
Não-Melanoma	1.358	971	2.329
Melanoma	1.038	753	1.791

Fonte: ([INCA, 2020](#)).

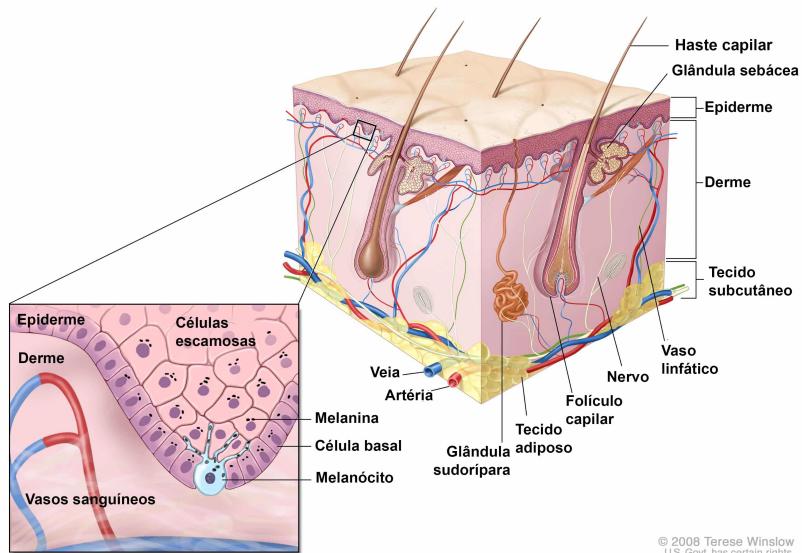
Como ilustra as Tabelas [1](#) e [2](#), esta doença ainda afeta grande parte da população brasileira, evidenciando a importância de criar meios alternativos para combater o câncer

de pele. Segundo INCA (2020) e GBM (2020) estima-se em média que o Brasil regista cerca de 5.560 novos casos de melanoma e 1.547 óbitos por ano, sendo esta doença comum em adultos e idosos, mas podendo ser encontrado em crianças e adolescentes.

2.1.2 Tipos de cânceres de pele

O câncer de pele é caracterizado pelo crescimento autônomo e desordenado das células que compõem a pele. Em alguns casos, estas células adquirem a capacidade de espalhar (metástase) e invadir outras orgãos e tecidos do corpo (MASSI; LEBOIT, 2014). Por isso que a detecção correta e em seus estágios iniciais se torna importante para o tratamento e sobrevivência do paciente.

Figura 1 – Composição celular da pele humana



Fonte: (GBM, 2020).

Como mostra a Figura 1, a pele humana é formada por mais de um tipo de célula, sendo o foco de estudo deste trabalho os seguintes tipos: Melanócitos, célula responsável pela produção de melanina no corpo; Célula Basal, célula constituinte da epiderme e responsável pela nutrição das células escamosas; e Queratinócitos, responsáveis pela produção de queratina da pele (MASSI; LEBOIT, 2014).

Com respeito a estes três tipos de células, diferentes tipos de cânceres de pele podem ser desenvolvidos, sendo os mais comuns: o Carcinoma Basocelular, o Carcinoma Espinocelular e o Melanoma (MASSI; LEBOIT, 2014). Segundo Botelho (2007):

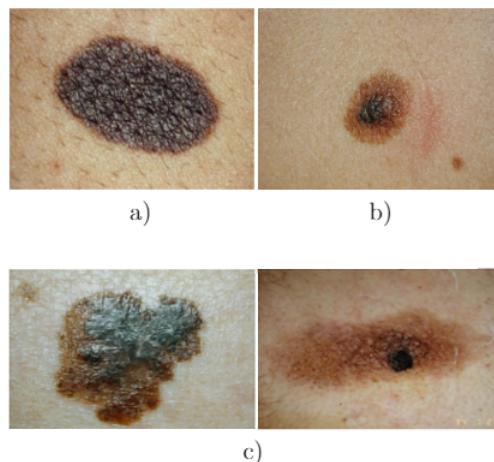
- **Carcinoma Basocelular:** Este é o tumor maligno cutâneo mais frequente dentre os três tipos, compreendendo cerca de 75 % dos diagnósticos. Contudo, é de baixa letalidade devido à facilidade de identificação e principalmente pelo fato de raramente

apresentar metástase. Esta doença origina-se nas células da camada basal da epiderme. É caracterizado pela formação de lesões (feridas ou nódulos) de crescimento lento. Normalmente localizada na região da cabeça e do pescoço, sujeitas a maior incidência solar.

- **Carcinoma Espinocelular:** Este tumor representa 20% dos diagnósticos de neoplasias cutâneas e origina-se nos queratinócitos da epiderme, podendo também surgir no tecido epitelial que cobre mucosas. É mais letal que o carcinoma basocelular, visto que apresenta maior possibilidade de realizar metástase. É caracterizada pela formação de feridas de rápida evolução no qual regularmente são acompanhadas de secreção e coceira.
- **Melanoma:** Esta é a neoplasia cutânea de menor incidência, porém de maior letalidade devido à dificuldade de identificação e à formação de metástases. Este tumor origina-se nos melanócitos e pode surgir em várias regiões do corpo. Uma de suas causas é a excessiva exposição solar, entretanto não a única. Apesar de mais letal entre os três tipos, se identificado precocemente e removido cirurgicamente, a probabilidade de cura terá um aumento significativo.

O maior problema na identificação destas doenças é a semelhanças destas com uma pinta (nevo) ou pequenas feridas no corpo, tornando suas detecções mais difíceis ao paciente, que não apresenta experiência nem olhar clínico do assunto. Segundo [GBM \(2020\)](#), o nevo é uma mancha que aparece na pele devido a produção de pigmento pelos melanócitos, mas de maneira natural e benigna. A maioria dos adultos possui entre 10 e 40 destes sinais pelo corpo, usualmente localizados da cintura para cima, em áreas que costumam ser mais expostas ao sol.

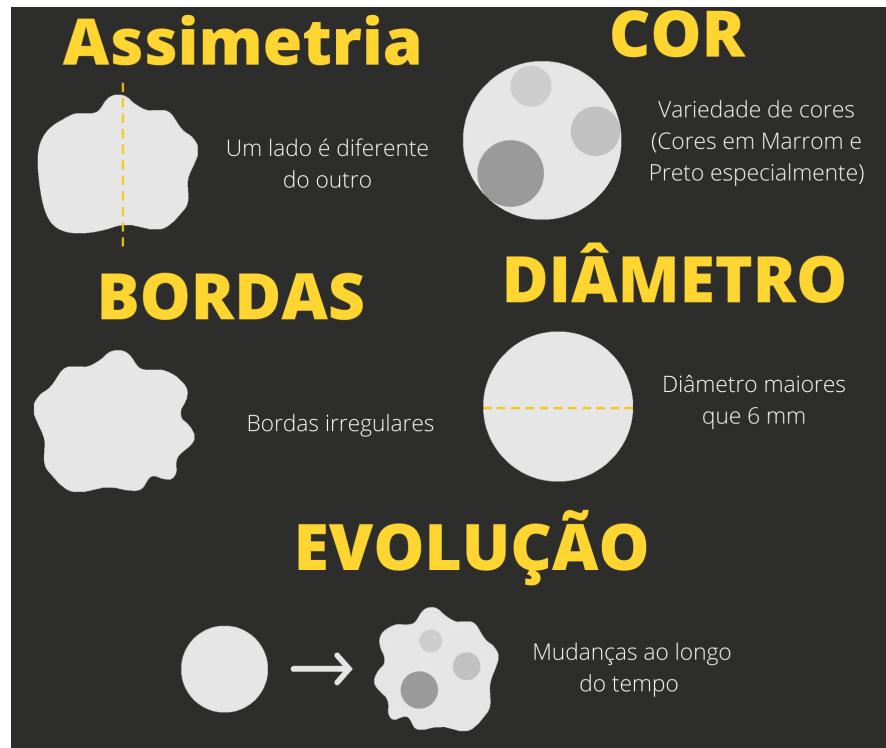
Figura 2 – Exemplo de sinais na pele como a) Nevo comum e b) Nevo atípico e c) Melanoma



Fonte: ([GBM, 2020](#)).

Segundo Massi e LeBoit (2014), existem diversos critérios utilizados para identificar e diferenciar nevos de lesões benignas ou malignas sendo alguma delas: localização da lesão, densidade de coloração, regiões de elevação e outros que serão abordadas na Seção 2.1.3. A Figura 2 exemplifica esta semelhança.

Figura 3 – Características para identificação de lesões malignas na pele



Fonte: Autor.

Ademais, para uma identificação mais simples e fácil de lesões tipo Melanoma ou Carcinoma, existem cinco características que são conhecidas popularmente como **ABCDE**, que facilitam a identificação destas lesões. Sendo elas: Assimetria, Bordas irregulares, Coloração variada ou muito escura, Diâmetros maiores que 6 mm e Evolução da pinta ao passar do tempo, como ilustrado na Figura 3 (BOTELHO, 2007; INCA, 2020; GBM, 2020).

2.1.3 Atributos analisados para identificação da doença

Como mencionado na Seção anterior, existe vários fatores que devem ser avaliados para identificar um câncer de pele. Alguns destes, que serão objeto de estudo desta monografia, são:

- Diâmetro
- Borda

- **Coloração**
- **Simetria**
- **Estruturas dermatoscópicas**

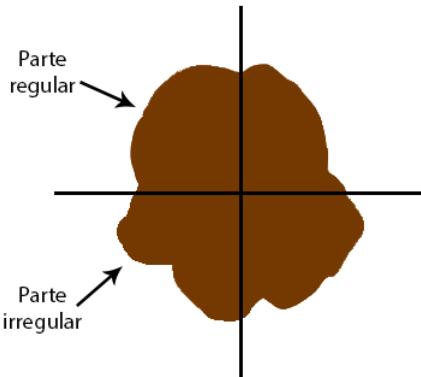
2.1.3.1 Diâmetro

O primeiro fator a ser analisado é o diâmetro da pinta. Considera-se como alta probabilidade de ser um melanoma sinais com tamanho maiores que 6 mm. Ressalva-se que há a possibilidade de existirem casos com diâmetros menores (MASSI; LEBOIT, 2014; INCA, 2020).

2.1.3.2 Borda

Outro fator importante é a borda da lesão. Quanto mais irregular for a borda, maior a probabilidade do sinal ser uma lesão maligna (Figura 4)(MASSI; LEBOIT, 2014; RUSSO et al., 2017).

Figura 4 – Exemplo de análise de bordas



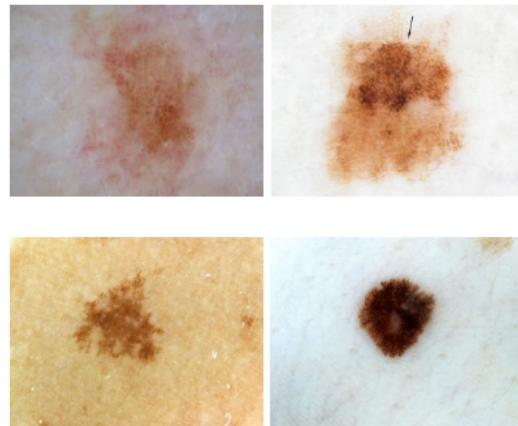
Fonte: (DERMNETNZ, 2008)[Modificado].

Segundo Russo et al. (2017), a presença de regiões pontiagudas ou com mudança drástica de nitidez, como apresentadas na Figura 5, também é um indicativo de melanoma.

2.1.3.3 Coloração

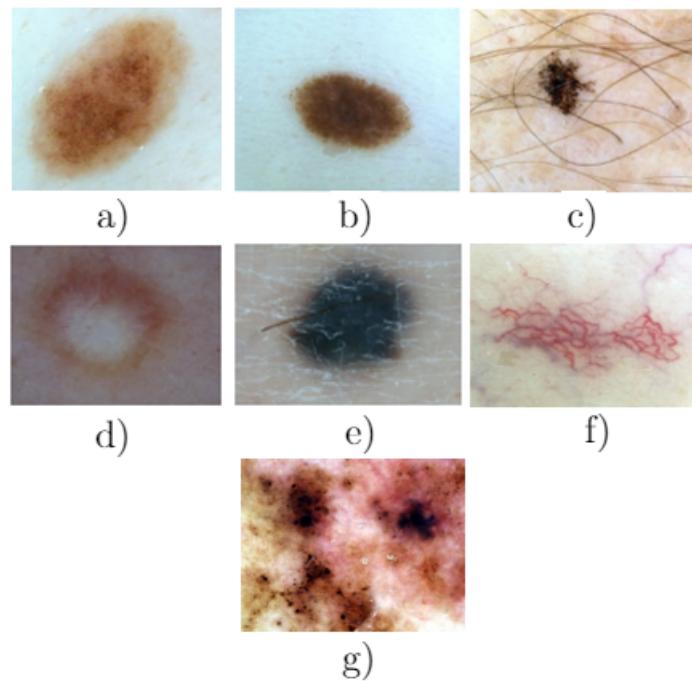
A existência de uma determinada cor ou vários tons é um forte indicativo de que a lesão seja um tumor maligno, na grande maioria dos casos. Da perspectiva médica, seis cores são consideradas importantes nesta análise, sendo elas: branco, vermelho, marrom claro, marrom escuro, cinza-azulado e preto (Ver Figura 6)(MENDONÇA et al., 2013).

Figura 5 – Exemplo de bordas incomuns em nevos e melanomas



Fonte: ([DERMNETNZ, 2008](#)).

Figura 6 – Exemplo de coloração em pintas e melanomas na pele: a) Marrom Claro; b) Marrom Escuro; c) Preto; d) Branco; e) Cinza; f) Vermelho; g) Vários tons;



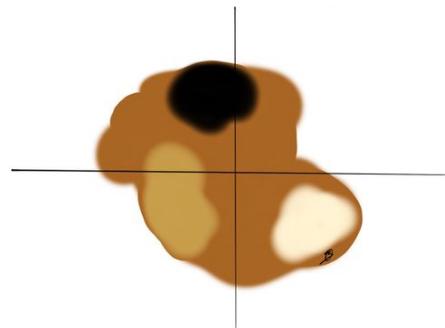
Fonte: ([DERMNETNZ, 2008](#)).

2.1.3.4 Simetria

Outro fator analisado em procedimentos para categorização de nevos ou melanomas, é quanto a simetria da mancha. Segundo [Massi e LeBoit \(2014\)](#), a presença de assimetria sugere a existência lesões malignas como melanoma.

Conforme ilustra a Figura 7, é marcado o ponto central da pinta e traçado duas retas perpendiculares entre si. Em seguida, é feito uma análise simples de simetria no

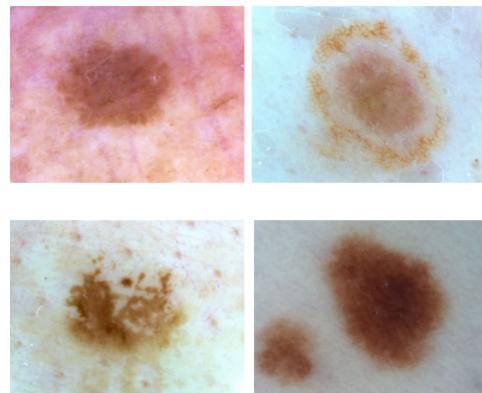
Figura 7 – Medição de simetria de uma lesão



Fonte: Autor.

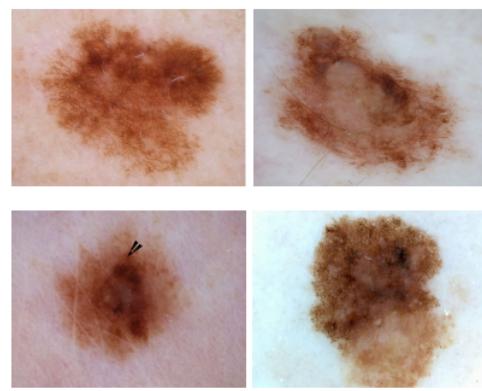
eixo horizontal e vertical. As imagens expostas nas Figuras 8 e 9 ilustram as relações de simetria em imagens reais.

Figura 8 – Exemplo de nevos simétricos



Fonte: ([DERMNETNZ, 2008](#)).

Figura 9 – Exemplo de nevos ou melanomas assimétricos

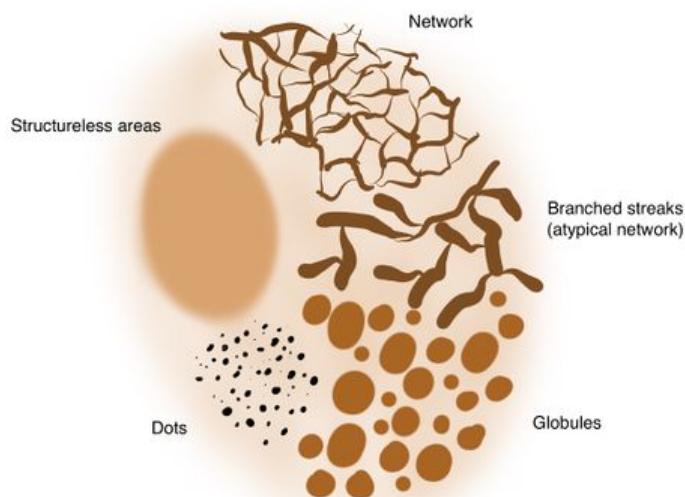


Fonte: ([DERMNETNZ, 2008](#)).

2.1.3.5 Estruturas dermatoscópicas

Outros fatores avaliados clinicamente são a presença e tipicidade de determinadas estruturas dermatoscópicas. Sendo estas estruturas: rede de pigmentos, área amorfa, pontos ou glóbulos e estrias (ver Fig. 10) (MASSI; LEBOIT, 2014; KITTLER et al., 2016; RUSSO et al., 2017).

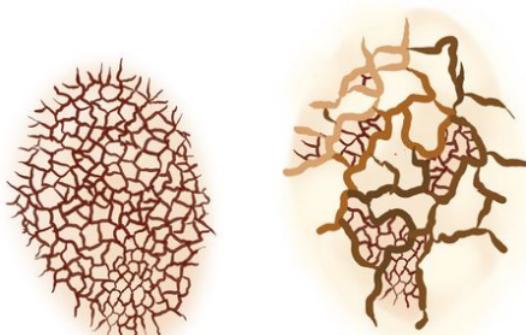
Figura 10 – Esquema de estruturas dermatoscópicas



Fonte: ([DERMNETNZ, 2008](#)).

A rede de pigmento consiste em linhas normalmente de coloração marrom que se intersectam formando um padrão nas pintas (KITTLER et al., 2016; RUSSO et al., 2017). Como representa a Figura 11, é considerado redes típicas aquelas que representam padrões bem definidos, com tamanho e coloração similares.

Figura 11 – Exemplo de tipicidade da rede de pigmentos a) típico e b) atípico



a) b)
Fonte: ([DERMNETNZ, 2008](#)).

A área amorfá é definida como a região de pigmentação difusa sem nenhuma estrutura específica na pinta (MASSI; LEBOIT, 2014). Esta região pode apresentar uma área difusa regular ou irregular, ilustradas nas Figuras 12 e 13.

Figura 12 – Exemplo de áreas amorfas regulares



Fonte: ([DERMNETNZ, 2008](#)).

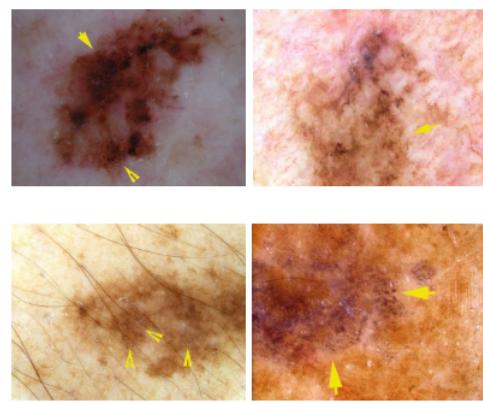
Figura 13 – Exemplo de áreas amorfas irregulares



Fonte: ([DERMNETNZ, 2008](#)).

Adicionalmente, outras três estruturas podem indicar a presença de um tumor maligno, sendo elas: pontos, glóbulos ou estrias. As imagens presentes nas Figuras 14 e 15 mostram a presença destas estruturas em lesões.

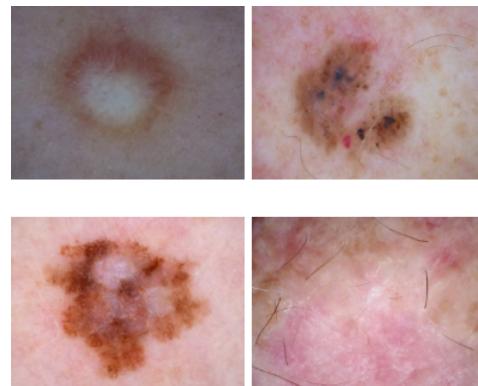
Figura 14 – Exemplo de pontos e glóbulos em melanomas



Fonte: ([DERMNETNZ, 2008](#)).

Segundo Russo et al. (2017), o câncer tipo melanoma normalmente apresenta mais estruturas dermatoscópicas do que nevos comuns. Estatisticamente 73% dos melanomas apresentam 4 ou mais estruturas como estas estudadas nesta Seção. Por isso a detecção destas estruturas se tornam algo crucial na predição da doença.

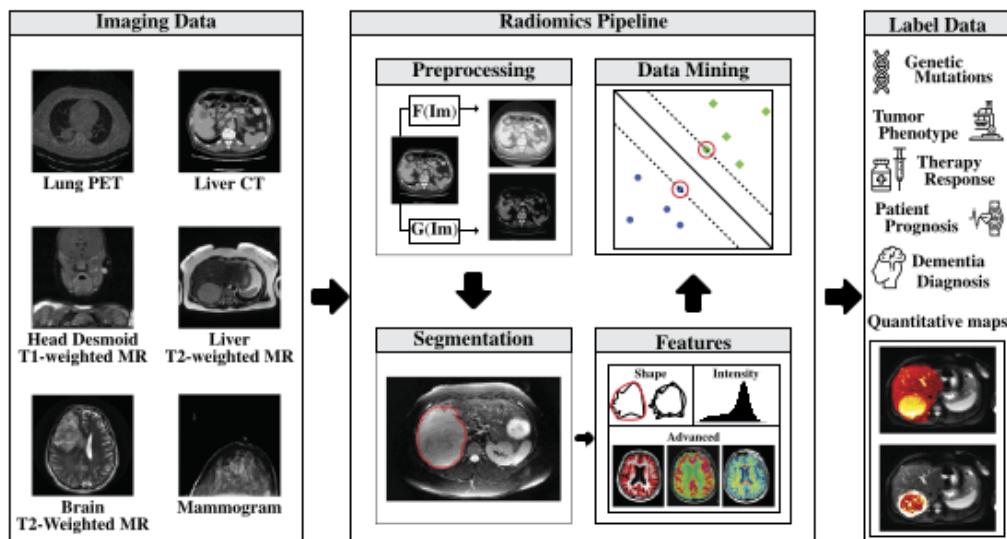
Figura 15 – Exemplo de estrias em melanomas e carcinomas

Fonte: ([DERMNETNZ, 2008](#)).

2.2 Radiomics

A análise quantitativa de atributos em imagens médicas vem sendo estudada durante várias décadas. Em 2012, esta área ganhou o nome de *Radiomics* ([ZHOU S. K.AND RUECKERT; FICHTINGER, 2020](#)). Tal metodologia ou conjunto de técnicas utiliza de imagens digitais para extrair atributos ou informações importantes a serem aplicadas em algoritmos de aprendizado de máquina.

Segundo [Zhou S. K.and Rueckert e Fichtinger \(2020\)](#), embora vários trabalhos apresentem abordagens parecidas, esta área ainda não apresenta uma metodologia bem definida. Porém, é possível ter uma noção de como é a abordagem geral desta técnica através da Figura 16.

Figura 16 – Resumo da metodologia *radiomics*Fonte: ([ZHOU S. K.AND RUECKERT; FICHTINGER, 2020](#)).

As imagens médicas são coletadas por um técnico ou especialista. Em seguida passam por um processo de pré-processamento, onde filtros ou transformações são aplicadas para melhorar a qualidade das mesmas. Através de um modelo computacional, uma região da imagem é segmentada e os atributos pertinentes são extraídos. Por fim, estes dados são introduzidos em modelos de ML para extração de informação sobre a doença.

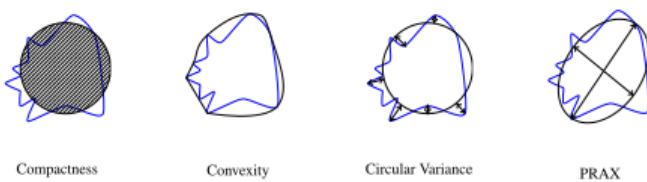
Em *Radiomics*, um atributo é qualquer tipo de informação que seja relevante para a solução de um problema médico. Sendo estes divididos em duas categorias: atributos computacionais ou semânticos ([ZHOU S. K.AND RUECKERT; FICHTINGER, 2020](#)).

Os atributos computacionais são valores numéricos ou categóricos que são extraídos através de visão computacional de uma imagem médica. Já os semânticos consistem em anotações feitas manualmente por um médico ou responsável. Conforme ([ZHOU S. K.AND RUECKERT; FICHTINGER, 2020](#)), este segundo grupo contém fatores clínicos que são pertinentes somente a aquela doença estudada e a combinação desses dois tipos de atributos podem gerar um impacto ainda maior na predição de doenças.

Dentro do grupo de atributos computacionais, estes podem ser subdivididos em comuns e incomuns. Sendo esta primeira categoria designada a características morfológicas, histograma da imagem, padrões cor e presença estruturas como listras ou pontos ([ZHOU S. K.AND RUECKERT; FICHTINGER, 2020](#)).

A análise morfológica, ilustrada pela Figura 17, é o estudo da formato do objeto, quanto ao tamanho, bordas, área e convexidade. Quanto a características relacionadas a coloração, estas são divididas em atributos de primeira ordem para análise em escala cinza de imagem e de ordem superior para identificação de padrões específicos como linhas, estrias ou padrões de cores.

Figura 17 – Exemplos de atributos morfológicos



Fonte: ([ZHOU S. K.AND RUECKERT; FICHTINGER, 2020](#)).

A Seção 3.2 detalha os conjuntos de dados PH2 ([MENDONÇA et al., 2013](#)) e HAM100000 ([TSCHANDL, 2018](#)), bem como os atributos semânticos contidos nestes. Já a Seção 3.3 detalha os procedimentos para obtenção de valores quantitativos das imagens.

2.3 Visão Computacional

Visão computacional, ou *computer vision* do inglês, é uma área da engenharia que visa estudar como sistemas e máquinas são capazes de compreender e interpretar o mundo real através de imagens ou vídeos (BURGER; BURGE, 2016). Devido a esse entendimento, visão computacional é utilizada atualmente em uma variedade de aplicações reais, tais como: reconhecimento de caracteres manuscritos, inspeção de máquinas, reconhecimento de imagens, biometria, análise de imagens médicas e outros (SZEJLIK, 2011).

2.3.1 Imagens Digitais

Uma imagem digital é a discretização de uma função bidimensional $I(x, y)$, onde x e y são as coordenadas espaciais e o valor de I em algum ponto (x, y) corresponde ao brilho ou níveis de cinza da imagem naquele ponto, sendo este valor denominado de elemento da imagem ou pixel (abreviação de *picture elements*) (BOTELHO, 2007). Para uma imagem colorida tipo RGB (*Red, Green, Blue*), a função I se torna a intensidade em cada canal de cor, dado por (BARROS; MORAIS; FERNANDES, 2017; BOTELHO, 2007):

$$I(x, y) = (I_R(x, y), I_G(x, y), I_B(x, y)) \quad (2.1)$$

onde $I_i(x, y), i \in R, G, B$ representa a intensidade do pixel no canal específico naquele ponto (x, y) . Se a intensidade for nula, expressa uma cor preta, se for máxima, branca.

Figura 18 – Posicionamento dos eixos em imagens digitais



Fonte: (BOTELHO, 2007).

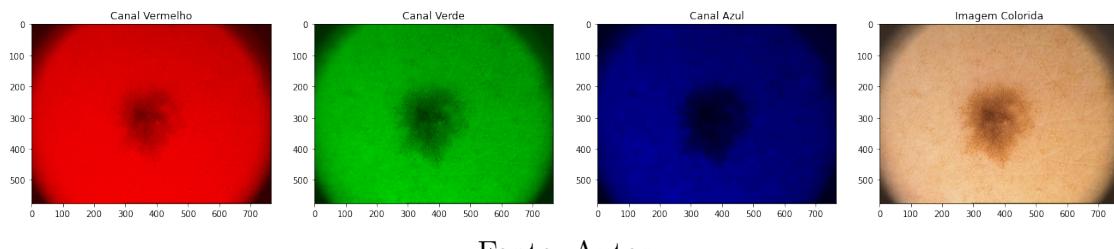
Segundo (BOTELHO, 2007; SZEJLIK, 2011), o posicionamento dos eixos x e y em uma imagem é dado da esquerda para direita (eixo x) e de cima para baixo (eixo y) como apresentada na Figura 18.

Estendendo esse conceito de discretização para toda imagem, combinando com o sistema de posicionamento adotado, uma imagem em escala cinza então pode ser representada como uma matriz (BOTELHO, 2007; BARROS; MORAIS; FERNANDES, 2017):

$$C_{cinza} = \begin{bmatrix} I(0,0) & I(0,1) & I(0,2) & \cdots & I(0,M) \\ I(1,0) & I(1,1) & I(1,2) & \cdots & I(1,M) \\ I(2,0) & I(2,1) & I(2,2) & \cdots & I(2,M) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I(N,0) & I(N,1) & I(N,2) & \cdots & I(N,M) \end{bmatrix} \quad (2.2)$$

onde C_{cinza} é a matriz que representa uma imagem em escala cinza, N e M são as dimensões de comprimento e altura da imagem (em pixels) sendo que $N, M \in \mathbb{N}$. Contudo, vale ressaltar que em uma imagem colorida cada canal de cor apresentará uma matriz (C_R, C_G, C_B). A Figura 19 ilustra essa decomposição da imagem na matriz de cada canal, sendo que a combinação destas que forma a imagem final.

Figura 19 – Representação de uma imagem em escala RGB como canais independentes



Fonte: Autor.

O número de matrizes e os valores de seus elementos dependem da escala de cor utilizada e a qualidade das imagens. Segundo Burger e Burge (2016), cada escala de cor apresenta uma variação e grandezas específicas. Sendo o valor do *bit* um valor inteiro, usualmente, ou decimal.

Na Seção 2.3.2 é realizado um breve estudo para compreender as relações entre a intensidade do bit, a escalas de cor e transformações entre os diversos sistemas

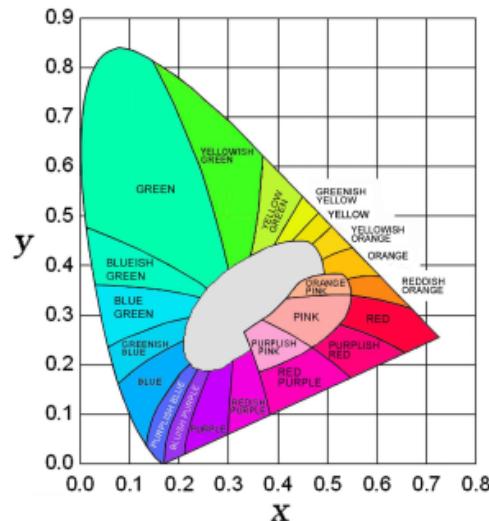
2.3.2 Sistemas de Representação de Cores

A caracterização de imagens digitais através das cores é amplamente utilizada devido ao fato destas serem invariantes quanto ao tamanho, resolução e orientação da imagem. Outro fator que contribuiu para isto é a facilidade de extração desta informação em imagens digitais (BOTELHO, 2007).

Cada sistema de cor apresenta uma abordagem diferente para compor as cores presentes na natureza. Dentre elas, a mais popular é RGB, as cores primárias, compostas por vermelho (*Red*), verde (*Green*) e azul (*Blue*) do espectro visível (BOTELHO, 2007).

Segundo (MALACARA, 2011), na tentativa de definir todas as cores através de radiações primárias, a Commission International de l'Éclairage (CIE) em 1943 criou o sistema de representação de cores XYZ, apresentado no diagrama da Figura 20, no qual todas as cores podem ser geradas através da combinação das três primárias.

Figura 20 – Diagrama do Sistema CIE XYZ



Fonte: (MALACARA, 2011)

Apesar de ser um sistema muito útil na determinação de cores, este sistema não é muito utilizado em programação ([MALACARA, 2011](#)).

Existem diversos outros sistemas de representação de cores atualmente, sendo utilizados neste trabalho os RGB, HSL, HSV, CIE L*a*b e CIE Lu'v'.

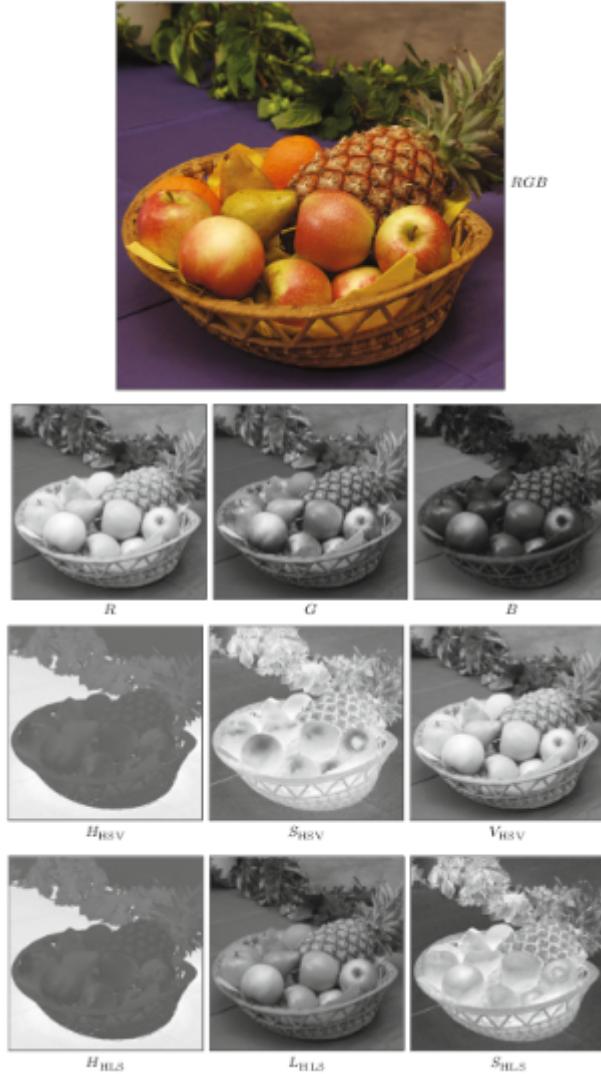
Aplicar estas outras representações se torna um artifício importante na extração de características das imagens. Observando a Figura 21 é visível a importância destas representações diferentes e seus canais para captar informações mais pertinentes ao problema.

O sistema RGB de representação de cores utiliza a combinação das três cores primárias: vermelho (R), verde (G) e azul (B) sendo que R, G e B são as coordenadas cartesianas responsáveis pela formação do cubo unitário (BOTELHO, 2007; BURGER; BURGE, 2016). A Figura 22 demonstra esse cubo unitário e a formação de algumas cores.

Segundo (BURGER; BURGE, 2016), o sistema RGB é aditivo, isto é, todas as cores se iniciam com preto e são criadas ao adicionar as cores primárias, sendo que a junção de todas resultará na cor branca.

Do ponto de vista matemático, (BOTELHO, 2007) diz que uma cor C de um pixel pode ser definida por $C = r.R + g.G + b.B$, onde R,G e B são as cores primárias e r,g e b são os coeficientes correspondentes as intensidades associadas a cada uma das cores, como

Figura 21 – Diferentes percepções da mesma imagem ao analisar canais de cores diferentes



Fonte: ([BURGER; BURGE, 2016](#))

mostra a Figura 23.

Supondo então uma cor $C = (R, G, B)$, a representação desta em escala cinza ou seu coeficiente de luminescência Y é dado por uma média ponderada dos canais RGB ([BURGER; BURGE, 2016; OPENCV, 2020](#)):

$$Y = 0.299R + 0.587G + 0.114B \quad (2.3)$$

onde R, G, B são os valores de cada canal e Y é a intensidade na escala cinza, também chamada de luminescência.

Atualmente, as imagens digitais são salvas na escala RGB, sendo assim, os seus valores podem ser obtidos diretamente do arquivo salvo pela câmera. Então, é necessário utilizar transformações, similares a Equação 2.3, para converter de uma escala para outra.

Figura 22 – Representação do sistema RGB como cubo unitário 3D

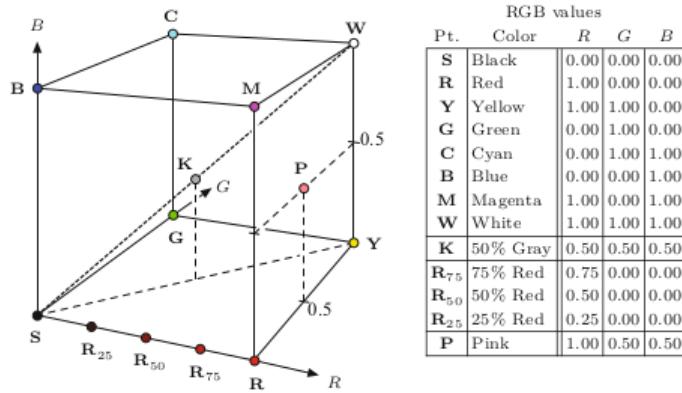
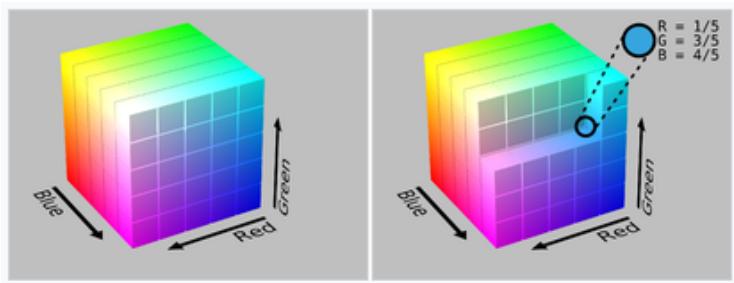
Fonte: ([BURGER; BURGE, 2016](#))

Figura 23 – Sistema RGB como cubo unitário e definição de uma cor a partir da combinação do sistema RGB como cubo unitário 3D

Fonte: ([BURGER; BURGE, 2016](#))

Como foco desta monografia é avaliar o desempenho de modelos de ML na predição de câncer de pele, a explicação de todas as representações não é necessária. As transformações entre as escalas RGB, HSL, HSV, CIE L*a*b e CIE Lu'v' foram implementadas utilizando a biblioteca [OPENCV \(2020\)](#). Nas literaturas [Burger e Burge \(2016\)](#), ([MALLACARA, 2011](#)) e [OPENCV \(2020\)](#) estão presentes seções pertinentes a cada uma das representações utilizadas.

2.4 Aprendizado de Máquina

2.4.1 Uma visão geral

A partir da década de 1970 houve um aumento da complexidade dos problemas vigentes e geração de dados em diversos setores. Tal fato originou uma necessidade de criação de novas ferramentas computacionais que facilitassem o entendimento destes problemas, isto é, o aprendizado de máquina ([FACELI et al., 2011](#)).

O aprendizado de máquina é definido como o processo de indução de uma hipótese

(ou aproximação de uma função) através da experiência (FACELI et al., 2011). Já Géron (2019) define ML como a ciência que permite o aprendizado de computadores através de dados. Em resumo, este aprendizado se baseia em executar um programa para ajustar parâmetros matemáticos de um modelo a partir dos dados existentes.

Existem diversas formas ou abordagens para que o computador aprenda com estes dados, sendo estes divididos em cinco categorias (FACELI et al., 2011; GÉRON, 2019):

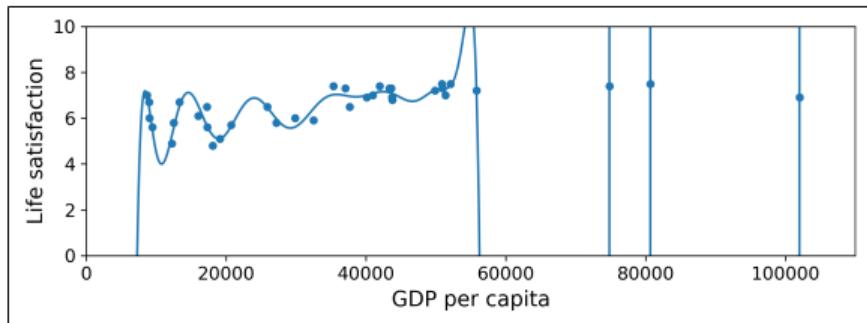
- **Supervisionado:** É considerado aprendizado supervisionado quando um modelo de ML pretende designar um rótulo ou valor a uma instância, onde é utilizado dados já rotulados para treinamento do modelo.
- **Não - Supervisionado:** Este outro método de aprendizado de máquina é utilizado quando deseja-se agrupar os dados em rótulos ou classes, porém estes não apresentam tal informação previamente e cabe ao algoritmo, através do modelo, associar tais características e impor a ele uma rótulo.
- **Semi - supervisionado:** É uma mistura dos métodos Supervisionado e Não - Supervisionado, onde parte dos dados da amostra estão rotulados e parte não, cabendo ao algoritmo rotular o restante dos dados.
- **Por reforço:** Neste tipo de aprendizado, o algoritmo deve aprender por si só a sequência de ações que deve tomar, reforçando ações positivas e punindo as negativas.
- **Evolutivo:** Este método de aprendizado de máquina, segue a teoria de evolução de Darwin, onde mudanças aleatórias são geradas na forma de processamento dos dados, até que se é criado um algoritmo que minimiza o erro da saída.

De maneira simplificada, pode-se dizer que ML cria um modelo matemático e ajusta seus parâmetros de acordo com os dados de entrada. O resultado deste modelo é predizer um valor numérico ou classificar algum dado em alguma categoria. Segundo Géron (2019), existem diversos desafios relacionados a estes dados sendo eles: treinamento com dados de baixa qualidade ou insuficientes, situações como sobreajuste (*overfitting*) e subajuste (*underfitting*).

No caso de sobreajuste, o modelo matemático memoriza os dados de treinamento e cria uma função ajustada que passa por praticamente quase todos os pontos do conjunto, como mostra a Figura 24 (GÉRON, 2019). Já o subajuste é exatamente ao contrário, o modelo utilizado é muito simples para descrever aqueles dados (GÉRON, 2019).

Segundo Géron (2019), antes de aplicar qualquer modelo de ML é necessário entender o problema, realizar uma análise de dados e aplicar um pré-processamento destes dados. Sendo esta primeira etapa já estudada na Seção 2.1.

Figura 24 – Exemplo de sobreajuste



Fonte: ([GÉRON, 2019](#))

A análise de atributos presentes no conjunto de dados permite a descoberta de valiosas informações que ajudam a entender o problema como um todo ([FACELI et al., 2011](#)). Estas características são definidas em duas categorias distintas: atributos qualitativos e quantitativos. O primeiro destes representam categorias ou qualidades como: pequeno, grande, alto e baixo. O segundo tipo são valores numéricos discretos ou contínuos como: 1, 100, 2.5, 0.3.

A etapa de pré-processamento é resumida como a aplicação de tratamentos nos dados para adequá-los aos modelos que serão submetidos. Segundo [Faceli et al. \(2011\)](#), é importante realizar uma transformação neste conjunto pois certos modelos de ML, como o *Support Vector Machine*, sofrem influência direta no formato, condições ou escala dos dados. Estas técnicas são aplicadas para minimizar os problemas ou inconsistências presentes em cada conjunto, gerando dados menos complexos e que facilitam o modelo a aprender os padrões presentes nele.

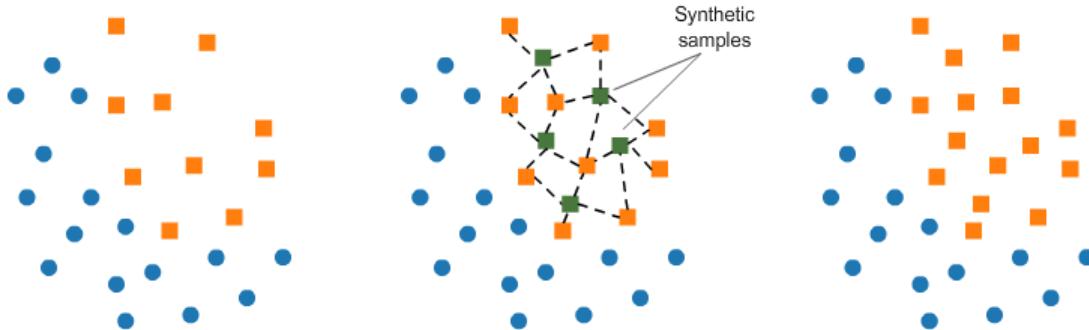
Em problemas de classificação, que é o foco desta monografia, é desejado avaliar um conjunto de dados e atribuí-lo uma classe. Entretanto, em problemas reais, muitas vezes estes conjuntos estão com classes desbalanceadas, o que pode atrapalhar o aprendizado do modelo. Este tipo de problema é bem comum dentro da análise de câncer de pele, ainda mais em Melanomas, uma vez que a ocorrência desta doença é muito menor do que a presença de sinais ou pintas pelo corpo.

Existem inúmeras técnicas de pré-processamento de dados. Neste trabalho foram utilizadas duas técnicas, sendo elas: SMOTE (*Synthetic Minority Oversampling Technique*) e Padronização dos dados.

A técnica SMOTE consiste em criar amostras sintéticas da classe minoritária na etapa de treinamento e utilizá-las no intuito de facilitar o aprendizado daquela classe no modelo de ML ([IBM, 2020](#)). A Figura 25 demonstra o procedimento desta técnica.

Já o procedimento de Padronização é utilizado principalmente para aprimorar alguns modelos de ML que são sensíveis a grandeza da variável ([SCIKIT-LEARN, 2020i](#)).

Figura 25 – SMOTE - Synthetic Minority Oversampling Technique



Fonte: ([IBM, 2020](#))

Supondo uma determinada característica x , como por exemplo o raio da pinta, de um conjunto de dados, seu valor após a padronização será ([SCIKIT-LEARN, 2020i](#)):

$$z = (x - u)/s \quad (2.4)$$

onde z é o valor padronizado, u e s a média e desvio padrão do conjunto de características respectivamente.

Na execução deste trabalho, após extrair as características conforme mencionado na Seção 3.3, os dados foram explorados e exibidos na Seção 4.1. Ademais, como os modelos sofrem influência no formato e distribuição dos dados, esta monografia teve como intuito avaliar o desempenho destes modelos na presença e ausência de pré-processamento. Para cada caso foi analisado o desempenho do modelo.

Foram utilizados modelos de classificação com aprendizado supervisionado introduzindo as características quantitativas extraídas de imagens e formulários médicos do conjunto de dados para classificá-los em câncer. Tais modelos buscam desenvolver uma função estimadora para predizer uma classe com base em um domínio de dados já rotulados ([FACELI et al., 2011](#)).

Para entender melhor os conceitos de ML abordados até o momento e como cada algoritmo funciona, a matemática e explicação desta se baseiam nos argumentos de [Faceli et al. \(2011\)](#) e [Géron \(2019\)](#).

Para um conjunto de observações de pares $\mathbf{D} = (\mathbf{x}_i, f(\mathbf{x}_i)), i = 1, \dots, n$, em que f simboliza uma função desconhecida e \mathbf{x}_i um conjunto de características. Um modelo de ML aprende uma aproximação \hat{f} para f utilizada para estimar valores de para novos dados de \mathbf{x} . Essa função aproximadora associada a classificação assume então valores em um conjunto discreto descrito por ([FACELI et al., 2011](#)):

$$y_i = \hat{f}(\mathbf{x}_i) \in (C_1, C_2, \dots, C_n) \quad (2.5)$$

onde y_i representa a classe estimada para aquele conjunto de dados \mathbf{x}_i e (C_1, C_2, \dots, C_n) são as possíveis classes.

Segundo [Faceli et al. \(2011\)](#), a função \hat{f} pode assumir diversas formas como combinações lineares ou não lineares dos atributos, expressões logísticas e etc. Também é importante ressaltar que \hat{f} é uma aproximação e que os modelos utilizam uma função de custo que tende a minimizar o erro daquela predição. No caso de problemas de classificação, é usual utilizar a função de custo 0 - 1: custo de uma previsão incorreta ($\hat{f}(\mathbf{x}) \neq f(\mathbf{x})$) é 1 e de um previsão correta ($\hat{f}(\mathbf{x}) = f(\mathbf{x})$) é 0 ([FACELI et al., 2011](#)).

Em problemas de classificação existem diversas formas de avaliar o desempenho do estimador como acurácia, precisão, sensibilidade e outros. Na Seção [2.4.3](#) é estudado estas métricas de desempenho em problemas de duas ou mais classes e a combinação de métricas utilizadas nesta monografia.

A melhor maneira de saber se o modelo consegue generalizar o problema é através do processo de treinamento e teste. Segundo [Géron \(2019\)](#), uma ótima opção para avaliar o modelo é realizar a predição de novos dados, separando o conjunto de dados total em conjunto de treinamento e conjunto de teste. [Géron \(2019\)](#) e [Faceli et al. \(2011\)](#) também indicam utilizar 80% dos dados para treinar o modelo e o restante para testá-lo.

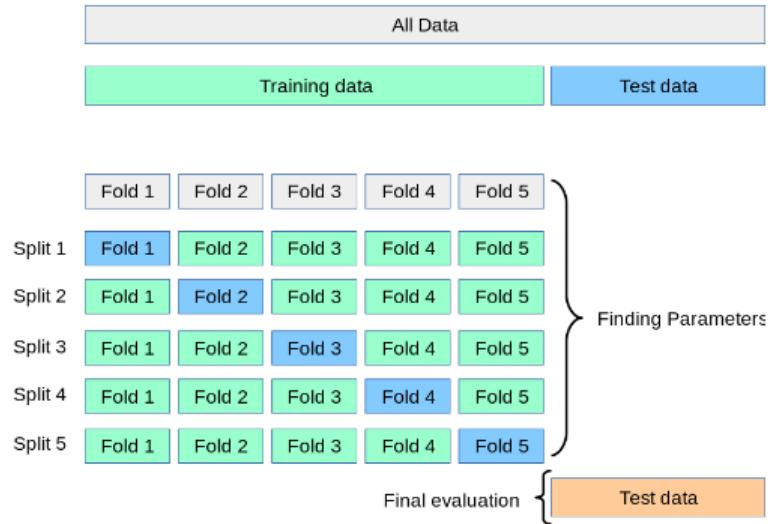
Durante o processo de aprendizagem do modelo é possível que ocorra um *overfitting* e *underfitting* dos dados. Segundo [Faceli et al. \(2011\)](#), para evitar este tipo de problema, deve-se utilizar de métodos de amostragem para obter estimadores mais confiáveis. Uma forma de solucionar esta problemática é alterar os hiperparâmetros dos modelos de ML.

Cada modelo apresenta uma formulação matemática própria, sendo o processo de controle destes hiperparâmetros denominado de regularização ([GÉRON, 2019](#); [SCIKIT-LEARN, 2020k](#)). Utilizar o próprio conjunto de testes para encontrar os melhores parâmetros entretanto é um erro que pode gerar sobreajuste e para evitar isso utiliza-se normalmente a metodologia de validação cruzada ([SCIKIT-LEARN, 2020a](#)).

Este processo busca criar um conjunto de validação nos dados de treinamento, de modo a deixar o conjunto de teste somente para avaliar o desempenho final do modelo treinado ([FACELI et al., 2011](#)). O processo de validação cruzada utilizada nesta monografia se chama *K-Fold Cross Validation*.

O conjunto de dados é dividido em K sub-porções de tamanhos iguais e cada porção é utilizada no processo de treinamento aplicando uma combinação de hiperparâmetro pré-estabelecidos em cada ciclo de funcionamento ([SCIKIT-LEARN, 2020a](#); [FACELI et al., 2011](#)). A Figura [26](#) mostra esta técnica.

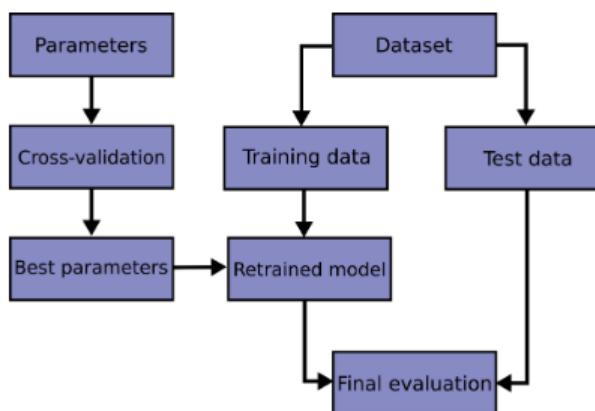
Em conjunto com a validação cruzada, também é utilizado o *grid search*. Ele realiza uma combinação de todos os hiperparâmetros sujeitados a treinamento e em conjunto com a validação cruzada encontra a combinação que apresentou melhores resultados

Figura 26 – *K-Fold Cross Validation*

Fonte: ([SCIKIT-LEARN, 2020a](#))

([SCIKIT-LEARN, 2020k](#)).

Por fim, a Figura 27 mostra um procedimento padrão de como realizar os processos de validação cruzada e a busca de melhores hiperparâmetros de um modelo.

Figura 27 – Procedimento usado para treinamento e avaliação dos modelos de ML utilizando a Validação Cruzada e *Grid Search*

Fonte: ([SCIKIT-LEARN, 2020a](#))

No Capítulo 3 será melhor explicado o desenvolvimento computacional e como foi utilizado estas técnicas neste trabalho.

Ademais, os modelos usados nesta monografia e seus estudos teóricos serão trabalhados nas subseções posteriores. Todos os modelos nesta monografia foram treinados sobre a perspectiva do aprendizado supervisionado e submetidos a testes com diferentes tipos de conjuntos e pré-processamento de dados.

2.4.2 Modelos

Os dados obtidos através da extração de informações das imagens dos conjuntos PH2 (MENDONÇA et al., 2013) e HAM10000 (TSCHANDL, 2018) são representados por um vetor de atributos \mathbf{x}_i . Para os algoritmos trabalhados, é fornecido estes conjuntos e o rótulo associado a ele $f(\mathbf{x}_i)$. Por exemplo, se a imagem for câncer de pele, a função f apresentará $f(\mathbf{x}_i) = 1$ e $f(\mathbf{x}_i) = 0$ caso contrário contrário. Para situações de análise multiclassificadas, valores diferentes de $f(\mathbf{x}_i)$ são atribuídos.

2.4.2.1 Regressão Logística

O modelo de Regressão Logística, também conhecido como *Logit Regression* é um modelo linear comumente utilizado em classificação. Este modelo é utilizado para estimar a probabilidade de um conjunto de atributos pertencentes a uma classe em particular (GÉRON, 2019).

De maneira simples, modelos lineares são combinações lineares das características da base de dados. Este por sua vez é definido como (SCIKIT-LEARN, 2020d):

$$\hat{f}(\mathbf{w}, \mathbf{x}) = w_0 + w_1x_1 + \dots + w_px_p \quad (2.6)$$

no qual \hat{f} é a função aproximadora, p o número de características, $w = (w_1, \dots, w_p)$ o vetor com os coeficientes (ou pesos dos atributos) e w_0 é o termo escalar.

Já na regressão logística o modelo calcula a soma ponderada das características de entrada e soma-os a um termo polarizador (w_0) e essa probabilidade é apresentada em formato de uma função logística. Matematicamente, este modelo é descrito por (GÉRON, 2019):

$$\hat{f} = \sigma(\mathbf{w} \cdot \mathbf{x}) \quad (2.7)$$

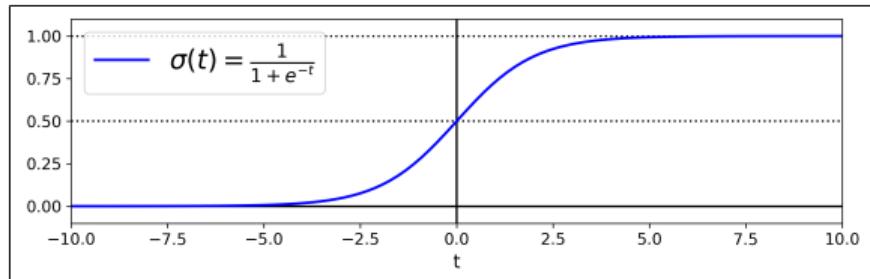
sendo \hat{f} a função aproximadora logística, σ a função logística e $\mathbf{w} \cdot \mathbf{x}$ é o produto escalar entre o vetor de pesos e de características.

O resultado da Equação 2.7 é um valor entre 0 e 1 que expressa a probabilidade de ocorrência da classe. De modo geral, segundo Géron (2019), se a probabilidade for maior que 0.5 o algoritmo classifica aquele dado como classe 1, caso ao contrário, como classe 0. Sendo que a função $\sigma(\cdot)$ também é denominada de função sigmoidal (apresenta um formato de S, como ilustra Figura 28).

Definir porém quais são os coeficientes não é uma tarefa fácil. De fato, Géron (2019) expõem que a função de custo para analisar todo o conjunto de dados pode ser dado por:

$$J(\mathbf{w}) = -\frac{1}{n} \sum_{i=1}^n \left[y^{(i)} \log(\hat{p}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{p}^{(i)}) \right] \quad (2.8)$$

Figura 28 – Função Logística



Fonte: (GÉRON, 2019)

onde J representa a função de custo, \hat{p} a probabilidade associada a classe y . Esta equação também é conhecida como *log loss*. Porém não existe uma equação bem definida que possa minimizar o custo afim de obter um melhor modelo.

Outra forma de calcular estes parâmetros o modelo, de maneira que minimize o erro do modelo, é resolver a seguinte equação (SCIKIT-LEARN, 2020d):

$$\min_{\mathbf{w}, c} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \log(\exp(-y_i(\mathbf{w} \cdot \mathbf{x}_i + c)) + 1) \quad (2.9)$$

onde o parâmetro C é uma variável computacional que equilibra a regularização de \mathbf{w} com a quantidade de desvios permitidos no treinamento.

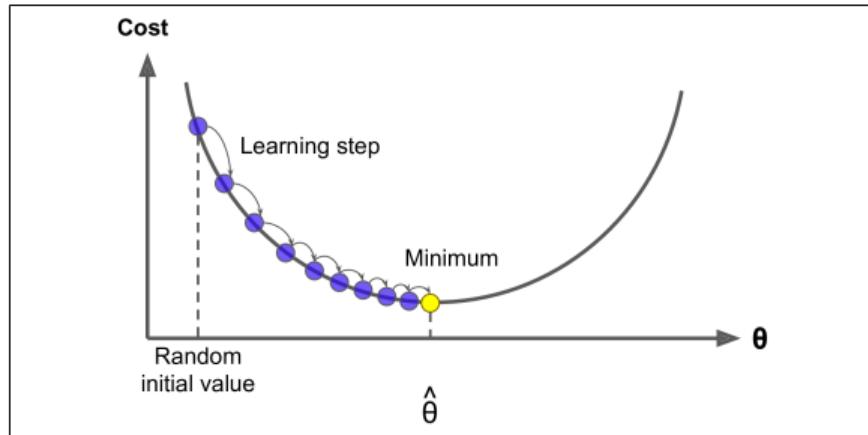
Géron (2019) introduz a função de custo como a apresentada na Equação 2.8. Desta forma, o ajuste do modelo é proveniente da minimização desta função através de outro método que será explicado a seguir. Por outro lado, a documentação da biblioteca SCIKIT-LEARN (2020d) apresenta a função de custo como a apresentada pela Equação 2.9. Esta é mais fácil de ser resolvida computacionalmente, porém ainda é necessário utilizar um método de otimização para encontrar os melhores valores de \mathbf{w} .

Um método utilizado para estes problemas de otimização é o *Gradient Descent*, capaz de encontrar a solução para uma variedade de problemas matemáticos. A ideia geral é supor valores que minimizam a função de custo e ir alterando-o iterativamente (GÉRON, 2019). De maneira prática, é suposto um valor inicial e então o algoritmo iterativamente encontra novos valores que minimizam o custo até que este converja para o mínimo como ilustra a Figura 29.

Um parâmetro muito importante nestes tipos de otimizadores é a taxa de aprendizado e número de iterações. Se for escolhido uma taxa de aprendizado baixo, é necessário muitas iterações para encontrar o mínimo da função. Se for escolhido uma taxa alta, é possível que o algoritmo passe pela valor de mínimo e acabe não convergindo.

Apesar de ser um método simples e que funciona para a maioria das funções de custo, este método apresenta falhas em funções similares a Figura 30. Se tal procedimento

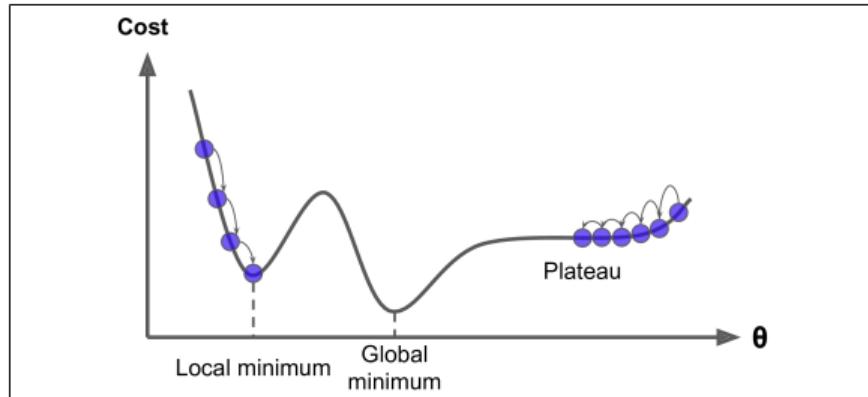
Figura 29 – Gradient Descent



Fonte: (GÉRON, 2019)

iniciar em um ponto a esquerda por exemplo, ele irá convergir para o mínimo local e se iniciar a esquerda, irá demorar para alcançar o mínimo global (GÉRON, 2019).

Figura 30 – Falhas no Gradient Descent



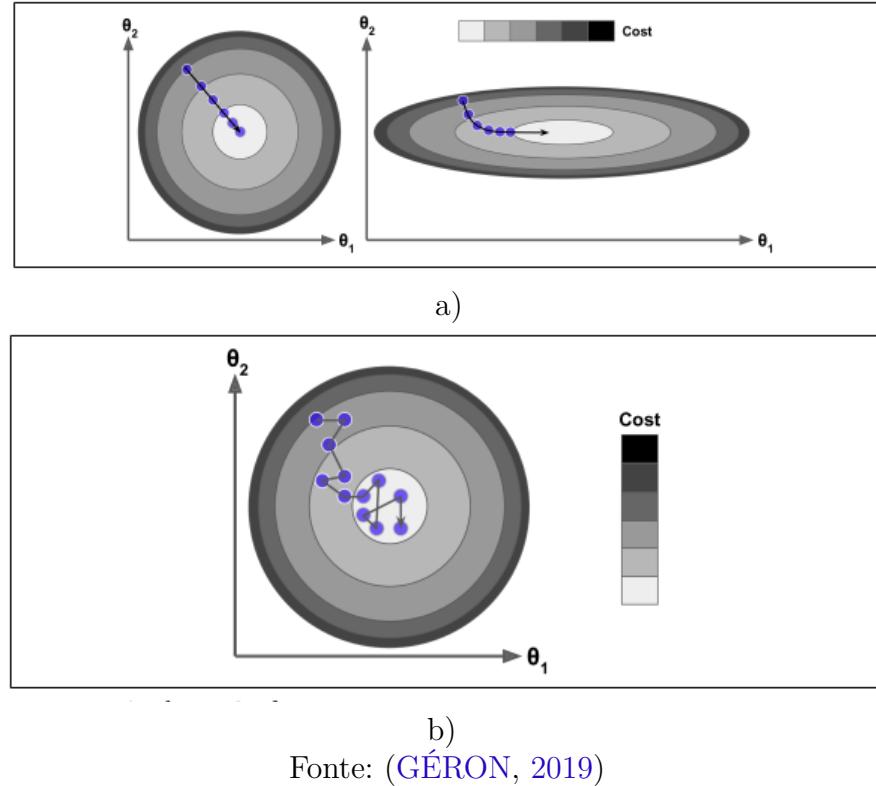
Fonte: (GÉRON, 2019)

Para aplicar este método no conjunto de dados e modelo utilizado, é possível utilizar diversas abordagens, valendo a pena ressaltar neste trabalho duas: *Batch Gradient Descent* (BGD) e *Stochastic Gradient Descent* (SGD).

Neste primeiro, para cada iteração é calculado a variação da função de custo de todos os dados presente no conjunto. Por ser um método que utiliza todos os valores para encontrar o parâmetro que minimiza a função de custo, ele acaba tendo um custo computacional elevado (GÉRON, 2019). Já o método SGD, o algoritmo utiliza um conjunto de pontos aleatórios no conjunto de dados em cada interação (GÉRON, 2019). Segundo Géron (2019), a vantagem deste método é que por utilizar menos informação em cada iteração, ele apresenta maior velocidade de processamento comparado ao BGD.

Já a desvantagem, ainda segundo Géron (2019), por ser um processo estocástico, mesmo quando este chegar no mínimo local, ele irá gerar novos passos encontrando outros valores que podem ser melhores ou piores que o anterior. Portanto, quando o algoritmo finaliza sua execução o parâmetro encontrado é bom para o problema, porém não o melhor possível. A Figura 31 mostra esta diferença na curva de aprendizado de ambos os métodos citados.

Figura 31 – Diferença na minimização do a) Batch e b) Stochastic Gradient Descent



Fonte: (GÉRON, 2019)

É valido ressaltar que no intuito de aproveitar o melhor de cada modelo, os hiperparâmetros C da Equação 2.9 e o método de otimização para encontrar os valores foram determinados pelo processo de validação cruzada. A combinação que apresenta as maiores taxas de acerto é escolhida para ser incorporada ao modelo.

2.4.2.2 Gaussian e Bernoulli Naive Bayes

O modelo de *Naive Bayes* (NB) é um modelo probabilístico baseado nos teoremas de Bayes (FACELI et al., 2011). Segundo Faceli et al. (2011), esse método se baseia na probabilidade de um evento A (a classe associada ao conjunto) ocorrer dado um evento B (os atributos) tomando a suposição de que estes eventos não estão correlacionados.

Aplicando o teorema de Bayes, pode-se definir um método para calcular a probabi-

lidade de uma classe dada por (FACELI et al., 2011; SCIKIT-LEARN, 2020f):

$$P(y_i | \mathbf{x}) = \frac{P(y_i)P(\mathbf{x} | y_i)}{P(\mathbf{x})} \quad (2.10)$$

sendo $P(y_i | \mathbf{x})$ a probabilidade de um conjunto de atributos \mathbf{x} pertencer a um classe y_i , $P(y_i)$ a probabilidade *a priori* da classe, $P(\mathbf{x} | y_i)$ a probabilidade de observar vários objetos que pertencem a classe y_i e $P(\mathbf{x})$ a probabilidade de ocorrência dos atributos.

Utilizando a suposição de que os atributos não apresentam interdependência, isto é (SCIKIT-LEARN, 2020f):

$$P(x_i | y_i, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i | y_i) \quad (2.11)$$

Utilizando a Equação 2.11 na 2.10, têm-se que (FACELI et al., 2011; SCIKIT-LEARN, 2020f):

$$P(y_i | \mathbf{x}) = \frac{P(y_i)\prod_{i=1}^n P(x_i | y_i)}{P(\mathbf{x})} \quad (2.12)$$

Como o termo $P(\mathbf{x})$ é constante dado o conjunto de dados, a probabilidade de um exemplo pertencer a classe y_i é proporcional a expressão (SCIKIT-LEARN, 2020f; FACELI et al., 2011):

$$P(y_i | \mathbf{x}) \propto P(y_i) \prod_{i=1}^n P(x_i | y_i) \quad (2.13)$$

A Equação 2.13 é denominada de *Classificador Naive Bayes* (FACELI et al., 2011). Segundo Faceli et al. (2011), os classificadores tipo NB são derivações da Equação 2.13, sendo que em cada uma destas, uma suposição acerca da distribuição de probabilidades é feita, resultando em um novo método.

Um exemplo de método é o *Maximum A Posteriori* (MAP), o qual assume que as probabilidades *a priori* das hipóteses de y_i são iguais. Então pode-se usar o termo $P(x_i | y)$ para encontrar a hipótese mais provável, onde a probabilidade é expressa por (FACELI et al., 2011; SCIKIT-LEARN, 2020f):

$$\hat{y} = \arg \max_y P(y_i) \prod_{i=1}^n P(x_i | y_i) \quad (2.14)$$

Apesar do MAP ser um ótimo método, sua aplicabilidade é limitada devido ao grande número de exemplos necessários para serem calculados, demandando um alto custo computacional. Para superar isso, outros métodos foram criados baseando-se em hipóteses específicas para criar funções discriminantes que possam facilitar o problema (FACELI et al., 2011). Dentro os diversos métodos existentes, este trabalho estudou o uso dos *Naive Bayes Gaussian* (NBG) e *Naive Bayes Bernoulli* (NBB).

No algoritmo NBG, utilizado em classificação, é assumido que os atributos seguem um comportamento Gaussiano e a probabilidade $P(x_i | y_i)$ é calculada pela relação

(SCIKIT-LEARN, 2020f):

$$P(x_i | y_i) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \quad (2.15)$$

sendo μ_y , σ_y a média e desvio padrão do atributo que apresentou maior probabilidade para classe y_i . Na parte computacional deste método, a documentação da biblioteca SCIKIT-LEARN (2020f) apresenta a possibilidade de adicionar um termo no cálculo da Equação 2.15, que representa a maior variação de todos as características de entrada .

É importante mencionar que o único hiperparâmetro para este modelo é o termo aditivo e que foi determinado pelo processo de validação cruzada.

Já no algoritmo de *Bernoulli Naive Bayes*, toma-se como suposto que o conjunto de dados apresenta uma distribuição multivariada de Bernoulli, isto é, mesmo havendo múltiplas características, cada uma assume ser uma variável binária 0 ou 1 (também chamada de booleana) (SCIKIT-LEARN, 2020f). Utilizando esta condição, a probabilidade $P(x_i | y_i)$ é dada pela expressão (SCIKIT-LEARN, 2020f):

$$P(x_i | y_i) = P(i | y_i)x_i + (1 - P(i | y_i))(1 - x_i) \quad (2.16)$$

onde $P(i | y_i)$ representa a probabilidade de ocorrência da característica i aparecer no conjunto de dados.

Este modelo necessita que os dados de entrada estejam no formato binário e caso não estejam, o algoritmo implementa um processo de binarização nos dados de entrada utilizando um valor limiar, que é definido como hiperparâmetro no código. Por fim, é feito a ressalva de que este hiperparâmetro foi encontrado utilizando a validação cruzada.

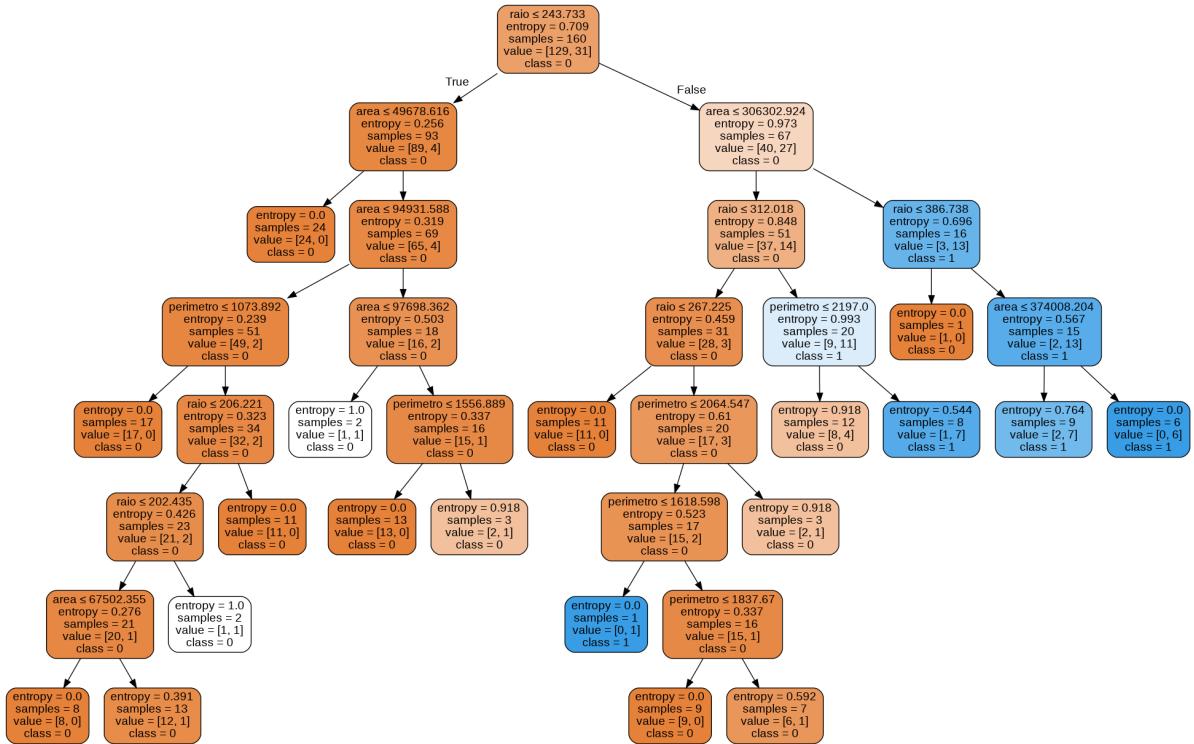
2.4.2.3 Árvore de Decisões

De acordo com Faceli et al. (2011), o modelo de Árvores de Decisões utiliza a estratégia de dividir para conquistar, resolvendo um problema de decisão. Um problema complexo é dividido em vários simples, estes são analisados e divididos novamente até que se obtenha um resultado para o problema geral (GÉRON, 2019).

O funcionamento deste modelo pode ser notado ao avaliar a metodologia da Figura 32. A Figura representa o desenvolvimento de uma árvore de decisão simplificada utilizada nesta monografia. É analisado os atributos de raio, área e perímetro das bordas de lesões do conjunto . Em problemas mais complexos, a árvore de decisão engloba todas as características do conjunto de treino, podendo dividir a mesma característica em mais de um sub grupo para realizar a sua construção.

De acordo com Gatto (2020) a principal vantagem deste método é a visualização do dados e do mecanismo de classificação de maneira altamente funcional.

Figura 32 – Exemplo de árvore de decisão para tarefa de classificação proposta



Fonte: Autor

Para entender melhor o funcionamento deste método, toma-se as seguintes definições dadas por [Faceli et al. \(2011\)](#):

- Um nó folha é rotulado com um função. Sendo que normalmente utiliza-se apenas os valores da variável principal para aplicar esta função com resposta binária (0 ou 1) no conjunto de dados.
- Um nó de divisão contem um teste condicional baseado nos valores do atributo.

Analizando a Figura 32, cada retângulo é chamado de nó folha e as setas que dividem o atributo são nós de divisão. É possível notar que cada nó folha apresenta um valor alvo e um limiar. Faz-se um teste condicional para avaliar se tal atributo está ou não dentro da faixa proposta.

A qualidade do resultado obtido após divisão utiliza a teoria da informação, também conhecida como teoria de ganho. Tal princípio se baseia no fato de que o conjunto de dados estará melhor definido ou puro, depois que ele passa para o próximo nó, isto é, após a divisão. Então, a cada iteração, o algoritmo escolhe a característica que fornece a ele melhor ganho de informação. Esse ganho, é cálculo através da Entropia de Shannon, ou simplesmente Entropia, calculada por ([FACELI et al., 2011](#); [GATTO, 2020](#)):

$$H_i = - \sum_{i=1}^n P(y_i) \log_2(P(y_i)) \quad (2.17)$$

onde n é o número de classes, $P(y_i)$ é a probabilidade de ser a classe y_i e o sinal negativo é para trazer a entropia a um valor positivo.

Durante o processo de aprendizagem, o algoritmo calcula utilizando uma característica A e B a entropia antes de aplicar a divisão e após ela em ambas os casos. A entropia de ganho, definida como $H_{antes} - H_{depois}$, é calculada para avaliar qual característica é mais relevante para divisão dos dados. A que obtiver a maior entropia de ganho é então utilizada para gerar uma nova ramificação.

Outra forma de avaliar a qualidade do resultado é avaliar a pureza daquele nó, isto é, o quanto aquele grupo de amostras consegue definir uma classe (FACELI et al., 2011; SCIKIT-LEARN, 2020b). Essa impureza é calculada através do índice de Gini (GÉRON, 2019):

$$G_i = 1 - \sum_{i=1}^n P(y_i)^2 \quad (2.18)$$

onde $P(y_i)$ é a probabilidade de cada classe.

Para esta função de custo, é calculada a impureza média dos nós. Tal valor é subtraído de G_i e o atributo que obtiver maior diminuição de impureza (G_i próximo de 0) é então selecionado para divisão.

De acordo com Géron (2019), o modelo de árvore de decisões apresenta tendências a sobreajustes devido a regra de divisão que permite a extensão da divisão até chegarem próximas aos dados de treino. Géron (2019) afirma que para contornar este problema é essencial regularizar alguns hiperparâmetros computacionais das árvores de decisão para generalizar o problema corretamente.

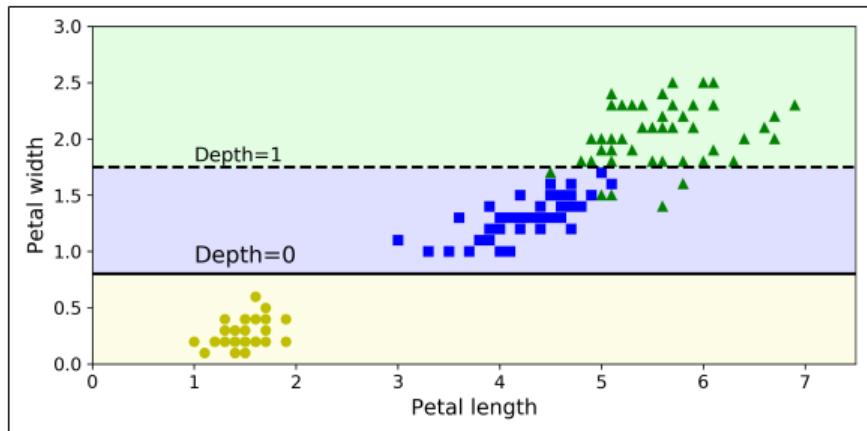
Dos diversos hiperparâmetros possíveis para regularização presentes na documentação do SCIKIT-LEARN (2020b), este trabalho procurou alterar o número mínimo de amostras em cada nó folha, número mínimo de amostras que cada nó deve ter após a separação, função de custo (Entropia de Shannon ou Índice Gini) e a profundidade da árvore de decisão.

Este último hiperparâmetro é de extrema importância para a criação do modelo. De acordo com Géron (2019), a profundidade determina em quantas vezes o conjunto treino poderá ser dividido.

A Figura 33 mostra a classificação de uma árvore de decisões utilizando profundidades diferentes. Utilizando uma profundidade de 0, o conjunto é dividido somente uma vez e não consegue predizer corretamente todas as classes, ao aumentar a profundidade, o modelo já aprende a dividir melhor o conjunto de treinamento. Ressalva-se também que utilizar valores altos de profundidade podem gerar um sobreajuste no modelo de treinamento, o que não é bom para a generalização do problema.

Segundo Géron (2019), vale ressaltar que o modelo de Árvores de Decisões é o

Figura 33 – Classificação da árvore de decisões com diferentes valores de profundidade



Fonte: ([GÉRON, 2019](#))

conceito base para criação do modelo de Florestas Aleatórias e Árvores Extremamente Aleatórias, que serão abordados nas próximas Subseções.

Notifica-se aqui, que os melhores valores para os hiperparâmetros mencionados anteriormente são levantados no processo de validação cruzada. Realizando a combinação entre todos os valores selecionados e mencionados na Metodologia (Seção 3.5)

2.4.2.4 Floresta Aleatória e Árvores Extremamente Aleatórias

O método Florestas Aleatórias faz parte de uma técnica conhecida como *Ensemble Learning*. Esse tipo de técnica se baseia na ideia de que juntando as previsões de diversos métodos, será obtido um resultado melhor do que utilizando somente um estimador ([GÉRON, 2019](#)).

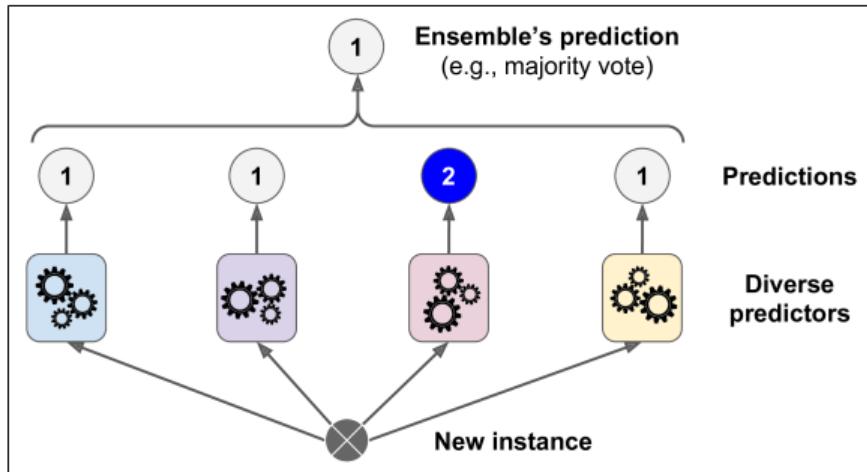
A Figura 34 exemplifica esta técnica, onde é utilizado 4 estimadores para realizar a votação da classe prevista.

É possível treinar um grupo de Árvores de Decisões que utilizam critérios e amostras diferentes, obtendo-se resultados distintos para cada árvore. Quando um novo dado é submetido a teste, é realizado uma votação entre estes modelos treinados e a classe mais votada é então selecionada ([GÉRON, 2019](#)). Esse agrupamento de Árvores de Decisões, segundo Géron (2019), é denominado de Árvores Aleatórias (*Random Forest* do inglês).

Como dito anteriormente, este modelo é totalmente baseado na construção de diversas árvores de decisão e em seu agrupamento. Além disto, estas árvores são criadas usando critérios aleatórios e portanto cada estimador tende a ser único, criando sua própria árvore de decisão.

A alta variância de resultados imposta por esse método tende a gerar um sobreajuste. Todavia, a utilização da aleatoriedade para criação das árvores soluciona este problema.

Figura 34 – Exemplo de um treinado do *Ensemble Learning*



Fonte: ([GÉRON, 2019](#))

Utiliza-se uma média entre os estimadores, o modelo de Floresta Aleatória reduz o erro a um custo computacional maior ([SCIKIT-LEARN, 2020c](#)).

Segundo ([GÉRON, 2019](#)), a Floresta Aleatória apresenta os mesmos hiperparâmetros do que Árvore de Decisão mais o número de estimadores que serão treinados. Então para este modelo, os hiperparâmetros avaliados pelo processo de validação cruzada foram os mesmos das árvores de decisão e a quantidade de árvores de decisão (estimadores) construídas.

O modelo de Árvores Extremamente Aleatórias é outro *Ensemble Learning* que também se baseia nas Árvores de Decisões. Segundo [Géron \(2019\)](#), este método segue o mesmo procedimento da Floresta Aleatória, porém além de escolher uma característica aleatória para dividir um nó das árvores de decisão, o limiar desta característica também é escolhido aleatoriamente.

Este procedimento torna o treinamento das Árvores Extremamente Aleatórias mais rápido do que o das Florestas Aleatórias, uma vez que não é mais necessário realizar o cálculo de limiar para dividir a amostra ([GÉRON, 2019](#); [SCIKIT-LEARN, 2020c](#)).

Ressalva-se ainda que, conforme [Géron \(2019\)](#), utilizar um grande número de estimadores pode causar um sobreajuste. Os hiperparâmetros deste modelo é exatamente igual aos do Floresta Aleatória e estes foram obtidos através do processo de validação cruzada. Assegurando-se de que as quantidades de estimadores fossem adequados para que não ocorra o sobreajuste.

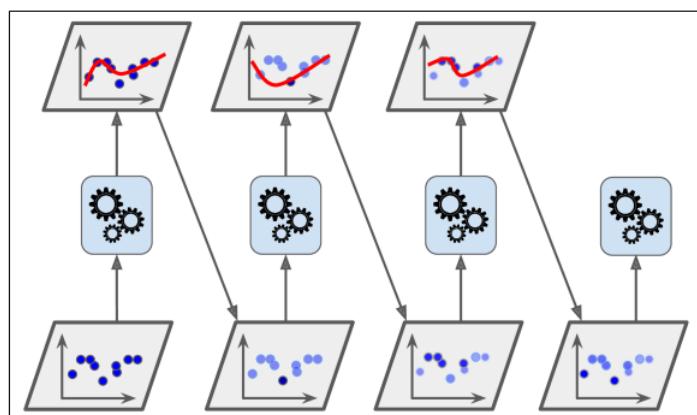
2.4.2.5 *Adaptative Boosting e Gradient Boosting*

O termo *Boosting* se refere a qualquer modelo que utiliza o método *Ensemble Learning* para realizar um combinação sequencial de estimadores para criar um mais potente (GÉRON, 2019). Segundo Géron (2019), a ideia geral deste método é que ao treinar estimadores de maneira sequencial, o modelo seguinte focará em corrigir o erro do modelo anterior. Dos diversos modelos existentes, este trabalho optou por estudar *Adaptative Boosting* (AdaBoost) e *Gradient Boosting*. Ressalva-se ainda que estes modelos podem utilizar diversos estimadores diferentes.

Para esta monografia, os estimadores usados para o modelo Gradient Boosting também são árvores de decisão, denominado por Géron (2019) e SCIKIT-LEARN (2020c) como *Gradient Tree Boosting*.

No modelo AdaBoost, o estimador seguinte utiliza o erro do anterior para criar um novo que seja capaz de se ajustar aos dados que o primeiro não conseguiu. O novo modelo é capaz ainda de classificar casos mais difíceis ou específicos (GÉRON, 2019). A Figura 35 exemplifica esta metodologia.

Figura 35 – Exemplo de um treinamento usando *AdaBoost*

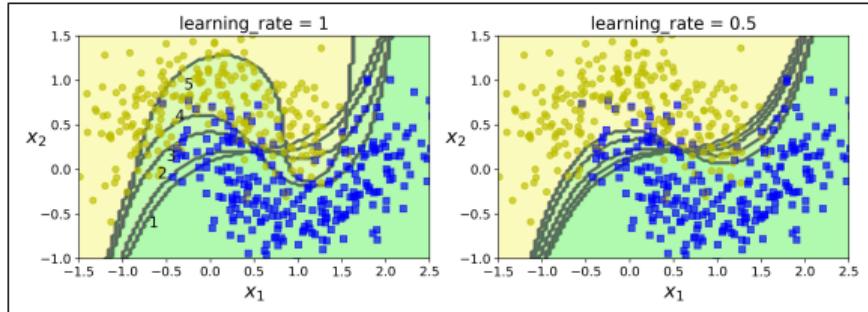


Fonte: (GÉRON, 2019)

Segundo Géron (2019), essa técnica de aprendizado sequencial se assemelha ao *Gradient Descent* estudado na Seção 2.4.2.1. A diferença somente é que ao invés de minimizar a função de custo para encontrar um parâmetro, o modelo calcula a técnica que obterá menor erro e adicionar ao conjunto, melhorando-o gradualmente.

Uma vez que o método utiliza uma técnica de otimização para treinamento, a taxa de aprendizado é um fator crucial no controle de qualidade do estimador e no tempo gasto computacionalmente para treinar o modelo. A Figura 36 exemplifica este argumento e reforça o que foi dito na Seção 2.4.2.1. Variando-se a taxa de aprendizado, pode-se melhorar a performance do modelo ou piorar.

Figura 36 – Diferença de resultados no modelo do *AdaBoost* variando a taxa de aprendizado



Fonte: (GÉRON, 2019)

Em resumo, o modelo AdaBoost combina as previsões de todos os estimadores combinados utilizando pesos em cada um (média ponderada) para inferir o valor final (SCIKIT-LEARN, 2020c). Segundo a documentação do SCIKIT-LEARN (2020c), as modificações de dados em cada iteração consistem na aplicação de pesos w_1, w_2, \dots, w_N para cada uma das amostras de treino. Inicialmente todos os modelos tem o mesmo peso $w_i = 1/N$ e para cada iteração os pesos são recalculados e cada nova amostra recebe um novo peso.

Olhando mais atentamente para o algorítimo do modelo, Géron (2019) define a taxa de erro de cada peso como:

$$r_j = \frac{\sum_{i=1}^n w^{(i)} (\hat{y}_j^{(i)} \neq y^{(i)})}{\sum_{i=1}^n w^{(i)}} \quad (2.19)$$

onde $\hat{y}_j^{(i)}$ é a previsão do j-ésimo estimador para a i-ésima instância.

Utilizando então este cálculo de erro, o modelo calcula um novo peso dado por (GÉRON, 2019):

$$\alpha_j = \eta \log \frac{1 - r_j}{r_j} \quad (2.20)$$

sendo α_j um novo peso previsto e η é a taxa de aprendizado, o hiperparâmetro mencionado nesta Seção.

Em seguida, a nova instância redistribui os pesos conforme a seguinte relação (GÉRON, 2019):

$$w^{(i)} \leftarrow \begin{cases} w^{(i)} & \text{se } \hat{y}_j^{(i)} = y^{(i)} \\ w^{(i)} \exp(\alpha_j) & \text{se } \hat{y}_j^{(i)} \neq y^{(i)} \end{cases} \quad (2.21)$$

Este processo é repetido e novos pesos são calculados até que se alcance o número de estimadores selecionados ou seja encontrado um estimador perfeito (GÉRON, 2019). O modelo final então utiliza as previsões e seus pesos para calcular o resultado final, isto

é, aquele que obteve se repetiu mais vezes. Segundo Géron (2019), esta decisão pode ser expressa matematicamente por:

$$\hat{y}(\mathbf{x}) = \arg \max_k \sum_{j=1}^N \alpha_j (\hat{y}_j^{(i)}(\mathbf{x}) = k) \quad (2.22)$$

onde k é a classe prevista pelo estimador.

Outro algoritmo *Boosting* bastante popular é o *Gradient Boosting* (GB). Similar ao AdaBoost, ele adiciona estimadores sequencialmente corrigindo os erros do estimador anterior. A diferença deste modelo para o outro é que ao invés de calcular os pesos para iteração, o GB realiza o ajuste do novo estimador utilizando o erro residual (GÉRON, 2019). Segundo a documentação de SCIKIT-LEARN (2020c), o valor previsto pelo modelo é dado por:

$$\hat{y}_i = F_m(x_i) = \sum_{m=1}^M h_m(x_i) \quad (2.23)$$

no qual \hat{y}_i representa o valor estimado, h_m são os estimadores do modelo e M é o número de estimadores.

Como qualquer outro método *Boosting*, os estimadores são adicionados sequencialmente, então têm-se que (SCIKIT-LEARN, 2020c):

$$F_m(x) = F_{m-1}(x) + h_m(x) \quad (2.24)$$

no qual h_m (uma nova árvore) é introduzido ao estimador anterior (F_{m-1}) para minimizar a soma de suas perdas (L_m).

Desse modo, a minimização da soma das perdas pode ser escrita da seguinte maneira (SCIKIT-LEARN, 2020c):

$$h_m = \arg \min_h L_m = \arg \min_h \sum_{i=1}^n l(y_i, F_{m-1}(x_i) + h(x_i)), \quad (2.25)$$

sendo o parâmetro $l(y_i, F(x_i))$ a função de perda ou custo do modelo do estimador.

De acordo com SCIKIT-LEARN (2020c), mesmo que este seja um método de classificação, função de perda é calculada com base no método dos mínimos quadrados. Os mínimos quadrados são escolhidos por possuírem robustez em resolver problemas de regressão (SCIKIT-LEARN, 2020c).

Inicialmente, F_0 é escolhido como a constante que minimiza a perda, isto é, a média empírica do valor alvo. Expandindo o termo $l(y_i, F_{m-1}(x_i) + h(x_i))$, utilizando a aproximação de Taylor, o valor de l pode ser aproximado por (SCIKIT-LEARN, 2020c):

$$l(y_i, F_{m-1}(x_i) + h_m(x_i)) \approx l(y_i, F_{m-1}(x_i)) + h_m(x_i) \left[\frac{\partial l(y_i, F(x_i))}{\partial F(x_i)} \right]_{F=F_{m-1}}. \quad (2.26)$$

o termo $\left[\frac{\partial l(y_i, F(x_i))}{\partial F(x_i)} \right]_{F=F_{m-1}}$ é a derivada da função perda com respeito ao segundo parâmetro, $F_{m-1}(x)$. Como l é diferenciável, o cálculo das derivadas parciais é resolvido facilmente.

Reescrevendo então a Equação 2.26, tem-se que (SCIKIT-LEARN, 2020c):

$$h_m \approx \arg \min_h \sum_{i=1}^n h(x_i) g_i \quad (2.27)$$

sendo o termo g_i o valor das derivadas parciais da Equação 2.26, também denominado de gradiente. Ademais, em cada iteração o estimador h_m é ajustado para atingir valores de gradiente negativos das amostras.

Como é utilizado um método de regressão para definir o modelo, o resultado não pode ser uma classe uma vez que as árvores de decisão preveem valores contínuos (SCIKIT-LEARN, 2020c). Para solucionar isto, segundo SCIKIT-LEARN (2020c), é feito um mapeamento do valor de $F_m(x_i)$ para uma classe ou probabilidade de pertencer a classe utilizando dependência de perda.

Para calcular o desvio (também chamado de *log-loss*), a probabilidade de x_i pertencer a uma classe positiva $P(y_i = 1|x_i) = \sigma(F_m(x_i))$ onde σ é a função sigmoide, discutida na Seção 2.4.2.1.

Conforme a documentação SCIKIT-LEARN (2020c), para regularizar as correções nos estimados, é introduzido um termo na Equação 2.24 referente a taxa de aprendizagem. Assim, cada vez que uma árvore (h_m) é adicionada ao modelo, a taxa de aprendizado regulariza o quanto o gradiente vai ser corrigido. A nova equação para o modelo se torna então (SCIKIT-LEARN, 2020c):

$$F_m(x) = F_{m-1}(x) + \eta h_m(x) \quad (2.28)$$

Finalmente, os hiperparâmetros avaliados pelo processo de validação cruzada nestes dois modelos foram: número de estimadores; taxa de aprendizagem e profundidade das árvores.

2.4.2.6 KNN e RNN

Os métodos *K - Nearest Neighbors* (KNN) e *Radius - Nearest Neighbors* (RNN) são, ambos, métodos baseados no conceito de distância e utilizam os exemplares do conjunto de treinamento para predizer novos dados (FACELI et al., 2011; SCIKIT-LEARN, 2020g).

A principal diferença entre os dois métodos a forma como estes escolhem os exemplares para compor o preditor: KNN utiliza um número específicos de amostras para classificação (os K Exemplares mais próximas) e RNN utiliza o conceito de densidade de pontos (as amostras dentro de uma região delimitada por um raio R) (SCIKIT-LEARN, 2020g).

Por ser uma técnica não paramétrica, apresenta resultados satisfatórios em situações de classificação quando os dados apresentam valores limites bem definidos. Apesar da simplicidade por trás deste conceito, estes métodos apresentam bons resultados em conjuntos de dados extensos (SCIKIT-LEARN, 2020g).

Segundo Faceli et al. (2011), a ideia do algoritmo é classificar um novo objeto com base nos exemplos do conjunto de treinamento que são mais próximos a ele. Para tal, utiliza-se do conceito distância para criar um modelo \hat{f} , sendo mais comum a utilização da distância Euclidiana, descrito na Eq. 2.29. Cada objeto associado a um rótulo representa um ponto no espaço definido pelas suas características. O algoritmo calcula a distância então como (FACELI et al., 2011):

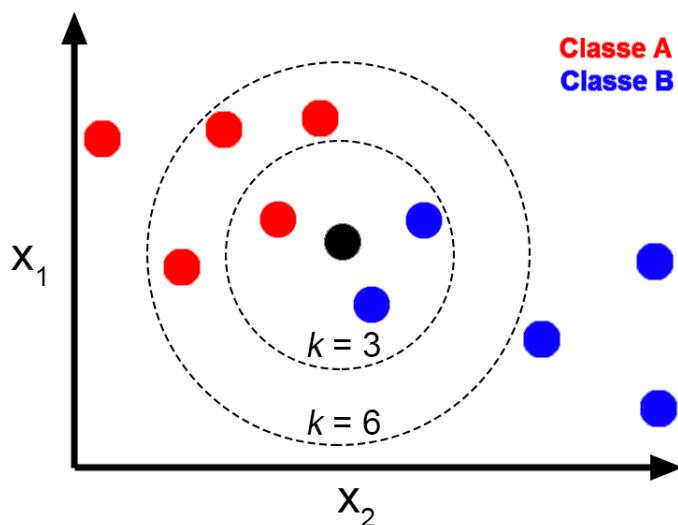
$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{l=1}^d (x_i^l - x_j^l)^2} \quad (2.29)$$

onde \mathbf{x}_i e \mathbf{x}_j são dois objetos representados por vetores no espaço das variáveis do conjunto de características e x_i^l, x_j^l são os elementos destes vetores, que correspondem aos valores das coordenadas l .

Segundo Faceli et al. (2011), na fase de treinamento, o algoritmo memoriza os rótulos dos objetos de treinamento e durante a predição de novos dados são obtidos K vizinhos mais próximos. Cada vizinho pertence a uma classe e o rótulo deste novo objeto é atribuído à classe que mais se repetiu entre os K vizinhos. Categorizando tal procedimento em termos matemáticos, têm-se que:

$$\hat{y}(\mathbf{x}_t) \leftarrow moda(f(\mathbf{x}_1), f(\mathbf{x}_2), f(\mathbf{x}_3), \dots, f(\mathbf{x}_K)) \quad (2.30)$$

Figura 37 – Impacto do valor de K no Exemplo de KNN

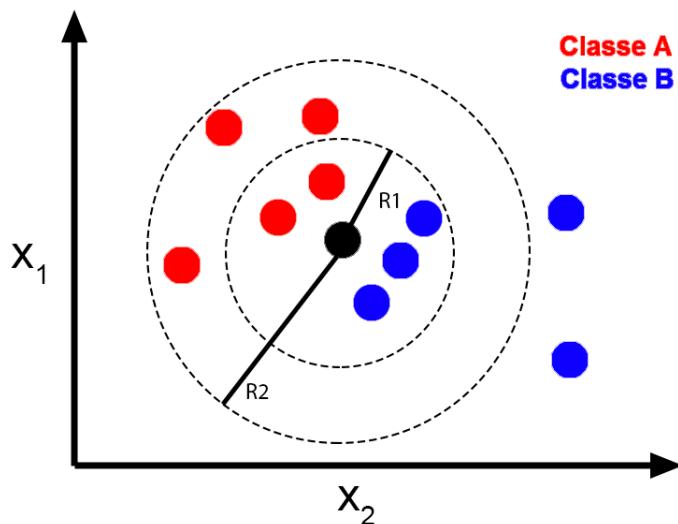


Fonte: Autor

Faceli et al. (2011) também afirma que escolher um valor de K apropriado não é uma tarefa trivial. De fato, como ilustra a Figura 37, a escolha de um $K = 3$ classificará o ponto vermelho como B e um $K = 6$ classificará o ponto como A , fazendo assim que uma escolha de K possa interferir na generalização do problema. Para solucionar isso, é possível estimar valores de K que estejam mais dentro do correto utilizando a validação cruzada (FACELI et al., 2011).

Com relação ao método RNN a diferença está na determinação do parâmetro distância (R), que irá ditar quantos objetos de treinamentos serão utilizados para avaliar o novo dado. A Figura 38 ilustra o funcionamento deste método para 2 valores de R distintos.

Figura 38 – Impacto do valor de R no Exemplo de KNN



Fonte: Autor

Ao utilizar R_1 , o algoritmo irá utilizar duas amostras A e três amostras B, logo classificará o ponto preto como B. Entretanto ao utilizar R_2 , o ponto será classificado como pertencente a classe A.

Outra forma de melhorar o funcionamento do algoritmo e fazer com que este generalize melhor o problema é a utilização de pesos as contribuições de cada vizinho (FACELI et al., 2011). Segundo Faceli et al. (2011) e a documentação (SCIKIT-LEARN, 2020g), é possível utilizar este hiperparâmetro como sendo inversamente proporcional a distância do objeto de treinamento ao de teste. Segundo Faceli et al. (2011), utilizar esta implementação dos pesos favorece pontos mais próximos do ponto teste.

É importante mencionar que os hiperparâmetros para estes dois modelos são k no KNN e R no RNN mais a utilização ou não dos pesos. Em ambos, os hiperparâmetros foram avaliados utilizando validação cruzada.

2.4.2.7 SVM

O *Support Machine Vector* (SVM), Máquina de Vetores de Suporte em português, é uma técnica de aprendizado de máquina baseado no aprendizado estático (FACELI et al., 2011). Essa teoria estabelece uma série de princípios que devem ser seguidos na obtenção de classificadores com uma boa capacidade de classificação (FACELI et al., 2011).

Em resumo, a teoria de aprendizado estático estabelece condições matemáticas (denominadas geométricas) para criar um classificador \hat{h} que gere saídas contínuas dentro de um conjunto de dados \mathbf{X} (FACELI et al., 2011). Segundo Faceli et al. (2011), os SVM utilizam este conceito e são definidos em subcategorias, sendo elas: SVM lineares e não lineares.

Os SVM lineares utilizam a hipótese dos dados apresentarem um comportamento linear. O modelo é então elucidado supondo que tais dados sejam separados linearmente e podem ser separados por uma função h descrita por (FACELI et al., 2011):

$$h(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b \quad (2.31)$$

sendo $\mathbf{w} \cdot \mathbf{x}$ é o produto escalar entre os vetores \mathbf{w} e \mathbf{x} e \mathbf{w} é o vetor normal ao hiperplano do conjunto e $\frac{b}{\|\mathbf{w}\|}$ é a distância do plano a origem.

A Equação 2.31 pode então ser usada para dividir um espaço amostral em duas categorias utilizando uma função sinal, ou seja (FACELI et al., 2011):

$$g(\mathbf{x}) = \operatorname{sgn}(h(\mathbf{x})) = \begin{cases} +1 & \text{se } \mathbf{w} \cdot \mathbf{x} + b > 0 \\ -1 & \text{se } \mathbf{w} \cdot \mathbf{x} + b < 0 \end{cases} \quad (2.32)$$

A Equação 2.32 mostra que a partir de $h(\mathbf{x})$ é possível obter infinitos hiperplanos que possa dividir a amostra. Então é necessário definir como plano canônico aquele cujo valores de \mathbf{w} e b que aproximam os pontos da amostra do plano $\mathbf{w} \cdot \mathbf{x} + b = 0$ e satisfazem a seguinte equação (FACELI et al., 2011):

$$|\mathbf{w} \cdot \mathbf{x} + b| = 1 \quad (2.33)$$

A Equação 2.33 implica no molde para classificação do modelo SVM para duas classes em uma amostra dada pelas inequações (FACELI et al., 2011):

$$\begin{cases} \mathbf{w} \cdot \mathbf{x} + b \geq +1 & \text{se } y_i = +1 \\ \mathbf{w} \cdot \mathbf{x} + b \leq -1 & \text{se } y_i = -1 \end{cases} \quad (2.34)$$

$$y_i(\mathbf{w} \cdot \mathbf{x} + b) - 1 \geq 0 , \quad \forall (\mathbf{x}_i, y_i) \in \mathbf{X} \quad (2.35)$$

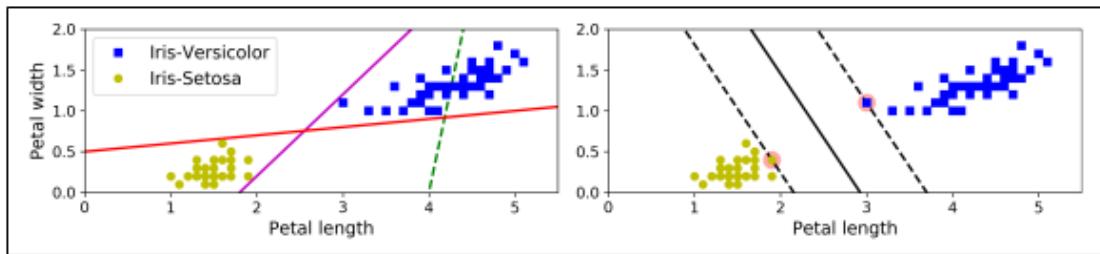
sendo \mathbf{X} o espaço dos dados amostrais.

O modelo então é ajustado para encontrar um pequeno \mathbf{w} para a Equação 2.34, isto é feito através da minimização da norma $\|\mathbf{w}\|$ como segue (FACELI et al., 2011):

$$\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2 \quad (2.36)$$

Segundo Faceli et al. (2011), as restrições impostas pelo sistema de Equações 2.34 e a Equação 2.36 são impostas de maneira a assegurar que não haja dados entre as margens de separação (supondo dados totalmente separáveis). Por esse motivo, este modelo é denominado de SVM linear com margens rígidas (ver Figura 39).

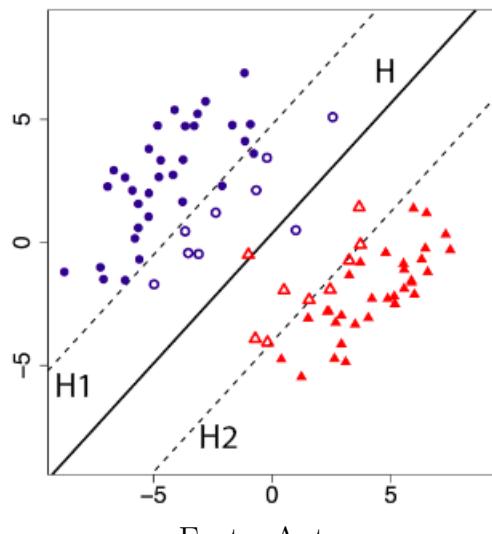
Figura 39 – Exemplo de classificação utilizando o modelo SVM linear com margens rígidas



Fonte: (GÉRON, 2019)

Porém em situações reais, exemplificadas pelas Figura 40, é difícil encontrar dados que sejam totalmente separáveis linearmente devido a presença de ruídos, *outliers* ou a própria natureza do problema (FACELI et al., 2011).

Figura 40 – Exemplo de classificação utilizando o modelo SVM linear com margens suaves



Fonte: Autor

Para contornar este problema, é introduzido variáveis de folga que suavizam (relaxam) as restrições, denominado por $\zeta_i \forall i = 1, 2, \dots, n$. Logo, a Equação 2.35 se torna

([FACELI et al., 2011](#)):

$$y_i(\mathbf{w} \cdot \mathbf{x} + b) \geq 1 - \zeta_i, \quad \zeta_i \geq 0, \quad \zeta_i \quad \forall i = 1, 2, \dots, n \quad (2.37)$$

A aplicação desta ideia suaviza as margens dos classificadores lineares permitindo que alguns objetos permaneçam entre os hiperplanos H1 e H2. A este tipo de SVM linear é dado o nome de SVM linear de margens suaves ([FACELI et al., 2011](#)).

Utilizando este conceito, o problema de otimização para encontrar um bom estimador é dado por ([SCIKIT-LEARN, 2020j](#); [FACELI et al., 2011](#)):

$$\min_{w,b,\zeta} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \zeta_i \quad (2.38)$$

onde C é um termo de regularização que impõe um peso à minimização dos erros no conjunto de treinamento.

Segundo [Faceli et al. \(2011\)](#) e [Géron \(2019\)](#), o modelo SVM linear é eficaz na classificação de amostras que apresentam certa linearidade, porém existem diversos casos onde a sua utilização perde a generalização do problema. Tais situações são resolvidas utilizando os SVM não lineares.

O conceito básico por trás destes modelos é lidar com problemas não lineares mapeando o conjunto de treinamento no seu espaço original para um espaço de maior dimensão, referido como *feature space* ([FACELI et al., 2011](#)). Segundo [Faceli et al. \(2011\)](#), dado um conjunto não linear de espaço de entrada X , existe uma função Φ que permite mapear o espaço de entrada em um novo espaço \mathfrak{I} , sendo que a escolha adequada da função Φ permite com que o conjunto de treinamento venha no novo espaço possa ser separado linearmente.

Aplicando este conceito na Equação 2.31, têm-se que ([FACELI et al., 2011](#)):

$$h(\mathbf{x}) = \mathbf{w} \cdot \Phi(\mathbf{x}) + b \quad (2.39)$$

onde a função Φ mapeia o espaço X encontrando um hiperplano no novo espaço \mathfrak{I} com maior margem de separação, garantindo assim uma boa generalização do problema.

Segundo [Faceli et al. \(2011\)](#), como \mathfrak{I} pode ter dimensões muito altas (até mesmo infinitas), pode-se ter um problema computacional muito custoso. Porém, para solucionar o problema de maneira mais simples, é introduzido funções que calculam o produtos escalares entre os objetos no espaço de características, tal função é denominada kernel ([FACELI et al., 2011](#)).

Pela definição, um kernel K é uma função que recebe os pontos \mathbf{x}_i e \mathbf{x}_j no espaço de entrada e calculada o produto escalar destes dois objetos, isto é ([FACELI et al., 2011](#)):

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \quad (2.40)$$

É comum empregar a utilização da função kernel sem conhecer necessariamente o mapeamento Φ , que acaba sendo gerado implicitamente (FACELI et al., 2011). Desta forma, o problema para aplicar tal modelo acaba sendo dado pelas equações (FACELI et al., 2011; SCIKIT-LEARN, 2020j):

$$\min_{w,b,\zeta} \frac{1}{2} ||\mathbf{w}|| + C \sum_{i=1}^n \zeta_i \quad (2.41)$$

$$y_i(\mathbf{w} \cdot \Phi(x_i) + b) \geq 1 - \zeta_i, \quad \zeta_i \geq 0, \quad i = 1, \dots, n \quad (2.42)$$

E como o problema tende a maximizar as margens (reduzindo $||\mathbf{w}||$), utiliza-se (SCIKIT-LEARN, 2020j):

$$\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha \quad (2.43)$$

$$y^T \alpha = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \dots, n \quad (2.44)$$

onde e é um vetor composto por uns, Q é uma matriz quadrada positiva definida por $Q_{ij} \equiv y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$. Os termos α_i é denominado de *dual coefficients*. Esta representação dual destaca o fato de que os vetores de treinamento são implicitamente mapeados em um espaço dimensional superior (talvez infinito) pela função Φ .

Uma vez que o problema de otimização é resolvido, a função decisão (função que determinada a classificação) é dada por (FACELI et al., 2011):

$$\hat{f} = \sum_{i \in SV} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b, \quad (2.45)$$

onde a classe prevista corresponde ao valor da função sinal, precisando apenas somar os vetores de suporte (ou seja, as amostras que estão dentro da margem).

Na presente monografia, no intuito de avaliar a melhor solução possível, foram utilizados três funções de kernel K distintas, presentes na Tabela 3. Sendo estes, segundo Faceli et al. (2011), os mais utilizados na prática: Polinomiais, RBF (*radial basis function*) e Sigmoidal. Ademais, acrescenta-se que os parâmetros expostos na Tabela 3 são hiperparâmetros determinados pelo usuário e não apresentam explicação teórica.

Tabela 3 – Tipos de Kernels avaliados na otimização das SVM

Tipo	Função $K(\mathbf{x}_i, \mathbf{x}_j)$	Parâmetros
Polinomial	$(\delta(\mathbf{x}_i \cdot \mathbf{x}_j) + k)^d$	δ, k e d
RBF	$\exp(-\sigma \mathbf{x}_i - \mathbf{x}_j ^2)$	σ
Sigmoidal	$\tanh(\delta(\mathbf{x}_i \cdot \mathbf{x}_j) + k)$	δ e k

Fonte: (FACELI et al., 2011).

Por fim, é feito a ressalva de que foram avaliados os quatro modelos de SVM (linear, polinomial, RBF e sigmoidal) para vários hiperparâmetros distintos através do processo de validação cruzada.

2.4.2.8 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNA), ou *Artificial Neural Network* do inglês, é uma ferramenta de ML massivamente utilizada para problemas de regressão e de classificação. A ideia desse modelo surgiu no começo da década de 40 com o progresso da Neurocomputação. Ela é fundamentada em criar modelos computacionais de altíssima funcionalidade com funcionamento similar ao do cérebro humano (SILVA; SPATTI; FLAUZINO, 2010).

Segundo Faceli et al. (2011) e Géron (2019), os primeiros estudos acerca deste tema foram desenvolvidos pelo neurofisiologista Warren McCulloch e o matemático Walter Pitts. Em 1943 eles propuseram um modelo matemático de neurônio artificial que executavam funções logísticas simples (FACELI et al., 2011). Foram eles também responsáveis por demonstrar que uma rede formada por esses neurônios possuía grande capacidade de descrever funções complexas (FACELI et al., 2011).

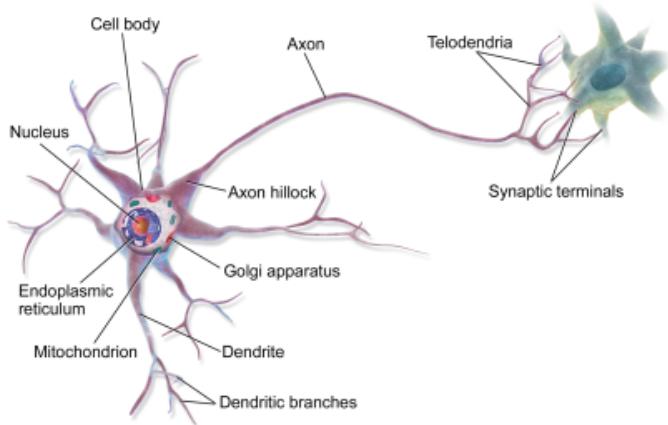
Estes trabalhos iniciais, segundo Faceli et al. (2011), tinham por objetivo compreender o sistema nervoso e o cérebro humano para utilizar deste conhecimento no intuito de desenvolver sistemas de aprendizado biologicamente plausíveis. Então para entender o funcionamento de uma RNA, é essencial entender as estruturas básicas de um cérebro humano e seu sistema nervoso.

O sistema nervoso, do qual o cérebro faz parte, é um conjunto complexo de células que determinam o funcionamento dos seres vivos, sendo sua unidade básica o neurônio (GÉRON, 2019; FACELI et al., 2011). Esta célula, representada pela Figura 41, é formada por corpo celular contendo o núcleo e estruturas celulares, muitas extensões menores chamadas de dendritos e uma extensão longa denominada axônio (GÉRON, 2019). Perto da extremidade do axônio, este se divide em diversos ramos chamados telodendros e na ponta destes componentes existem regiões denominadas sinapses (GÉRON, 2019).

O sinal biológico transmitido entre os neurônios é realizado por um impulso elétrico conduzido pelo axônio até chegar na sinapse e se comunicar ou não com outro neurônio (FACELI et al., 2011). Ao realizar esta comunicação de forma específica, o cérebro pode gerar respostas no corpo humano como pensamentos e ações.

As Redes Neurais Artificiais se baseiam neste mecanismo de funcionamento do sistema nervoso. Desse modo, elas são compostas por neurônicos artificiais, que são densamente interconectados e estes são capazes de computar funções matemáticas (FACELI et al., 2011). Estas unidades são dispostas de uma ou mais camadas interligadas por um grande número de conexões, normalmente unidireccionais (FACELI et al., 2011).

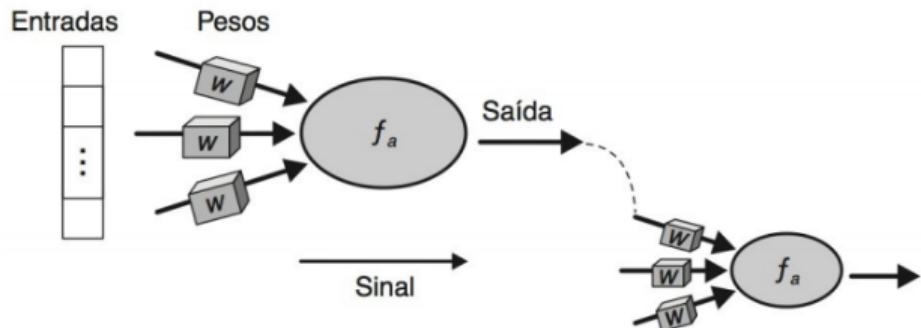
Figura 41 – Neurônio Biológico



Fonte: (GÉRON, 2019)

Além disso, as conexões entre neurônios possuem pesos associados, que regulam a atividade do neurônio na rede (FACELI et al., 2011). De acordo com (FACELI et al., 2011), estes pesos são ajustados durante o processo de aprendizado, que nesta monografia se configura como supervisionado (SCIKIT-LEARN, 2020h). Acrescenta-se ainda que os pesos podem possuir valores positivos ou negativos, isto é, um comportamento excitatório, quando positivos, e inibitório, quando negativos (FACELI et al., 2011).

Figura 42 – Neurônio Artificial



Fonte: (FACELI et al., 2011)

A Figura 42 ilustra um modelo de neurônio artificial. As entradas representam a admissão dos dados no neurônio, uma função matemática (f_a) é responsável por ponderá-los, utilizando os pesos w e associá-los emitindo um sinal de resposta. É possível utilizar várias funções f_a diferentes nestes modelos (FACELI et al., 2011).

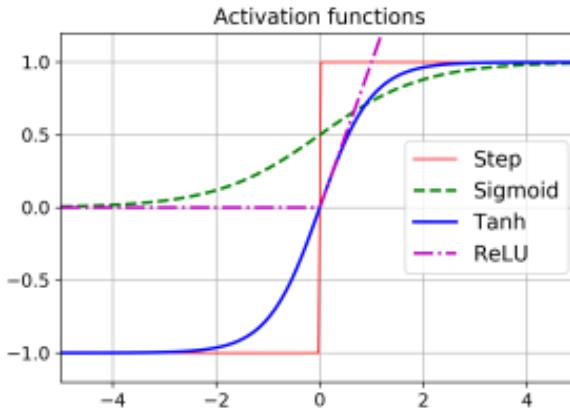
Supondo então um objeto \mathbf{x} com d atributos representado na forma de um vetor $\mathbf{x} = [x_1, x_2, \dots, x_d]$ e um neurônio com d terminais de entrada cujos pesos são w_1, w_2, \dots, w_d representados pelo vetor \mathbf{w} , pode-se escrever a entrada do neurônio como (FACELI et al.,

2011):

$$u = \sum_{j=1}^d x_j w_j \quad (2.46)$$

A saída do neurônio é definido através da aplicação de uma função de ativação (f_a) com este valor de entrada (FACELI et al., 2011). A Figura 43 mostra exemplo de comportamentos possíveis para 4 funções de ativações distintas (*Step, Sigmoid, Tanh, ReLU*).

Figura 43 – Exemplos de funções de ativação



Fonte: (GÉRON, 2019)

Matematicamente pode-se definir estas funções, utilizando os valores da Figura 43. A função *Step* ou Degrau, impõe uma saída igual 0 ou 1, é comum também usar uma função limiar que resulte em 1 ou -1, como mostra a figura (GÉRON, 2019; FACELI et al., 2011). Esta por sua vez é expressa por (SILVA; SPATTI; FLAUZINO, 2010):

$$f_a(u) = \begin{cases} -1 & \text{se } u \geq 0 \\ 1 & \text{se } u < 0 \end{cases} \quad (2.47)$$

Para a função *Sigmoid*, apresenta o mesmo comportamento da função logística apresentada na Seção 2.4.2.1. Porém no caso das redes neurais, esta é descrita como (SILVA; SPATTI; FLAUZINO, 2010):

$$f_a(u) = \frac{1}{1 + e^{-u}} \quad (2.48)$$

Para a função *Tanh*, esta se comporta como uma Tansigmoide, isto é (SILVA; SPATTI; FLAUZINO, 2010):

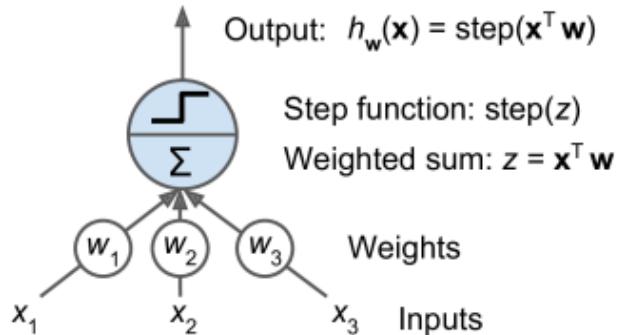
$$f_a(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}} \quad (2.49)$$

Já a função *ReLU* apresenta um comportamento nulo até $u = 0$ e após a origem, $f_a = u$ (SILVA; SPATTI; FLAUZINO, 2010):

$$f_a(u) = \begin{cases} 0 & \text{se } u \leq 0 \\ u & \text{se } u > 0 \end{cases} \quad (2.50)$$

Em 1957, Frank Rosenblatt inventou o *Perceptron*, uma das estruturas mais simples das RNA (GÉRON, 2019). Esta estrutura apresenta uma unidade linear (Figura 44) onde os dados de entrada são submetidos a soma ponderada, como dito anteriormente e aplicados em uma função de ativação f_a (GÉRON, 2019). A Figura 44 apresenta a estrutura da primeira e mais simples rede perceptron, que era composta por apenas um neurônio (FACELI et al., 2011).

Figura 44 – Unidade Linear com função de ativação *Step*



Fonte: (GÉRON, 2019)

O treinamento desta rede perceptron se baseia no ajuste dos pesos de entrada perante os erros de predição durante o treino. Segundo Faceli et al. (2011), os ajustes nos pesos desta rede de uma única camada se dão pela seguinte maneira:

$$w_j(t+1) = w_j(t) + \eta x_i^j (y_i - \hat{f}(\mathbf{x}_i)) \quad (2.51)$$

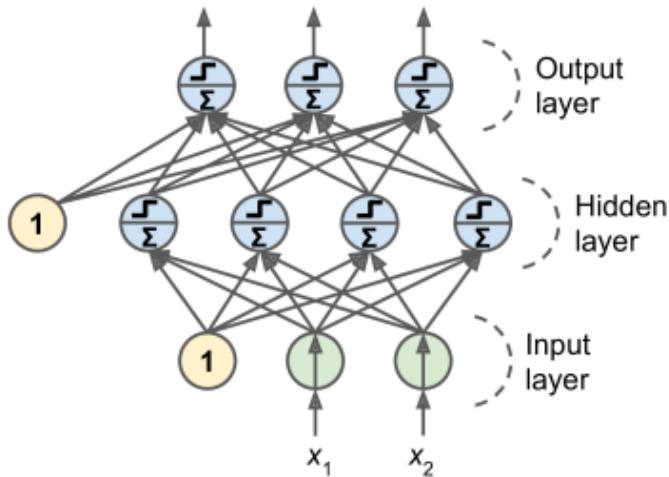
onde $w_j(t)$ é o peso da j-ésima conexão no instante de tempo t , η é a taxa de aprendizado, x_i^j é o valor do j-ésimo atributo do elemento \mathbf{x}_i e $\hat{f}(\mathbf{x}_i)$ é a saída prevista pela rede.

Segundo Faceli et al. (2011), uma RNA pode apresentar diversos neurônios dispostos em uma ou mais camadas. A esta estrutura da-se o nome de *Multi Layer Perceptron* (MLP). A Figura 45 exemplifica a configuração mais comum para tipo de rede, onde têm como características principais três conjunto de camadas: camada de entrada, onde os dados são submetidos; camada oculta, onde estão presentes os perceptrons e a camada de saída.

Nessa configuração, todos os neurônios de uma camada l estão conectados a todos os neurônios de uma camada $l+1$ (FACELI et al., 2011). Caso l seja a primeira camada da rede, esta estará conectada a todos os atributos de entrada $x_j (j = 1, 2, \dots, d)$. A camada de saída é arquitetada para cada tipo de tarefa, podendo ter apenas um único neurônio em problemas de classificação binário ou n neurônios para classificar n classes.

O maior problema durante muito tempo para a implementação deste modelo foi a ausência de uma técnica que permitisse o treinamento destas redes (FACELI et al., 2011).

Figura 45 – Componentes de uma RNA-MLP



Fonte: Autor

Este problema foi resolvido devido a utilização do algoritmo *back-propagation*, baseado em gradiente descendente (FACELI et al., 2011).

Este método de aprendizagem é constituído de duas fases de iteração: uma fase para frente (*forward*) e outra para trás (*backward*). O algoritmo back-propagation funciona da seguinte maneira: a camada de entrada aplica os dados para a primeira camada intermediária, onde ocorre a aplicação da função de ativação pelos neurônios, que geram valores de saída, os quais são transmitidos para as próximas camadas (FACELI et al., 2011). A camada seguinte utiliza o resultado das camadas anteriores para aplicar suas funções de ativação e o processo continua até os neurônios da camada de saída produzirem seus valores de saída $f_b(\mathbf{x}_i)$. Então, $f_b(\mathbf{x}_i)$ é comparado com o valor desejado y_i (FACELI et al., 2011).

O valor do erro da rede é calculado para ajustar os pesos de entrada dos neurônios da camada de saída até a primeira camada através da equação de ajuste dos pesos para o algoritmo back-propagation (FACELI et al., 2011):

$$w_{jl}(t+1) = w_{jl}(t) + \eta \mathbf{x}^j \delta_l \quad (2.52)$$

no qual w_{jl} o valor do peso entre o neurônio l e a j -ésima característica de uma amostra \mathbf{x}_i ou j -ésimo valor de saída para um neurônio de camadas internas. δ_l representa o erro associado ao neurônio l e x^j o valor da entrada recebida pelo neurônio.

Os erros são conhecidos apenas nos neurônios da camada de saída (FACELI et al., 2011). Desse modo, os demais precisam ser estimados e retropropagados para o restante da rede. Sendo assim, os erros dos neurônios das camadas intermediárias são estimados com base nos erros da próxima camada (FACELI et al., 2011). A formulação empregada

nas estimativas dos erros, pode ser apresentada a seguir (FACELI et al., 2011):

$$\delta_l = \begin{cases} f'_a e_l & \text{se } n_l \in c_{sai} \\ f'_a \sum w_{lk} \delta_k & \text{se } n_l \in c_{int} \end{cases} \quad (2.53)$$

onde n_l representa o neurônio l , c_{sai} a camada de saída, c_{int} a camada intermediária, f'_a a derivada parcial da função de ativação e e_l o erro do neurônio da camada de saída.

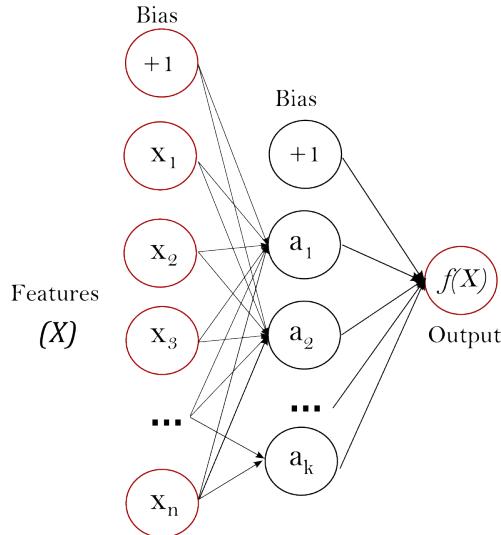
O termo e_l possui a seguinte formulação quadrática (FACELI et al., 2011):

$$e_l = \frac{1}{2} \sum_{q=1}^k (y_q - \hat{f}_q)^2 \quad (2.54)$$

Segundo Faceli et al. (2011), a derivada da função de ativação norteia o algoritmo sobre o ajuste dos pesos. Desse modo, se o f'_a for positivo um ajuste deve ser feito no sentido de minimizá-la para diminuir o erro associado.

Segundo a documentação (SCIKIT-LEARN, 2020h), as RNA-MLP apresentam um termo numérico adicionado em cada camada oculta chamada *bias*. Tal termo é uma grandeza numérica que auxilia no aprendizado da rede se comportando como um dado de entrada. A Figura 46 mostra a aplicação deste termo em uma rede neural de 1 camada oculta.

Figura 46 – RNA-MLP de 1 camada oculta



Fonte: (SCIKIT-LEARN, 2020h)

Por fim, os hiperparâmetros para RNA-MLP avaliados nesta monografia foram: número de camadas (1,2 e 3 camadas ocultas), número de neurônios por camada e qual função de ativação na saída estarão utilizando. Ademais, adotou-se como taxa de aprendizado o valor de $\eta = 0,001$.

2.4.3 Métricas de avaliação

Após a criação de um modelo para predição de um resultado, é necessário realizar um teste para definir métricas que corroboram se aquele algoritmo é bom ou não para generalizar determinado problema.

Avaliar um classificador é muitas vezes uma tarefa mais árdua do que avaliar um regressor (GÉRON, 2019). Segundo Géron (2019), existem diversas medições de desempenho que possam ser utilizadas para avaliar a eficácia de um classificador.

Para realizar um entendimento adequado das métricas envolvendo a predição de um problema binário e multiclasse, esta Seção se baseará nas definições presentes na monografia de Faceli et al. (2011) e documentação SCIKIT-LEARN (2020e).

Por simplicidade, seja um problema de duas classes (binário), denotando-se um classe positiva (a presença de câncer) e outra negativa, têm-se a matriz de confusão ilustrada na Figura 47 em que (FACELI et al., 2011):

- VP (Verdadeiro Positivo) corresponde ao número de exemplos da classe positiva classificados corretamente
- VN (Verdadeiro Negativo) corresponde ao número de exemplos da classe negativa classificados corretamente
- FP (Falso Positivo) corresponde ao número de exemplos da classe negativa classificados como positivos quando o valor real deste é negativa
- FN (Falso Negativo) corresponde ao número de exemplos da classe positiva classificados como negativos quando o valor real deste é positivo

Ademais, $n = VP + VN + FP + FN$ representa o número total de amostras submetidas a teste.

Figura 47 – Matriz de confusão para um problema binário

		Detectada	
		Sim	Não
Real	Sim	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	Não	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Fonte: Autor.

Dentro destes valores, é possível também utilizar outras métricas para verificar a eficiência de um modelo, sendo elas (FACELI et al., 2011; SCIKIT-LEARN, 2020e):

- **Accuracy score:** acurácia em português, representa quantos valores o modelo realmente acertou se era realmente positivo, ou realmente negativo

$$Ac(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^{n-1} 1(\hat{y}_i = y_i) = \frac{VP + VN}{n} \quad (2.55)$$

- **Precision:** precisão em português, proporção dos exemplos positivos classificados corretamente entre todos aqueles preditos como positivos por \hat{f}

$$Pr = \frac{VP}{VP + FP} \quad (2.56)$$

- **Recall:** revocação em português, corresponde à taxa de acerto na classe positiva

$$Rec = \frac{VP}{VP + FN} \quad (2.57)$$

- **Especificidade:** corresponde à taxa de acerto na classe negativa

$$Esp = \frac{VN}{VN + FP} \quad (2.58)$$

Outra métrica importante, segundo [SCIKIT-LEARN \(2020e\)](#), é a utilização do *Balanced accuracy score* (B_S). Tal métrica é boa para avaliar corretamente a performance de estimados com conjunto de dados balanceados, o caso deste projeto ([SCIKIT-LEARN, 2020e](#)). O cálculo desta métrica é dado por ([SCIKIT-LEARN, 2020e](#)):

$$B_S = \frac{1}{2} \left(\frac{VP}{VP + FN} + \frac{VN}{VN + FP} \right) \quad (2.59)$$

Se o classificador executa igualmente bem em qualquer classe, este termo se reduz à Ac convencional (ou seja, o número de previsões corretas dividido pelo número total de previsões) ([SCIKIT-LEARN, 2020e](#)). Em contraste, se a Ac convencional estiver elevado apenas porque o classificador tira vantagem de um conjunto de teste desequilibrado, então a precisão balanceada obtém um métrica para avaliar a generalização do problema de maneira mais correta ([SCIKIT-LEARN, 2020e](#)).

Segundo [Faceli et al. \(2011\)](#), a precisão pode ser vista como uma medida de exatidão do modelo e a revocação de sua completude. Por exemplo, se $Pr = 1$, significa que todos os itens rotulados como pertencentes a uma classe C são realmente da classe C . Já, se $Rec = 1$, significa que todos os itens da classe C foram rotulados como C pelo modelo, porém não informa quantas previsões destas rotuladas estão corretas ou não ([FACELI et al., 2011](#)).

É conveniente então combinar precisão e revocação em uma única métrica chamada *F measure* (Medida F) ([GÉRON, 2019](#)). A precisão e revocação não são discutidas

isoladamente mas sim combinadas em uma métrica cuja expressão é dada por (FACELI et al., 2011):

$$F_\beta = (1 + \beta^2) \frac{Pr \times Rec}{\beta^2 Pr + Rec}. \quad (2.60)$$

Se utilizar $\beta = 1$, têm-se um mesmo grau de importância a precisão e revocação, isto é (FACELI et al., 2011):

$$F_1 = 2 \frac{Pr \times Rec}{Pr + Rec}. \quad (2.61)$$

A métrica F_1 favorece classificadores que têm precisão e revocação semelhantes, entretanto isso nem sempre é o ideal (GÉRON, 2019). Segundo Géron (2019), em alguns contextos, uma das duas métricas se torna mais importante como no exemplo desta monografia. No problema de predição de câncer, é melhor ter um estimador que dê sinal positivo para a presença de câncer, mesmo que o paciente não tenha a doença de fato.

Outra métrica pertinente a este trabalho é o Coeficiente de Correlação de Matthews (MCC do inglês). O coeficiente de correlação de Matthews é usado no aprendizado de máquina como uma medida da qualidade das classificações binárias (duas classes) (SCIKIT-LEARN, 2020e). O MCC é basicamente um valor de coeficiente de correlação entre -1 e +1. Um coeficiente de +1 representa uma predição perfeita, 0 uma predição aleatória média e -1 uma predição inversa (SCIKIT-LEARN, 2020e).

No problema binário, esta métrica é dada pela expressão (SCIKIT-LEARN, 2020e):

$$MCC = \frac{VP \times VN - FP \times FN}{\sqrt{(VP + FP)(VP + FN)(VN + FP)(VN + FN)}}. \quad (2.62)$$

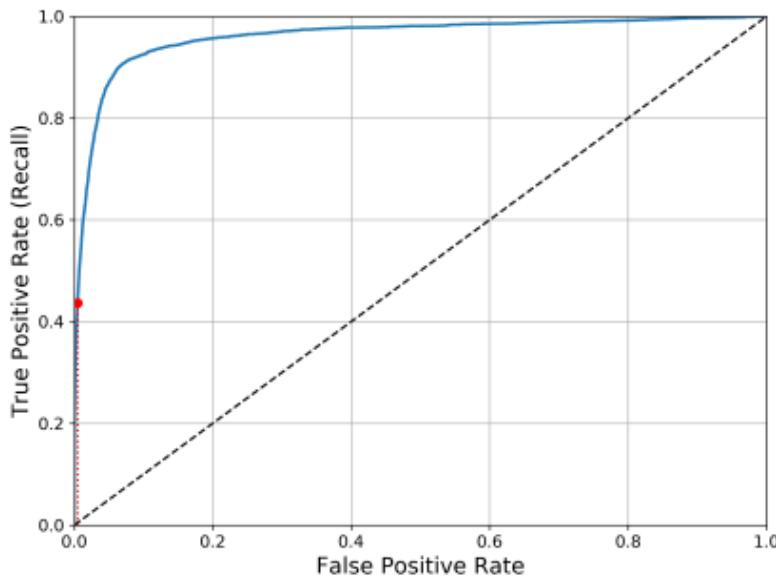
Uma forma alternativa de avaliar classificadores em problemas binários é utilizando as curvas ROC (*Receiving Operating Characteristics* (FACELI et al., 2011)). O gráfico ROC é um gráfico bidimensional com eixos X e Y representando as medidas de taxa de falso positivo (TFP) e taxa de verdadeiros positivos (TVP), respectivamente (FACELI et al., 2011). A Figura 48 ilustra o espaço ROC e sua representação gráfica.

A linha diagonal da Figura 48 representa classificadores que realizam predições totalmente aleatórias e a curva azul representa o desempenho de um estimador qualquer. Segundo 42, classificadores abaixo da linha diagonal podem ser considerados classificadores ruins.

Para montar um gráfico ROC, o estimador define uma probabilidade da amostra pertencer a classe positiva e então os valores contínuos destas taxas são discretizados utilizando um valor limiar (por exemplo $Prob \geq 0.5$) (FACELI et al., 2011).

Utilizando a Curva ROC característica do modelo, é comum extrair uma métrica numérica denominada AUC (*Area Under the Curve*) (FACELI et al., 2011). Esta medida produz valores entre 0 e 1, que representam a área sobre a Curva ROC do modelo. Ademais,

Figura 48 – Gráfico ROC



Fonte: (GÉRON, 2019).

valores próximos 0 significa que a previsão do modelo está invertida, próximo de 0.5, um modelo prevê de maneira totalmente aleatória e mais próximo de 1 que o modelo tem uma boa previsão. Uma das principais vantagens da análise ROC é fornecer mais uma métrica para comparação de modelos, sendo esta independente do número de acerto ou erros do teste (FACELI et al., 2011).

Uma desvantagem da análise ROC é que esta é originalmente limitada a problemas binários, entretanto, com o avanço da computação e novas teorias, métodos para adquirir tal medida já estão disponíveis (FACELI et al., 2011; SCIKIT-LEARN, 2020e).

Segundo Faceli et al. (2011), o problema do multiclasse pode ser facilmente avaliado utilizando cada classe como positiva e comparando-a com o resto negativo. Este método fornece um valor de desempenho para cada classe que pode ser combinados para criar um valor único pra o problema.

A partir daqui, é feito o estudo sobre a as métricas de desempenho mencionadas anteriormente para o caso multiclasse. Na tarefa de classificação multiclasse, as noções de precisão, revocação e medidas F podem ser aplicadas a cada rótulo independentemente (SCIKIT-LEARN, 2020e).

Existem algumas maneiras de combinar estes resultados entre rótulos através de uma média ou combinação específica de problemas binários. Segundo a documentação de SCIKIT-LEARN (2020e), a forma como estes valores são combinados computacional é especificado pelo argumento *average*. É importante notar que apesar de existir mais de um modelo para a combinação das métricas binárias, esta monografia utilizou da opção *weighted* pois esta é mais indicada para conjunto de dados desbalanceados.

Tomando como notação ([SCIKIT-LEARN, 2020e](#)):

- \hat{y} o conjunto de pares previstos (amostra, classe)
- y o conjunto de pares reais (amostra, classe)
- L o conjunto de classes
- S o conjunto de amostras
- y_s o sub-conjunto de y da amostra s , isto é, $y_s := \{(s', l) \in y | s' = s\}$
- y_l o sub-conjunto de y de classe l
- \hat{y}_s e \hat{y}_l seguem a mesma lógica para o conjunto de \hat{y}
- $Pr(A, B) := \frac{|A \cap B|}{|A|}$ para algum conjunto A e B
- $Rec(A, B) := \frac{|A \cap B|}{|B|}$
- $F_\beta(A, B) := (1 + \beta^2) \frac{Pr(A, B) \times Rec(A, B)}{\beta^2 Pr(A, B) + Rec(A, B)}$

As novas métricas são então definidas como mostra a Tabela 4 ([SCIKIT-LEARN, 2020e](#)):

Tabela 4 – Tabela de métricas para problemas multiclasse

<i>Average</i>	<i>Precision</i>	<i>Recall</i>	F_β
Micro	$Pr(y, \hat{y})$	$Rec(y, \hat{y})$	$F_\beta(y, \hat{y})$
Samples	$\frac{1}{ S } \sum_{s \in S} Pr(y_s, \hat{y}_s)$	$\frac{1}{ S } \sum_{s \in S} Rec(y_s, \hat{y}_s)$	$\frac{1}{ S } \sum_{s \in S} F_\beta(y_s, \hat{y}_s)$
Macro	$\frac{1}{ L } \sum_{l \in L} Pr(y_l, \hat{y}_l)$	$\frac{1}{ L } \sum_{l \in L} Rec(y_l, \hat{y}_l)$	$\frac{1}{ L } \sum_{l \in L} F_\beta(y_l, \hat{y}_l)$
Weighted	$\frac{1}{\sum_{l \in L} \hat{y}_l } \sum_{l \in L} \hat{y}_l Pr(y_l, \hat{y}_l)$	$\frac{1}{\sum_{l \in L} \hat{y}_l } \sum_{l \in L} \hat{y}_l Rec(y_l, \hat{y}_l)$	$F_\beta(y, \hat{y})$

Fonte: ([SCIKIT-LEARN, 2020e](#)).

Para o valor de B_S , é calculado a média da precisão e revogação para cada classe e então feito uma média de todos os resultados ([SCIKIT-LEARN, 2020e](#)).

Nos problema multiclasse, o Coeficiente de Matthews é definido em termos da matriz de confusão, utilizando as seguintes definições ([SCIKIT-LEARN, 2020e](#)):

- $t_k = \sum_i^K C_{ik}$ o número de vezes que a classe k foi prevista corretamente
- $p_k = \sum_i^K C_{ki}$ o número de vezes que a classe k foi prevista
- $c = \sum_k^K C_{kk}$ o número amostras corretamente previstas

- $s = \sum_i^K \sum_j^K C_{ij}$ o número total de amostras

O novo valor de MMC é então descrito por (SCIKIT-LEARN, 2020e):

$$MCC = \frac{c \times s - \sum_k^K p_k \times t_k}{\sqrt{(s^2 - \sum_k^K p_k^2) \times (s^2 - \sum_k^K t_k^2)}} \quad (2.63)$$

Como a análise ROC foi inicialmente idealizada para o problema binário, o problema multiclasse é resolvido utilizando o conceito binário e estendendo-o através de uma média (SCIKIT-LEARN, 2020e). Duas estratégias de média são atualmente suportadas, segundo documentação SCIKIT-LEARN (2020e): o algoritmo Um-Vs-Um calcula a média das pontuações ROC AUC e o algoritmo Um-Vs-Resto calcula a média das pontuações ROC AUC para cada classe contra todas as outras classes. Nesta monografia, por questão de padronização, será utilizada a primeira.

Esse método calcula o valor de AUC de todos os possíveis pares de combinação de classe através do cálculo uniforme (SCIKIT-LEARN, 2020e):

$$ROC_S = \frac{2}{c(c-1)} \sum_{j=1}^c \sum_{k>j}^c (\text{AUC}(j|k) + \text{AUC}(k|j)) \quad (2.64)$$

onde c é o número de classes, $\text{AUC}(j|k)$ é o AUC com a classe j como positiva e k negativa. Sendo no geral, $\text{AUC}(j|k) \neq \text{AUC}(k|j)$ no caso multiclasse.

No intuito de facilitar a visualização dos resultados e classificar os modelos treinados nesta monografia, buscou-se aplicar todas estas métricas para tentar descrever todos os aspectos importantes em um problema de classificação. Segundo Faceli et al. (2011), comparar vários modelos entre si não é uma tarefa fácil. Cada projeto tem um propósito diferente e portanto uma métrica diferente pode ser criada para determinar o desempenho do algoritmo (GÉRON, 2019).

Para esta monografia, utilizou-se uma métrica criada especialmente para projeto que realiza uma média aritmética do B_S , ROC_S e F_β com $\beta = 0.5$, isto é:

$$F_M = \frac{B_S + ROC_S + F_\beta}{3} \quad (2.65)$$

3 Metodologia

3.1 Fluxo de trabalho

A metodologia empregada nesta monografia pode ser resumida pelo fluxo de trabalho representado na Figura 49.

Figura 49 – Fluxo de trabalho desenvolvido para esta monografia



Fonte: Autor.

Está disponível em <<https://github.com/thiagolopes97/TCC-ThiagoLopes>> a implementação computacional desenvolvida nesta metodologia e todos os resultados provenientes dela.

3.2 Conjunto de dados

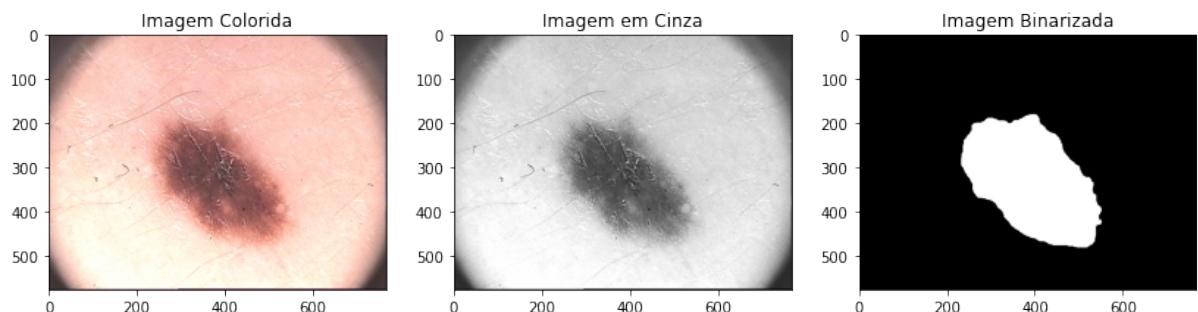
Como mencionado na Seção 2.4.1, a primeira etapa para analisar o problema é entender o conjunto de dados e suas informações. Assim, esta etapa da metodologia foi executada pensando em entender os conjuntos de dados PH2 (MENDONÇA et al., 2013) e HAM100000 (TSCHANDL, 2018), suas imagens e os conjuntos de informações que estes trazem.

O conjunto de dados PH2 é um conjunto de imagens obtidas pelo Serviço Dermatológico do Hospital Pedro Hispano em Matosinhos, Portugal (MENDONÇA et al., 2013). As imagens dermatoscópicas foram obtidas utilizando o mesmo procedimento e condições (MENDONÇA et al., 2013). Realizar esta padronização no momento de registrar tais sinais é de extrema importância pois garante total possibilidade de uma comparação direta entre as amostras e suas escalas (tamanho do sinal em relação ao número de pixels da imagem) são as mesmas.

O conjunto apresenta 200 imagens com resolução de 768x560 pixels. Sendo estas imagens, 80 nevos comuns, 80 nevos atípicos e 40 melanomas. Cada imagem também contém uma máscara de segmentação, isto é, uma imagem binarizada que contém a região de interesse. Este *dataset* também inclui anotações médicas, chamadas de variáveis semânticas em Zhou S. K. and Rueckert e Fichtinger (2020), contendo Diagnóstico Clínico e critérios dermatológicos discutidos na Seção 2.1.3 (rede de pigmentos, área amorfia, pontos ou glóbulos e estrias).

A Figura 50 mostra um exemplo de lesão analisada em escala colorida, cinza e sua respectiva máscara.

Figura 50 – Exemplo de lesão e máscara presente no conjunto de dados



Fonte: Autor.

A Tabela 5 mostra quais variáveis foram fornecidas pelo conjunto de dados. Ressalva-se ainda que tais atributos são de extrema importância, pois a presença ou ausência destas podem indicar a ocorrência de câncer de pele, que no caso deste *dataset*, seria o melanoma.⁴

Tabela 5 – Variáveis semânticas do conjunto de dados PH2

Critério	Variáveis Semântica
Diagnóstico Clínico	0 - Nevo Comum
	1 - Nevo Atípico
	2 - Melanoma
Assimetria	0 - Totalmente simétrico
	1 - Assimétrico em 1 eixo
	2 - Totalmente Assimétrico
Rede de Pigmentos	AT - Atípico
	T - Típico
Pontos e Glóbulos	A - Ausente
	AT - Atípico
	T - Típico
Estrias	A - Ausente
	P - Presente
Área amorfa	A - Ausente
	P - Presente
Veia Azuladas	A - Ausente
	P - Presente
Cor	1 - Branco
	2 - Vermelho
	3 - Marrom claro
	4 - Marrom escuro
	5 - Cinza-Azulado
	6 - Preto

Fonte: ([MENDONÇA et al., 2013](#)).

Já conjunto de dados HAM10000 é de imagens obtidas pelo ViDIR Group (Departamento de Dermatologica da Universidade Médica de Viena) e a Faculdade de Medicina da Universidade de Queensland de Herston), ambas na Áustria ([TSCHANDL, 2018](#)). As imagens foram coletas de diferentes populações e devido ao número grande de pacientes, utilizou-se de mecanismos semi-automáticos para limpar e melhorar estas imagens ([TSCHANDL, 2018](#)).

O conjunto final apresenta 10015 contendo 7 classes. Destas 10015 imagens, são 1099 lesões tipo ceratose benigna, 115 dermatofibroma, 6705 nevos melanocíticos, 142 lesões vasculares, 327 carcinoma espinocelular, 514 carcinoma basocelular e 1113 melanoma. Todas imagens apresentam dimensão de 450x600 pixels. Cada imagem contém uma máscara

de segmentação que será utilizada para delimitar a região de interesse.

O conjunto HAM1000 também traz informações sobre os pacientes como gênero, idade e localização do sinal, porém não sobre anotações médicas semelhantes ao PH2. A Tabela 6 mostra as características da população do conjunto de dados.

Tabela 6 – Variáveis do conjunto de dados HAM10000

Critério	Variáveis
Gênero	Masculino
	Feminino
	Não - registrado
Idade	Valor numérico inteiro
Localização	Costas
	Extremidade inferior
	Tronco
	Extremidade inferior
	Abdômen
	Face
	Peito
	Pé
	Desconhecido
	Pescoço
	Escalpo
	Mãos
	Orelha
	Genitálias
	Acral

Fonte: ([TSCHANDL, 2018](#)).

Ressalva-se que tais atributos não apresentam o mesmo peso e relevância comparados aos da Tabela 5. Entretanto, ainda assim, contribuem para a análise do problema e são utilizados no treinamento dos modelos.

Entender os atributos e as informações fornecidas nos conjuntos de dados é o primeiro passo para compreender o problema. Segundo [Faceli et al. \(2011\)](#), analisar estas variáveis e utilizá-las na criação do modelo é eficaz e pode trazer resultados maiores, uma vez que grandezas categóricas e lineares completam-se.

É importante a utilização de máscaras para caracterização destas imagens, visto que a parte essencial para a resolução do problema é uma fração da imagem total fornecida. Avaliar as figuras sem utilizar suas máscaras iria corromper a análise pois estaria utilizando a coloração do fundo (pele do paciente), que não é uma característica decisiva para a presença de câncer.

Utilizando estes conjuntos de dados, o presente trabalho buscou analisar a eficiência de cada modelo de resolver tanto o problema binário (presença de Câncer) quanto o problema multiclasse (a qual classe a imagem pertence).

3.3 Extração das características

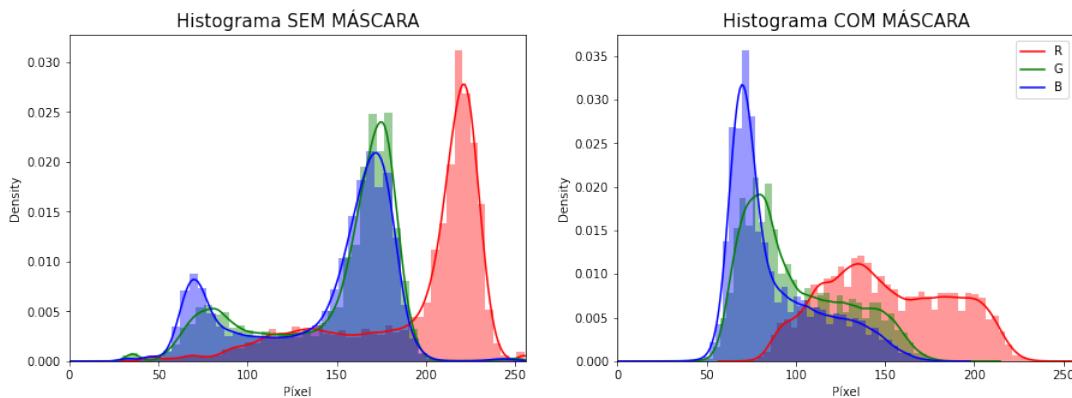
A extração dos dados, empregados no desenvolvimento deste trabalho, foi inspirada na metodologia usada por diversos autores como [Botelho \(2007\)](#), [Barros, Morais e Fernandes \(2017\)](#) e [Ganzeli et al. \(2011\)](#). A ideia central por trás destes trabalhos é a geração de dados estatísticos a respeito da lesão a partir de suas imagens e máscaras. Utilizam-se atributos qualitativos que possam descrever os parâmetros da regra ABCDE citados na Seção 2.1 ([BARROS; MORAIS; FERNANDES, 2017](#)).

Durante esta etapa, utilizou-se do módulo computacional criado para obter dois tipos principais de variáveis quantitativas: Coloração e Morfológica. Sendo que 212 variáveis são de Coloração e 19 Morfológicas. As Seções 3.3.1 e 3.3.2 explicam o mecanismo de obtenção de todas as 231 variáveis extraídas das imagens.

Em resumo, para calcular as variáveis de coloração, utilizou-se o conceito de histogramas. Segundo [Burger e Burge \(2016\)](#), um histograma é a distribuição de frequência de intensidade que aparece em uma imagem. A partir destas, são obtidas outras informações estatísticas importantes.

É feito a ressalva, como ilustrada pela Figura 51, que para ter um histograma que realmente represente o problema ou a qualidade que se deseja medir na imagem, deve-se utilizar a máscara para delimitar tais regiões de interesse.

Figura 51 – Diferença de histogramas ao utilizar ou não as máscaras para delimitar a região de interesse



Fonte: Autor.

A partir deste conjunto de valores, é possível extrair parâmetros estatísticos comuns

que podem ser provenientes destes histogramas (BURGER; BURGE, 2016). Para esta monografia, buscou-se por parâmetros estatísticos que pudessem descrever o conjunto analisado em sua totalidade.

Para o cálculo destas propriedades utilizaram-se medidas destacadas nos livros de Devore (2016) e Faceli et al. (2011).

As primeiras medidas estatísticas analisadas são de localidade. Supondo então que de um histograma, obtém-se n valores numéricos: $\mathbf{x}^j = (x_1, x_2, \dots, x_n)$ e x_i um valor dentro de \mathbf{x}^j , defini-se a média, ou valor médio como (FACELI et al., 2011; DEVORE, 2016):

$$\bar{x}^j = \frac{1}{n} \sum_{i=1}^n x_i \quad (3.1)$$

Também é utilizado a moda, que por definição é o valor de x_i que mais aparece em \mathbf{x}^j (DEVORE, 2016). Outras medidas muito importantes nestas análises estatísticas são mediana e o quartis.

A mediana é então definida ao ordenar a amostra de maneira crescente e aplicar a seguinte relação matemática (FACELI et al., 2011):

$$\text{mediana}(\mathbf{x}^j) = \begin{cases} \frac{1}{2}(x_r + x_{r+1}) & \text{se } n \text{ for par } (n = 2r) \\ x_r & \text{se } n \text{ for ímpar } (n = 2r + 1) \end{cases} \quad (3.2)$$

Já os quartis são semelhantes à mediana, pois também utilizam o ordenamento crescente dos dados (FACELI et al., 2011). Os valores são ordenados em quartos e assim, é possível obter medidas desta sequência. Por exemplo, o 1º Quartil apresenta o valor da amostra que apresenta 25% dos dados abaixo dele (FACELI et al., 2011). Seguindo esta lógica, o 2º quartil é a mediana, 3º apresenta o valor que apresenta 75% dos dados abaixo deste. Outras duas informações simples e importantes acerca destes dados são os valores de máximo e mínimo da amostra.

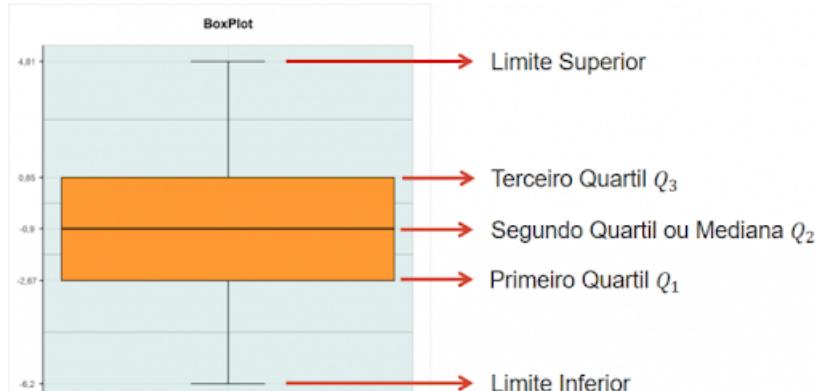
Outra maneira de visualizar estes atributos estatísticos mais facilmente é a utilização do gráfico *boxplot*. Onde os dados de máximo, mínimo, mediana e os quartis são expostos em uma única imagem.

Por fim, acrescentou-se ainda duas variações do cálculo de média para implementar as informações pertinentes ao trabalho. Sendo estas, a média geométrica, calculada por (DEVORE, 2016):

$$\bar{x}_{geo} = \sqrt[n]{x_1 \cdot x_2 \cdot x_3 \cdots x_n} \quad (3.3)$$

Já a média harmônica é calculada por (DEVORE, 2016):

$$\bar{x}_{harm} = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \frac{1}{x_3} + \dots + \frac{1}{x_n}} \quad (3.4)$$

Figura 52 – Exemplo de um *boxplot*

Fonte: Autor.

Para este projeto, devido à presença de porções escuras, utilizou-se uma transformação de valores para que as variáveis média harmônica e média geométrica apresentassem valores significativos aos problemas.

Outras medidas importantes para avaliar as propriedades estatísticas de uma amostra são as medidas de espalhamento. Estas, por sua vez, medem a dispersão ou espalhamento de um conjunto de valores (FACELI et al., 2011). Elas permitem inferir se os valores estão espalhados ou concentrados em torno de um valor (FACELI et al., 2011).

Para esta monografia, foram utilizados as medidas amplitude, variância, desvio padrão e desvio absoluto. Segundo (FACELI et al., 2011), a amplitude, também chamada de intervalo, é a medida mais simples de espalhamento que existe, sendo calculada por:

$$\text{amplitude}(\mathbf{x}^j) = \max_{i=1,\dots,n} (x_i) - \min_{i=1,\dots,n} (x_i) \quad (3.5)$$

Já a variância é dada pela expressão (FACELI et al., 2011):

$$\text{variância}(\mathbf{x}^j) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x}^j)^2 \quad (3.6)$$

e por definição, o desvio padrão é a raiz do valor da variância, isto é, desvio padrão(\mathbf{x}^j) = $\sqrt{\text{variância}(\mathbf{x}^j)}$.

Uma outra possibilidade de estimativa de dispersão robusta é o desvio absoluto (AAD do inglês), expresso por (FACELI et al., 2011):

$$\text{AAD}(\mathbf{x}^j) = \frac{1}{n-1} \sum_{i=1}^n |x_i - \bar{x}^j| \quad (3.7)$$

Todas estas medidas estatísticas foram utilizadas em alguma parte durante extração de características. Por fim, as seções 3.3.1 e 3.3.2 explicarão a metodologia envolvida para adquirir as variáveis utilizadas no treinamento deste projeto.

3.3.1 Variáveis de Coloração

Esta etapa da metodologia consiste na aplicação das transformações entre representações RGB e escalas Cinza, HSV, HSL, CIEE Lu'v' e CIEE L*a*b. A imagem em escala RGB foi fornecida pelo conjunto de dados.

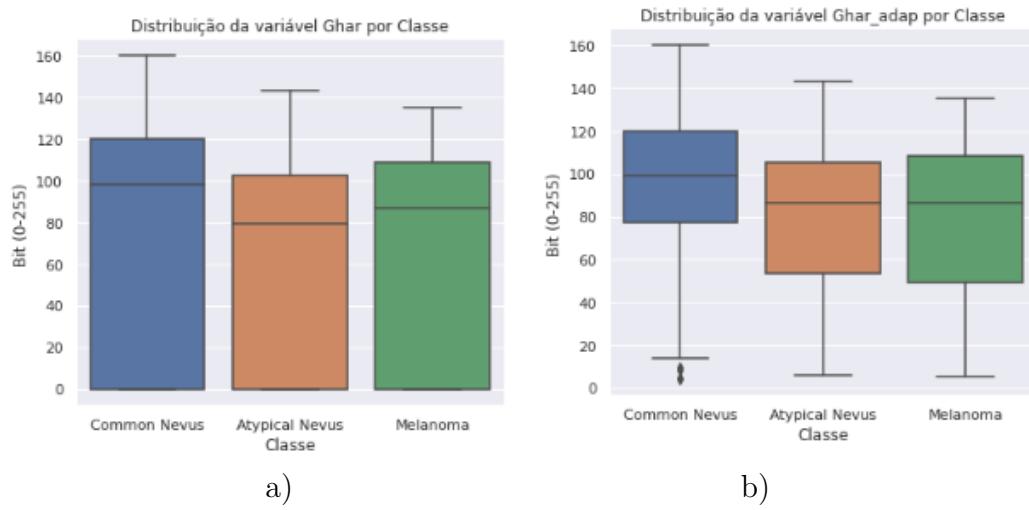
Para o desenvolvimento computacional desta etapa, utilizou-se a biblioteca [OPENCV \(2020\)](#). Esta por sua vez é uma biblioteca de código aberto desenvolvida para trabalhos de visão computacional e manipulação de imagens.

Utilizando as funções presentes da documentação [OPENCV \(2020\)](#), as imagens dos conjuntos foram copiadas e cada cópia passou por uma transformação para os formatos de escala Cinza, HSV, HSL, CIEE Lu'v' e CIEE L*a*b. Ressalta-se que todos estes sistemas de representação de cores apresentam 3 canais, com exceção do cinza, que apresenta somente 1.

Após a transformação, utilizou-se a máscara para encontrar os pares de pontos (x, y) que estavam dentro da parte binarizada da máscara. Estes pontos por sua vez tem sua representação nas imagens transformadas. O valor da intensidade nestes pontos, de cada imagem, em cada canal é então armazenado. Com estes valores obtidos são calculados a média, variância, desvio padrão, valor máximo, mínimo, mediana, 1º quartil, 3º quartil, moda, amplitude, desvio absoluto, média harmônica e média geométrica para cada canal.

Durante a execução desta etapa, observou que alguns canais apresentavam um único valor x_i nulo. Isto acarretava na média harmônica e geométrica valores tendendo a zero, como ilustra a Figura 53 parte a).

Figura 53 – Diferença na distribuição dos dados utilizando: a) sem a adaptação b) com adaptação



Fonte: Autor.

Para solucionar tal problemática, é feito uma adaptação atribuindo um valor

diferente do real quando é lido um valor nulo. No caso deste trabalho, quando é encontrado um valor nulo, atribui-se o valor de 0,001 para x_i . É importante ressaltar que esta adaptação não apresenta significativa interferência nas outras variáveis.

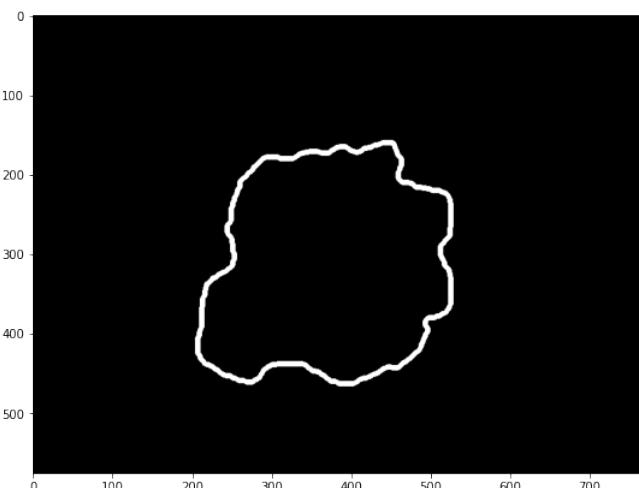
3.3.2 Variáveis Morfológicas

As variáveis morfológicas são aquelas relacionadas a geometria do problema. Estas por sua vez estão relacionadas ao Diâmetro, Borda e Assimetria da pinta. Dentro destes parâmetros, foram calculados primeiramente as seguintes grandezas básicas: raio, perímetro, área e assimetria nos eixos x e y.

A primeira etapa deste procedimento é encontrar o contorno da imagem binarizada. Este por sua vez é obtido utilizando uma função da biblioteca OpenCV, a *findContours()*.

Através da imagem binarizada, essa função retorna o seu contorno, isto é, o conjunto de pontos pertencentes a borda da imagem (BARROS; MORAIS; FERNANDES, 2017). A Figura 54 ilustra o funcionamento desta primeira etapa bem como a criação do contorno da imagem.

Figura 54 – Contorno da imagem binarizada

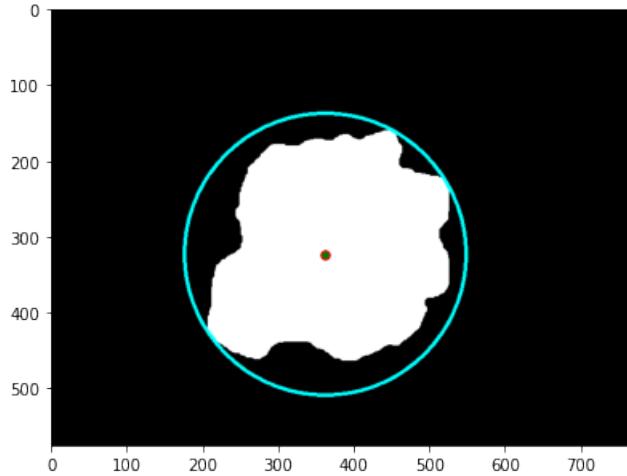


Fonte: Autor.

O cálculo do diâmetro é realizado através do contorno de uma circunferência a partir de um vetor de s de S pontos de contorno da imagem binarizada, no qual $s = [s_0, \dots, s_{S-1}]$. O método localiza um círculo de área mínima a partir do vetor s e o diâmetro da pinta terá o valor deste círculo. Para encontrar este círculo, foi utilizada a função da biblioteca OpenCV *minEnclosingCircle()* (BARROS; MORAIS; FERNANDES, 2017).

Essa função tem como saída o centro da circunferência e o raio do círculo. A Figura 55 mostra o resultado da obtenção do diâmetro na imagem binarizada.

Figura 55 – Circunferência em torno da mancha.



Fonte: Autor.

As variáveis área e perímetro também foram calculadas utilizando a biblioteca OpenCV e as funções *contourArea()* e *arcLength()* respectivamente. A entrada destas é contorno, calculada anteriormente e as suas saídas são valores numéricos em pixels correspondentes a área e perímetro da mancha.

O cálculo das simetrias em *X* e *Y* realizado utilizando os pontos pertencentes a máscara da imagem. Com os pontos da região branca (região de interesse) a assimetria de uma imagem *k* é representada pelas expressões matemáticas (BARROS; MORAIS; FERNANDES, 2017):

$$\text{Assimetria } X(k) = \left| \sum_{\substack{i=0 \\ i < c_x}}^{N-1} \sum_{j=0}^{M-1} I^k(i, j) - \sum_{\substack{i=0 \\ i \geq c_x}}^{N-1} \sum_{j=0}^{M-1} I^k(i, j) \right| \quad (3.8)$$

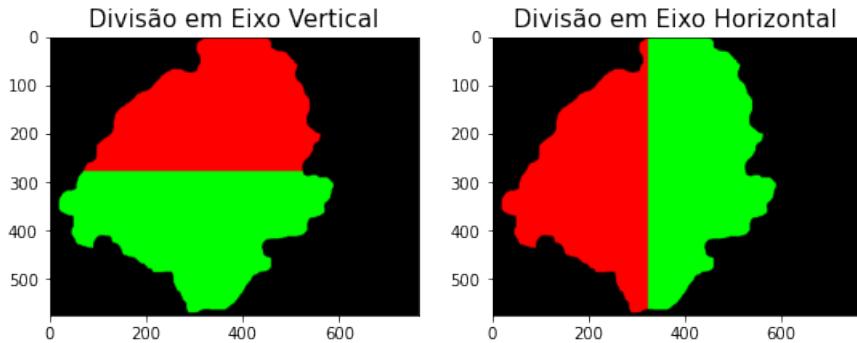
$$\text{Assimetria } Y(k) = \left| \sum_{\substack{i=0 \\ j < c_y}}^{N-1} \sum_{j=0}^{M-1} I^k(i, j) - \sum_{\substack{i=0 \\ j \geq c_y}}^{N-1} \sum_{j=0}^{M-1} I^k(i, j) \right| \quad (3.9)$$

onde c_x e c_y são os valores do centro de massa da imagem binarizada.

Em suma, o processo para obtenção destes dois parâmetros consiste em encontrar os pontos que definem o centro de massa da imagem e em seguida dividir esta imagem nos sentidos horizontais e verticais. A assimetria em *X* corresponde à diferença de áreas das áreas horizontais e a assimetria em *Y* das áreas verticais. A Figura 56 demonstra essa metodologia e as divisões feitas para execução dos cálculos de assimetria.

As bordas em uma imagem são o resultado de mudanças bruscas em alguma propriedade física ou espacial entre duas regiões adjacentes de superfícies iluminadas (GANZELI et al., 2011).

Figura 56 – Divisão do contorno.

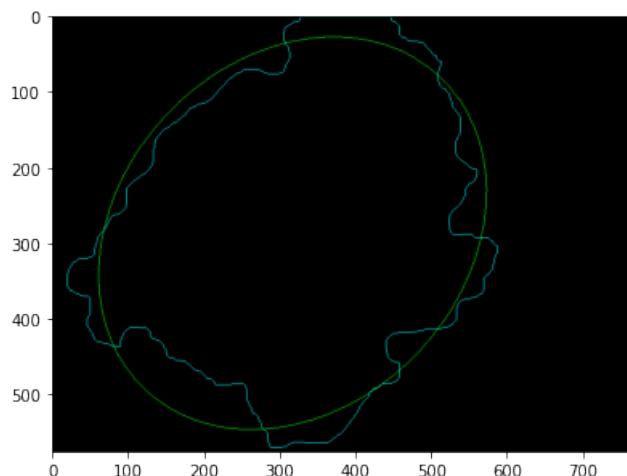


Fonte: Autor.

Após calcular o contorno, ilustrado pela Figura 54, foi calculado as coordenadas de cada ponto da borda, primeiro em coordenadas retangulares, e em seguida, transformando-as para coordenadas polares (GANZELI et al., 2011). Para cada ponto, calcula-se a distância do centro até a extremidade para cada ângulo. Em seguida, utilizando os pontos em coordenadas retangulares e a função *fitEllipse()*, criou-se uma elipse, cujo erro quadrático entre a borda da elipse e do contorno seja o menor possível (GANZELI et al., 2011).

A Figura 57 ilustra esse metodologia e a criação da elipse. É utilizado este método inventado por Ganzeli et al. (2011), pois uma elipse é a forma geométrica com bordas mais regulares que podem ser associadas ao problema deste trabalho.

Figura 57 – Elipse e contorno da pinta



Fonte: Autor.

Em seguida, pontos da elipse são transformados para coordenadas polares, obtendo-se uma distância para cada ângulo, similarmente ao processo mencionado anteriormente (GANZELI et al., 2011). A distância da elipse é subtraído do raio da borda. Assim,

elimina-se a variação da borda em função do formato da mancha, ficando-se apenas com as flutuações relativas à irregularidade da borda (GANZELI et al., 2011). Obtém-se um gráfico, como mostrado na Figura 58, que representa as saliências e reentrâncias da borda (GANZELI et al., 2011).

Figura 58 – Elipse e contorno da pinta



Fonte: (GANZELI et al., 2011).

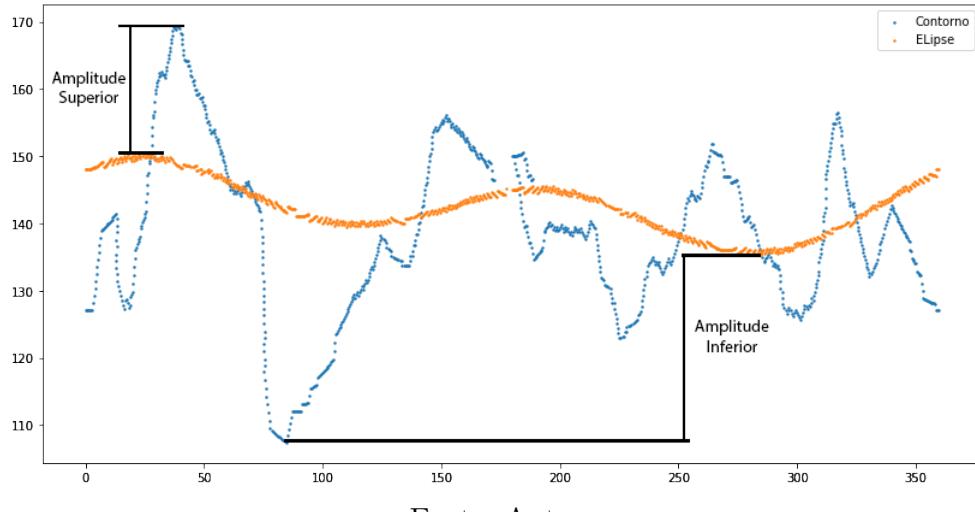
Segundo Ganzeli et al. (2011), vale ressaltar que esta metodologia para extrair estas informações não é afetado pela posição da mancha, nem sua direção.

A área dessa função de flutuação é extraída, e sua grandeza indicará o quanto irregular é a borda da mancha (GANZELI et al., 2011). Nesta monografia, esta variável se denomina diferença de áreas polares. Ademais, calcula-se a ondulação da borda tomando-se o maior e o menor picos, e fazendo-se a diferença entre eles e essa diferença é dividida pela maior distância da mancha, obtendo-se assim um número representativo da amplitude de variação da borda (GANZELI et al., 2011). A esta grandeza, dá-se o nome de amplitude absoluta.

No intuito de adquirir ainda mais variáveis que possam ser utilizadas para aprimorar as informações sobre o problema, este trabalho ainda idealizou outras variáveis. A primeira delas é a aplicação dos conceitos estatísticos levantados na Seção 3.3 para as distâncias encontradas anteriormente. Ou seja, calculou-se a média, variância, desvio padrão, valor máximo, mínimo, mediana, 1º quartil, 3º quartil, moda, amplitude, desvio absoluto, média harmônica e média geométrica de todas as distâncias encontradas.

Por fim, incrementou-se ainda mais duas variáveis denominadas neste trabalho como amplitude superior e amplitude inferior. Esta primeira por sua vez é a diferença entre os maiores valores de distância encontrados no contorno e na elipse. Já a amplitude inferior é a diferença entre os menores valores. A Figura 59 demonstra estas grandezas e como estas são calculadas.

Figura 59 – Cálculo da amplitude inferior e superior



Fonte: Autor.

Por fim, no contexto deste trabalho, elaborou-se um módulo em Python direcionado a realizar de maneira robusta este processo de extração utilizando o conjunto de imagens. Ademais, algumas funções auxiliares foram criadas para tratar e manipular estes dados para todas as etapas de desenvolvimento do projeto. Tais algoritmos estão disponíveis em <<https://github.com/thiagolopes97/TCC-ThiagoLopes>>.

3.4 Escolha dos modelos de ML

Como mostrado na Seção 2.4, modelos de classificação no aprendizado de máquina buscam generalizar uma função \hat{f} capaz de predizer uma classe um problema proposto. Para tal, os modelos tendem a ajustar uma função genérica aos dados apresentados.

Para esta monografia, todos os modelos foram submetidos ao aprendizado de máquina supervisionado. Para total comparação da qualidade destes modelos em generalizar o problema deste trabalho, realizou-se uma divisão de 4 amostras.

A amostra 1 apresenta todos os dados categóricos e rótulos presentes no conjunto PH2. A amostra 2 apresenta todas as características quantitativas calculadas na extração de características e os rótulos associados as imagens deste mesmo conjunto. A amostra 3 é uma combinação dos dados presentes nas amostra 1 e 2. Já a amostra 4 é composta pelos dados categóricos e atributos extraídos do conjunto de dados HAM10000. Todas amostras foram submetidas ao mesmo procedimentos de preparação e teste, que serão explicados na Seção 3.6.

Os algoritmos analisados nesta monografia são: Regressão Logística; *Naive Bayes*(Bernoulli e Gaussian); Árvore de Decisões; Floresta Aleatória; Árvores Extremamente Aleatória; KNN (*k-Nearest Neighbors*); RNN (*Radius Nearest Neighbors*); *Adapta-*

tive Boosting; Gradient Boosting; SVM Support Vector Machine (Linear, Polinomial, RBF, Sigmoidal) e Rede Neural (1,2 e 3 camadas ocultas).

As implementações destes algoritmos foram feitas utilizando as bibliotecas Pandas, Numpy e Scikit-Learn. Esta primeira que é utilizada para manipular e analisar conjunto de dados. O Numpy que é utilizado na computação científica, principalmente para cálculos matriciais multidimensionais e funções matemáticas prontas. Já o Scikit-learn é uma biblioteca de código aberto desenvolvida para aprendizado de máquina no Python, onde consta diversas funções já prontas que executam os modelos.

3.5 Obtenção dos hiperparâmetros

Treinar um determinado modelo de ML não é uma tarefa fácil. Como já exposto por [Faceli et al. \(2011\)](#) e [Géron \(2019\)](#), o treinamento destes modelos sem a adequação necessária pode levar ao sobreajuste ou subajuste.

Para evitar esta situação, é comum segmentar a base de dados em um conjunto de treinamento/ validação cruzadas e outro de teste. Segundo [Géron \(2019\)](#), utilizar 80% do conjunto total de amostra é um bom valor para treinar o modelo.

Neste trabalho, os dados envolvidos no processo de treinamento também correspondem a 80% da base de dados, assim, a maior parte dos dados é empregada para treinamento do modelo e aplicação da validação cruzada. Já o conjunto de teste é usado para avaliar o resultado do treinamento.

Outra maneira de evitar problemas na hora de treinar um modelo, é realizar um controle dos hiperparâmetros de cada algoritmos de ML, mencionada na Seção 2.4. Já o processo de validação cruzada é usado para determinação da melhor combinação de hiperparâmetros para cada modelo utilizado. A Figura 26 esquematiza o funcionamento do processo de validação cruzada empregado pela biblioteca ([SCIKIT-LEARN, 2020a](#)), utilizada nesta metodologia.

A Tabela 7 apresenta os valores de hiperparâmetros testados neste trabalho para cada modelo analisado

É feito a ressalva de que todos os dados foram separados em treinamento e teste, sendo utilizado somente os dados de treinamento para obtenção dos hiperparâmetros através da validação cruzada. O processo de validação cruzada realizou o método *k-fold cross validation* para buscar as melhores combinações dos hiperparâmetros expostos na Tabela 7.

No caso desta monografia, utilizou-se 5 partições iguais neste processo de validação cruzada. Assim, no processo de treinamento 4 partes são usadas para treinar o modelo e 1 parte para avaliar seu desempenho. Como conjunto foi dividido em 5 partes, o processo

Tabela 7 – Hiperparâmetros e seus valores analisados em cada modelo de ML

Modelo	Hiperparâmetros	Valores
Regressão Logística	C Solver	1,5,10,50,100,200,250,300,500 'newton-cg', 'lbfgs', 'liblinear'
NBG	Variável de suavização	$10^{-9}, 10^{-10}, 10^{-5}, 10^{-3}, 10^{-15}$
NBB	α Limiar para binarização	1, 1.5, 0.5, 2, 0.05 0, 0.5, 1, 20, 50
Árvore de Decisão	Função de custo Escolha do atributo Profundidade	'gini', 'entropy' 'best', 'random', 35, 40, 45, 50, 55, 60, 65, 70, 350, 400, 450, 500, 1000 0,5,10,12,14,15
	n^o mínimo de amostras para dividir um nó n^o mínimo de amostras em um nó	0, 5, 10, 25, 50, 100, 200
Floresta Aleatória	Função de custo Profundidade Número de estimadores	'gini', 'entropy' 5, 10, 20, 40, 50, 70, 80, 350, 400, 450, 500, 1000 10, 25, 50, 100, 200
Extra Tree	Função de custo Profundidade Número de estimadores	'gini', 'entropy' 5, 10, 20, 40, 50, 70, 80, 350, 400, 450, 500, 1000 10, 25, 50, 100, 200
KNN	k-vizinhos Cálculo da distância	1,5,10,20,30,50, 100,200,300,1000,1500 'uniform', 'distance'
RNN	Raio Cálculo da distância	0.5, 1, 1.5, 2, 2.5, 10000, 250000, 50000, 55000, 75000,100000 'uniform', 'distance'
AdaBoost	Número de estimadores Taxa de aprendizagem	10, 50, 100, 200, 300, 400 0.5, 1, 1.5, 2
GradientBoost	Número de estimadores Taxa de aprendizagem	10, 50, 100, 200, 300, 400 0.5, 1, 1.5, 2
SVM - Linear	C ϵ - sensitive	1,2,5,10,100,200, 300, 400,500 0.5,0.8,1,2,5,8,10
SVM - Sigmoid	C ϵ - sensitive	1,2,5,10,100,200, 300, 400,500 0.5,0.8,1,2,5,8,10
SVM - RBF	C ϵ - sensitive	1,2,5,10,100,200, 300, 400,500 0.5,0.8,1,2,5,8,10
SVM - Polinomial	Grau C ϵ - sensitive	1,2,3,4,5,10,20,30,50 1,2,5,10,100,200, 300, 400,500 0.5,0.8,1,2,5,8,10
RNA - 1 Camada	Neurônios da camada oculta Função de ativação	(50),(100),(150),(200),(300), (500) 'identity', 'logistic', 'tanh', 'relu'
RNA - 2 Camada	Neurônios da camada oculta Função de ativação	(50,50),(100,50),(100,100), (100,150),(300,300),(500,300) 'identity', 'logistic', 'tanh', 'relu'
RNA - 3 Camada	Neurônios da camada oculta Função de ativação	(50,50,50),(100,100,50),(200,150,50),(100,100,200), (300,300,300),(500,500,500),(500,300,100) 'identity', 'logistic', 'tanh', 'relu'

Fonte: Autor.

de validação cruzada repete o procedimento mais vezes até testar todas as combinações de hiperparâmetros possíveis. Ao fim da execução deste processo, é fornecido a melhor combinação de hiperparâmetros encontrado pelo algoritmo, isto é, os hiperparâmetros que obtiveram melhor desempenho.

Para tal procedimento, foi criado uma função auxiliar para testar todos os hiperparâmetros indicados na Tabela 7. Em resumo, a função faz com que todos eles sejam submetidos ao processo de validação cruzada e todas as combinações possíveis entre os hiperparâmetros selecionados sejam testadas. Em seguida, após realizar os testes de todas as possibilidades de combinação perante determinado modelo, esta função indica os melhores hiperparâmetros com base métrica B_S mencionada na Seção 2.4.3.

Devido ao grande número de combinações possíveis e cada método exige um processamento e tempo distinto, realizou-se todos os treinamentos utilizando uma tolerância de convergência de 0,001 e 10000 iterações. A utilização destes valores se justificam pois

em testes preliminares, certos algoritmos apresentaram longos tempos de convergência. Logo, ao utilizar a validação cruzada, treinando o modelo para cada combinação, o tempo de execução para alguns modelos ultrapassavam mais de 3 horas para somente um teste.

3.6 Treinamento dos modelos

A Figura 60 resume os passos tomados na etapa de treinamento. Toda a implementação computacional está exposta em <<https://github.com/thiagolopes97/TCC-ThiagoLopes>>.

Figura 60 – Resumo da metodologia utilizada na etapa de treinamento dos modelos



Fonte: Autor.

A primeira etapa antes de qualquer treinamento de um modelo é a preparação dos dados. Neste projeto, estes estão separados em dados categóricos (semânticos) e numéricos.

Foi realizado a transformação dos dados categóricos para valores que a máquina ou algoritmo consigam processar. Segundo [Faceli et al. \(2011\)](#), estes atributos podem se tornar valores discretos ou em formato binário. Utilizar a transformação para variáveis discretas pode induzir ao erro físico de que tais características sejam dependentes ou apresentam uma ordem de importância ([FACELI et al., 2011](#)). Por exemplo, ao analisar a Tabela 5, com relação ao Critério Pontos e Glóbulos, atribuir valores de 0, 1 e 2 para suas variáveis semânticas fornece um ordem de importância sobre elas. Para evitar isto, utilizou-se a transformação booleana, exemplificado pela Figura 61. Ressalva-se ainda que esta técnica foi aplicada em todos as variáveis categóricas fornecidas nos conjuntos PH2 e HAM10000 mencionadas na Seção 3.2.

Outro fator relevante para o treinamento de um modelo ML é a escala da variável utilizada, isto é, a ordem de grandeza e a forma como esta está exposta. Os valores sem alteração, isto é, os valores originais refletem a ordem destas características ([FACELI et al., 2011](#)). O problema de utilizar estes dados sem um pré-processamento prévio é que alguns modelos estão sujeitos a variação de resultado, tanto para pior quanto melhor. Para modelos como SVM ou RNA que utilizam pesos para cada valor de entrada, variáveis que apresentam uma ordem de grandeza muito elevada podem inviésar o sistema.

Para este projeto, foi utilizado a normalização dos dados, como sugerido por ([FACELI et al., 2011](#)). Para esta tarefa usou-se a biblioteca ([SCIKIT-LEARN, 2020i](#)) para

Figura 61 – Variáveis categóricas do conjunto PH2: a) antes da transformação boleana; b) após transformação boleana

	img_id	asymmetry	pigment_net	dots_glob	streaks	regressionA	blue_whit
0	IMD003	Simétrico		Típico	Ausente	Ausente	Ausente
1	IMD009	Simétrico		Típico	Ausente	Ausente	Ausente
2	IMD016	Simétrico		Típico	Típico	Ausente	Ausente
3	IMD022	Simétrico		Típico	Ausente	Ausente	Ausente

	Assim: S.	Assim: P.A.	Assim: T.A.	Pig: Típico	Pigm: Atípico	PG: Ausente	PG: Típico	PG: Atípico
0	0.0	1.0	0.0	0.0	1.0	0.0	1.0	0.0
1	0.0	1.0	0.0	0.0	1.0	0.0	1.0	0.0
2	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0
3	0.0	1.0	0.0	0.0	1.0	0.0	1.0	0.0
4	0.0	1.0	0.0	0.0	1.0	0.0	1.0	0.0

a)

b)

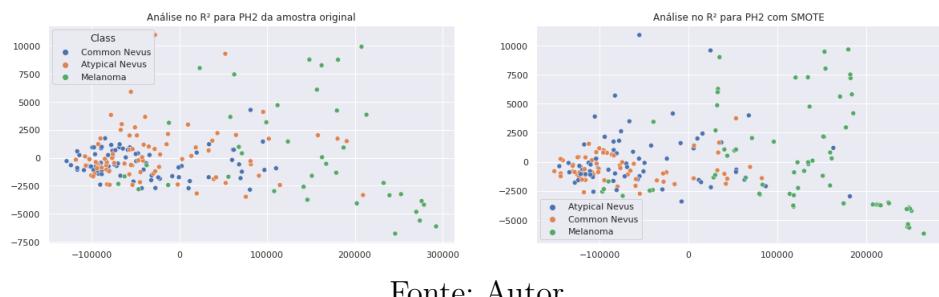
Fonte: Autor.

aplicar a normalização descrita pela Equação 2.4.

Outro problema também presente nesta monografia é a presença de dados desbalanceados. Isto ocorre neste projeto uma vez que casos de ocorrência de câncer de pele são menos comuns do que a ocorrência de sinais ou as outras doenças presentes nos conjuntos de dados.

No intuito de contornar este problema, foi utilizado a aplicação do SMOTE, mencionado na Seção 2.4, que consiste na criação de amostras sintéticas para balancear o conjunto de dados. A utilização desta técnica tem um peso importante para a generalização do modelo uma vez que este irá ter o mesmo número de instâncias para aprender a detectar as duas situações diferentes.

Figura 62 – Diferenças nas amostras com a utilização do SMOTE



Fonte: Autor.

Como ilustra a Figura 62, a aplicação desta técnica utiliza o conceito de distância

para introduzir os valores sintéticos. A introdução destas amostras não diminui a qualidade da predição das outras classes, mas sim fortalece a predição da classe minoritária.

No intuito de extrair o melhor de cada projeto e dos recursos computacionais disponíveis, foram avaliadas 5 possibilidades de preparação e treinamento utilizando as técnicas abordadas nesta Seção.

O primeiro teste é treinar o modelo utilizando somente o algoritmo e hiperparâmetros fornecidos pelo Scikit-Learn, sem utilizar a validação cruzada ou pré-processamento prévio. O segundo buscou-se realizar o treinamento utilizando a validação cruzada para obtenção dos melhores hiperparâmetros. O terceiro teste consiste na utilização tanto da normalização dos dados quanto da validação cruzada. O quarto é a aplicação da técnica SMOTE e da validação cruzada. Já o quinto teste é a combinação da normalização, SMOTE e validação cruzada para o treinamento do modelo. Nota-se que todos as 4 amostras foram submetidas a este procedimento e cada conjunto novo foi submetido ao teste binário e teste multiclasse.

Apesar da aplicação destes testes requerirem maior tempo, umas vez que é necessário treinar o modelo mais vezes, eles são justificáveis uma vez que alguns modelos podem ou não melhorar com sua aplicação. Como será visto no Capítulo 4, certos modelos apresentaram queda ou até mesmo uma grande melhora quando aplicado um destes 5 testes.

Em resumo, com os dados provenientes dos conjuntos PH2 e HAM10000 mais a extração de características, o modelo é treinado utilizando o conjunto de treinamento (80 % dos dados totais), fornecendo 4 amostras de treino. Cada amostra é então submetidas a 5 testes de processamento, gerando 5 novos conjuntos de treinamento cada uma. Estas por sua vez passam pelo processo de treinamento (com ou sem validação cruzada), conforme mencionado nesta Seção. Após o treinamento de cada uma, o conjunto de teste é usado para validar o treinamento e determinar a eficiência do modelo.

3.7 Teste e avaliação de desempenho dos modelos

Esta Seção aborda a metodologia utilizada para avaliar o desempenho obtido pelos modelos de ML com o processo de treinamento descrito até a seção anterior.

Após treinar os modelos utilizando a biblioteca Scikit-Learn, estes foram utilizados para realizar previsões com dados novos. Estes dados novos são o conjunto teste (20% dos dados totais), cujos modelos ainda não tiveram contato. Após realizar a predição, os resultados são então avaliados.

Como mencionado na Seção 2.4.3, os modelos foram avaliados na etapa de validação cruzada utilizando somente o B_S , onde os modelos com melhores valores teriam seus

hiperparâmetros selecionados. Já para a comparação de eficiência total dos modelos, outras métricas foram utilizadas como Pr, Ac, MMC e entre outras, sendo válido ressaltar a métrica de desempenho total definida neste trabalho por F_M . Esta por sua vez será utilizada para comparação entre métodos distintos.

Além destas métricas numéricas, também é utilizado métricas visuais como construção do gráfico ROC (Figura 48) e da matriz de confusão (Figura 47). A avaliação de desempenho então com estes dados de teste se torna de extrema importância, pois é ela que representa a real capacidade do modelo em prever determinada classe do problema.

Como resultado desta metodologia, para cada modelo de machine learning são apresentados os resultados de cada métrica mencionada na Seção 2.4.3 e o teste com maior F_M entre o mesmo tipo de modelo foi escolhido para ser o valor real.

4 Resultados

4.1 Apresentação dos dados dos conjuntos

Esta Seção destina-se a expor informações estatísticas importantes acerca de cada conjunto de dado utilizado nesta monografia. Bem como análises importantes das variáveis e classes, que justifiquem a metodologia utilizada neste projeto.

Como o foco desta monografia é avaliar o desempenho dos modelos ao utilizar estes dados fornecidos ou gerados, esta Seção é apenas uma breve explicação de quais e como os dados foram analisados.

Devido ao grande número de gráficos e variáveis analisadas, todos os resultados na íntegra estão dispostos em <<https://github.com/thiagolopes97/TCC-ThiagoLopes>>, porém alguns destes resultados serão mencionados nesta Seção devido a importância no projeto.

A Figura 63, apresenta um resumo das distribuições das classes do conjunto PH2 com relação aos dados semânticos fornecidos pelos médicos que efetuaram a coleta. Avaliar o comportamento destes dados com relação a classe, pode trazer informações importante acerca do problema analisado.

Como já abordado na Seção 2.1 e vista na Figura 63, as variáveis categóricas são de extrema importância e de impacto no conjunto de dados uma vez que estas estão ligadas a presença, ausência ou tipicidade de uma estrutura dermatológica.

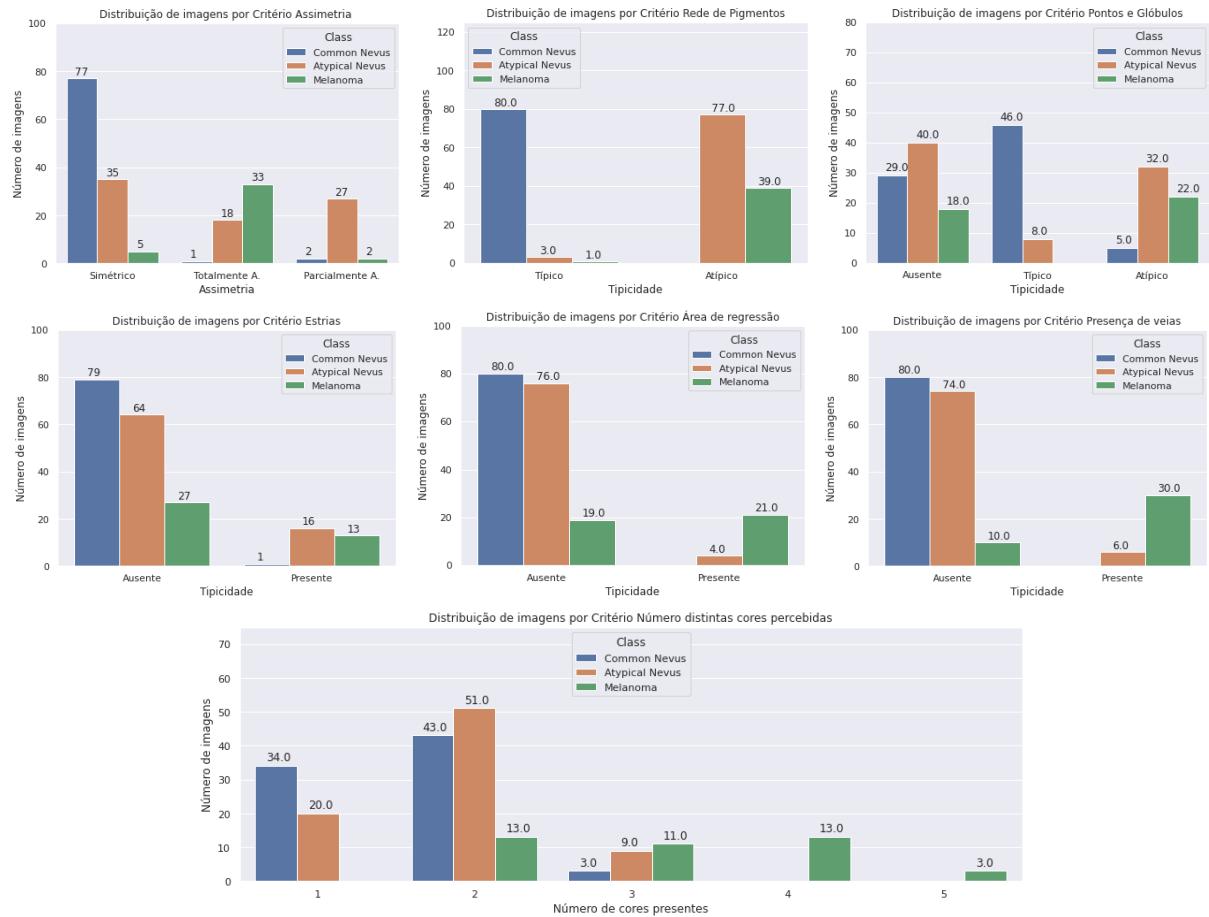
Analizando estes dados, ainda sem utilizar modelos de ML, já é possível observar que os critérios assimetria, rede de pigmentos, pontos e glóbulos bem como a presença de mais de uma cor na pinta são fortes indicativos de um câncer de pele. O que já dá indícios que estas serão categorias chaves para decisão de classe no modelo.

Com relação ao conjunto HAM10000, estes por sua vez apresentam características não tão relevantes para o problema. De certa forma, ainda assim é interessante avaliar estes dados para entender o comportamento dos mesmos perante a ocorrência ou não de câncer.

A Figura 64 mostra as ocorrências de imagens e suas classes de acordo com o gênero presente nos dados. A primeira vista, é possível perceber que a incidência de casos de pessoas do gênero masculino é um pouco maior do que o feminino. Ocorrência similar de acordo com o previsto pelo INCA (2020) e exposto nas Tabelas 1 e 2.

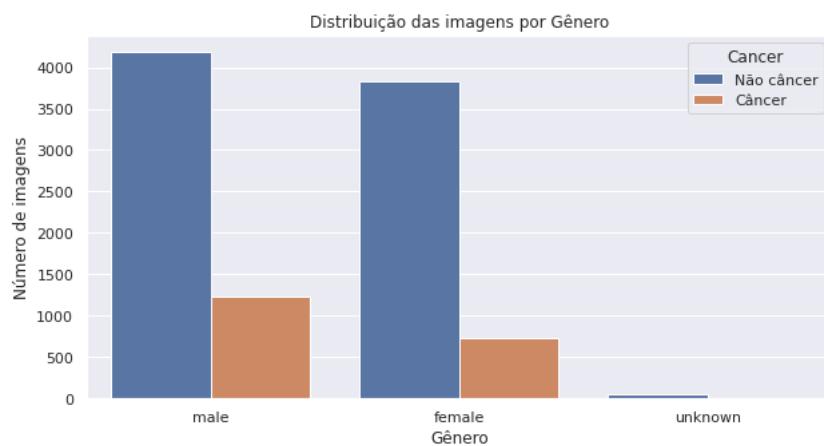
Já a distribuição deste mesmo conjunto em relação a localização do sinal pode ser visualizada na Figura 65. Nesta imagem é possível perceber que a maioria das ocorrências de câncer estão localizadas na face, costas, extremidade superior e inferior. É importante

Figura 63 – Distribuição das imagens do conjunto PH2 por Critério



Fonte: Autor.

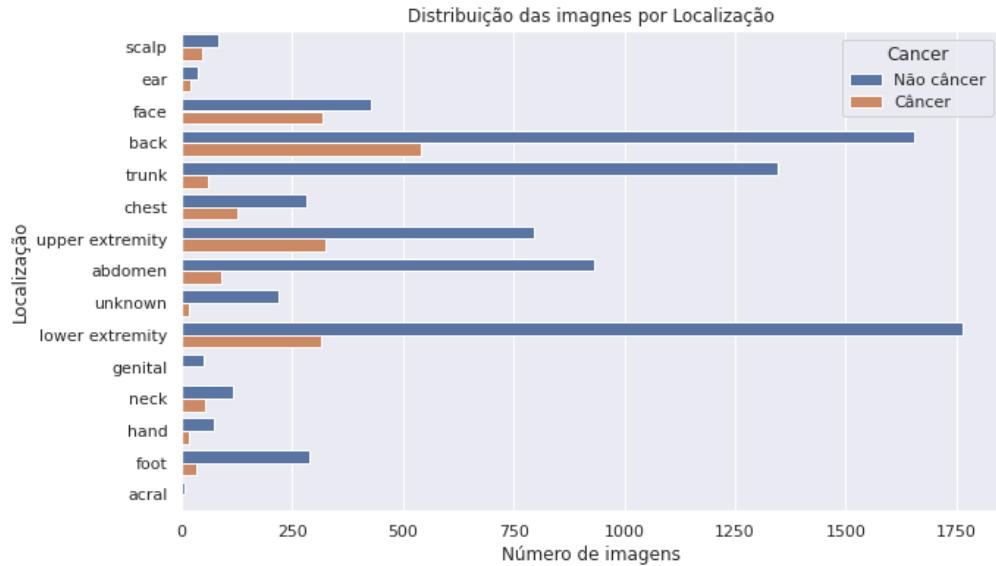
Figura 64 – Distribuição das imagens do conjunto HAM10000 por gênero



Fonte: Autor.

analisar este tipo de dado, pois do ponto de vista médico, regiões com maior incidência solar no corpo tendem a ter mais mutações genéticas que podem vir a se tornarem um câncer de pele.

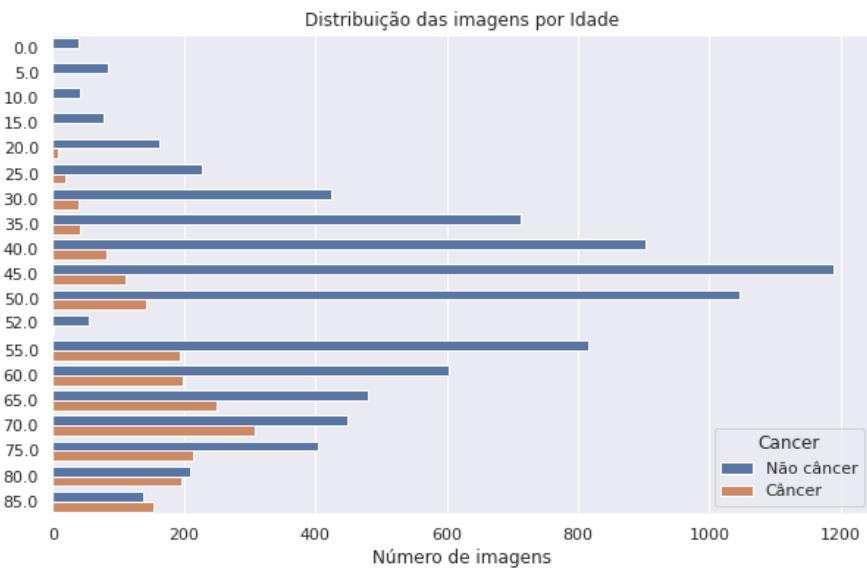
Figura 65 – Distribuição das imagens do conjunto HAM10000 por localização



Fonte: Autor.

Outra variável semântica presente no conjunto HAM10000 é a idade. A Figura 66 reforça o que discutido na Seção 2.1, que pessoas mais velhas tendem a ter mais sinais ou problemas na pele. Pela imagem, é possível perceber um grande aumento na ocorrência de câncer a partir dos 55 anos de idade.

Figura 66 – Distribuição das imagens do conjunto HAM10000 por idade

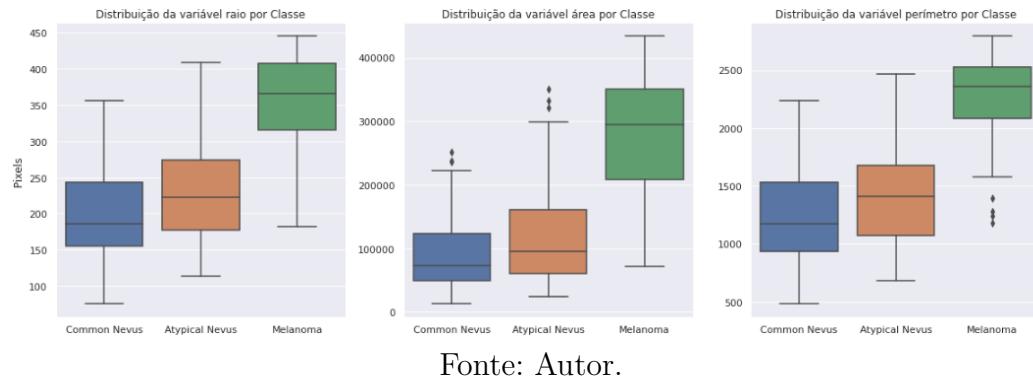


Fonte: Autor.

Já distribuição das grandezas qualitativas obtidas no processo de extração de características também tiveram uma análise exploratória. Os seus valores na integra para todas as variáveis estão disponíveis em <<https://github.com/thiagolopes97/TCC-ThiagoLopes>>,

bem como suas análises gráficas. Porém, é importante ressaltar algumas destas grandezas devido a importância destas no conjunto de dados.

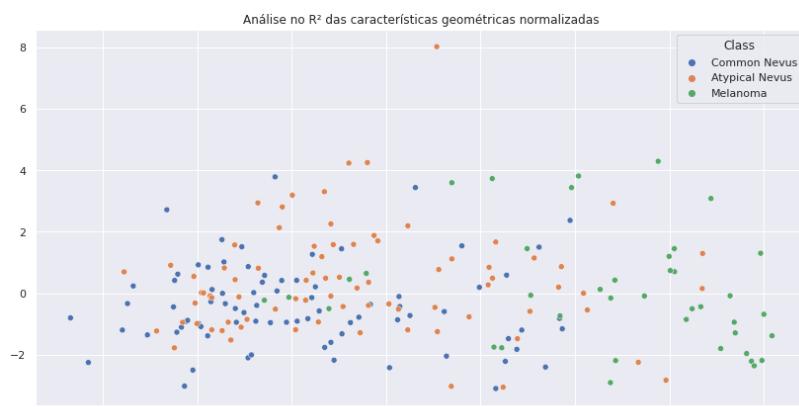
Figura 67 – Distribuição das variáveis morfológicas para o conjunto das imagens do conjunto PH2



Fonte: Autor.

A Figura 67 ilustra a distribuição de 3 características avaliadas: raio, área e perímetro obtidas no conjunto de dados PH2. Como é possível perceber, a classe melanoma apresenta a maioria dos seus dados em regiões bem definidas e acima das demais classes. Ou seja, a concentração dos dados estão normalmente em um região superior. Do ponto de vista médico, mencionado na Seção 2.1, isto faz sentido pois como afirma INCA (2020), pintas maiores tendem a ser câncer de pele. É importante salientar que estas regiões bem definidas são de extrema importância para os algoritmos conseguirem encontrar os padrões dos dados necessários para predizer aquela classe.

Figura 68 – Análise no espaço bidimensional das variáveis morfológicas normalizadas do conjunto PH2



Fonte: Autor.

Outra forma de visualizar estes dados, avaliando a sua combinação, é através da análise gráfica destes valores em um espaço bidimensional. Para isto, utilizou-se o algoritmo PCA (*Principal Component Analysis*) da biblioteca Scikit-learn por possuir eficiência na

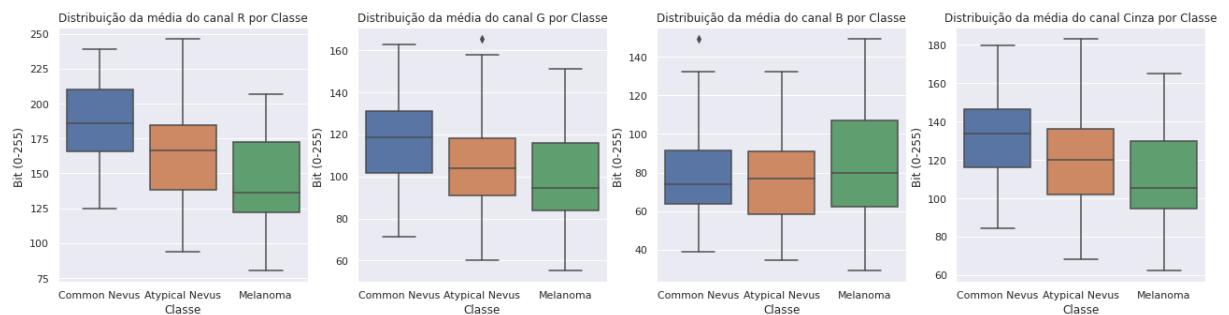
identificação de um hiperplano reduzido de dimensionalidades - usado na projeção - mais próximo aos dados ([GÉRON, 2019](#)).

Visualizar estes dados em duas dimensões auxilia entender o comportamento dos mesmos. A Figura 68, que ilustra esta visualização em duas dimensões, corrobora para os resultados obtidos para variáveis morfológicas. Isto é, a ocorrência de câncer é maior em pintas de tamanho elevado.

Para as variáveis de coloração, toma-se o exemplo da Figura 69, das médias dos canais RGB e Cinza. Analisando estas figuras é possível perceber que não existe faixa de valores bem definidas entre as classes, igual as variáveis morfológicas. Porém trabalhos como [Barros, Morais e Fernandes \(2017\)](#) mostram que utilizando apenas a média e desvio padrão destes canais em modelos mais robustos, como RNA por exemplo, é possível obter resultados satisfatórios.

É feito a ressalva de que esta situação se repetiu para todas as outras variáveis dos respectivos canais.

Figura 69 – Distribuição das variáveis dos Canais RGB e Cinza para o conjunto das imagens do conjunto PH2



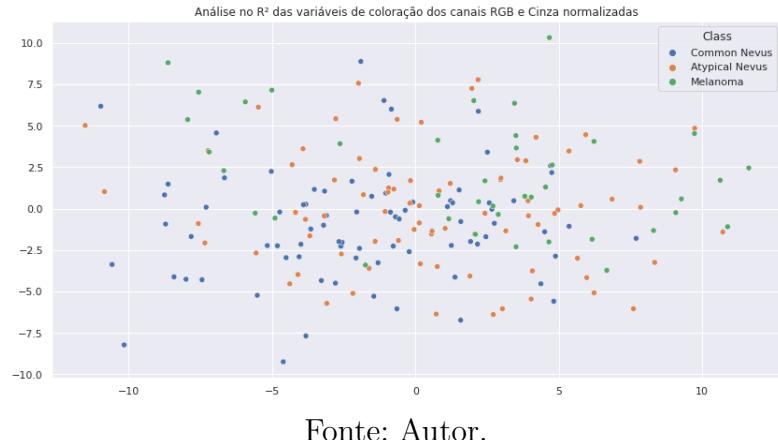
Fonte: Autor.

Analizando os dados utilizando a representação bidimensional (Figura 70) é possível perceber que os dados não estão bem representados por estas variáveis. Ou seja, a utilização destas variáveis para definir as classes não está bem visível.

Neste intuito, se baseando na metodologia de [Barata et al. \(2013\)](#), buscou-se obter novas variáveis que pudessem expressar melhor a coloração da imagem e apresentar características que pudessem distinguir as classes que foram utilizadas as outras representações de cor. Para isto, foi utilizado as outras 4 representações de cores: HSV, HSL, CIE Lu'v' e CIE L*a*b.

Ressalva-se que a utilização destas outras cores, apesar de todas representarem a mesma grandeza: coloração, não possuem o mesmo valor. Isto decorre do fato de que todas utilizam além de métricas distintas para medir as cores mas também variáveis físicas diferentes, como é o caso da *Hue* por exemplo.

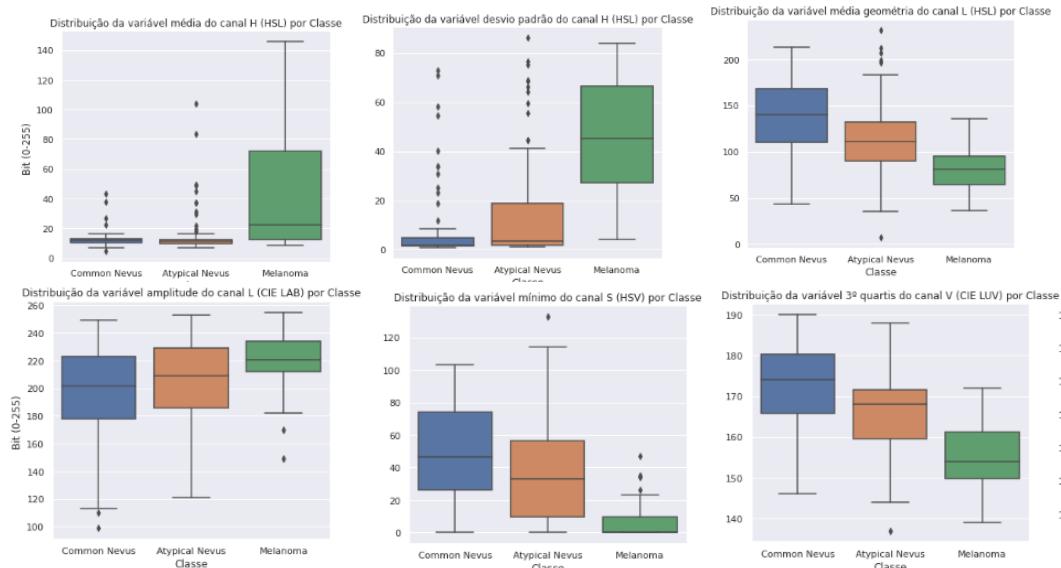
Figura 70 – Análise no espaço bidimensional das variáveis de coloração dos canais RGB e Cinza normalizadas do conjunto PH2



Fonte: Autor.

Neste novos atributos, é válido ressaltar algumas variáveis expressivas e suas representações bidimensionais.

Figura 71 – Distribuição de algumas variáveis das representações de cores HSV, HSL, CIE L*a*b e CIE Lu'v'



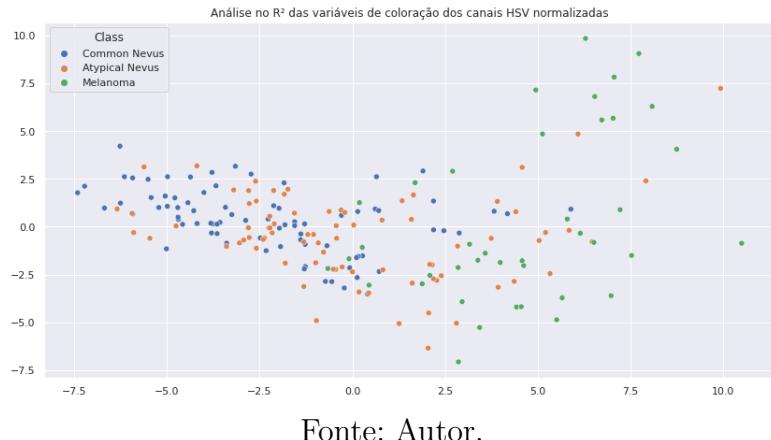
Fonte: Autor.

A Figura 71 mostra 6 variáveis analisadas de diversos canais diferentes. Nesta imagem é possível perceber que com algumas características e propriedades estatísticas é possível diferenciar classes mais facilmente. Isto fica claro ao analisar a média e desvio padrão do canal H (HSL), onde a classe melanoma apresenta regiões bem definidas. Entretanto, também existe outras variáveis como a amplitude ou valor mínimo dos canais L (CIE L*a*b) e S (HSV) apresentam uma região em comum.

As Figuras 72, 73, 74 e 75 apresentam as representações bidimensionais dos canais

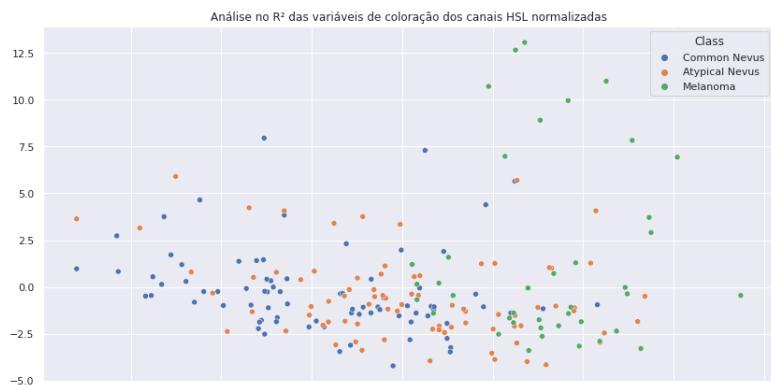
HSV, HSL, CIE Lu'v' e CIE L*a*b*. Na combinações das variáveis de cada canal é possível perceber que ocorre uma melhora na separação das classes, comparando estas imagens com a Figura 70. O que justifica a implementação destas variáveis e destes canais na metodologia deste trabalho.

Figura 72 – Análise no espaço bidimensional das variáveis de coloração dos canais HSV normalizadas do conjunto PH2



Fonte: Autor.

Figura 73 – Análise no espaço bidimensional das variáveis de coloração dos canais HSL normalizadas do conjunto PH2

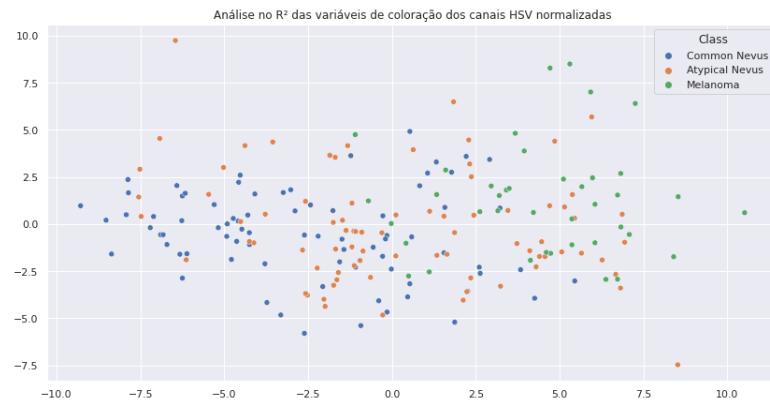


Fonte: Autor.

Já pela Figura 76, observa-se que é possível visualizar uma certa pré-disposição da classe melanoma mais a direita do gráfico. Poder perceber esta distribuição já fornece uma noção se os dados são ou não linearmente separáveis, se estão muito misturados ou distribuídos.

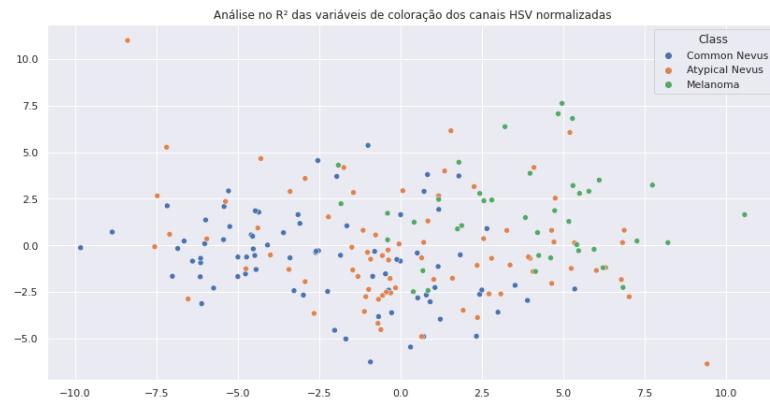
Por fim, o objetivo dos modelos treinados é reconhecer estes padrões através das formulações matemáticas e criar uma função generalizadora que permite predizer a classe da imagem. A implementação computacional desta análise exploratória, bem como os outros gráficos estão em (LOPES, 2020).

Figura 74 – Análise no espaço bidimensional das variáveis de coloração dos canais CIE Lu'v' normalizadas do conjunto PH2



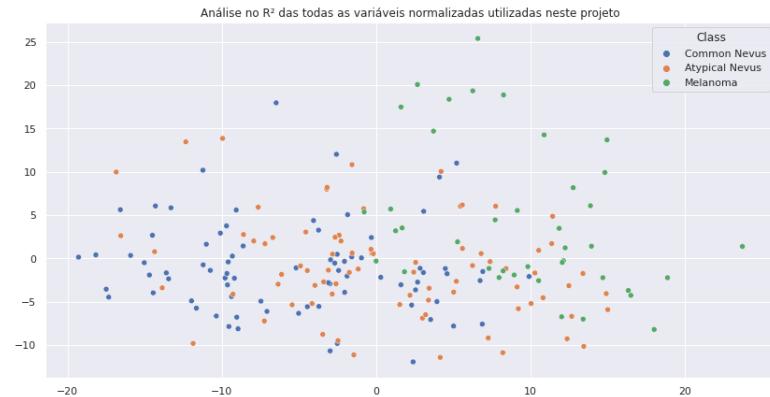
Fonte: Autor.

Figura 75 – Análise no espaço bidimensional das variáveis de coloração dos canais CIE L*a*b normalizadas do conjunto PH2



Fonte: Autor.

Figura 76 – Análise no espaço bidimensional de todas as variáveis utilizadas neste projeto



Fonte: Autor.

4.2 Avaliação de desempenho dos modelos

Esta Seção destina-se a avaliar o comportamento de cada modelo para amostra e teste utilizado. Expor os resultados obtidos para cada modelo. No Anexo A estão dispostas as tabelas contendo os resultados de cada modelo, com o melhor resultado do modelo destacado em cinza. Sendo os melhores melhores discutidos mais afundo na Seção seguinte. Os resultados gráficos e matrizes de confusão de todos os modelos estão disponíveis em <<https://github.com/thiagolopes97/TCC-ThiagoLopes>>.

4.2.1 Regressão Logística

Para o modelo de regressão logística, avaliou-se o seu treinamento utilizando a validação cruzada e a regularização de seus hiperparâmetros.

Para a amostras 1, que só continha variáveis categóricas, este modelo apresentou grande desempenho em todos os tipos de tratamento. No teste binário, o melhor modelo treinado conseguiu alcançar uma acurácia de 0,9250 e F_M de 0,9011. Já para o teste multiclasse, o modelo obteve , em seu maior desempenho, acurácia de 0,9250 e F_M de 0,9421. Pelos dados das Tabelas 10 e 11, nota-se que este modelo teve mais facilidade de ligar com o caso multiclasse desta amostra 1.

Já o mesmo modelo para amostra 2, contendo somente as variáveis qualitativas advinda da metodologia de extração de características teve um desempenho diferente. Para o teste binário, o melhor modelo analisado obteve uma acurácia de 0,9000 e F_M de 0,8779. No teste multiclasse entretanto, o modelo teve uma queda de desempenho em todos os testes, quando comparado com o anterior. Neste caso, o melhor modelo apresentou uma acurácia de 0,6500 e F_M de 0,6956.

A amostra 3, contendo a junção das variáveis categóricas e qualitativas avaliadas por esse modelo oferecem uma forte combinação, melhorando o desempenho ainda mais. Durante o teste binário, o modelo alcançou uma acurácia de 0,9500 e F_M de 0,9282. No multiclasse, 0,900 de acurácia e F_m de 0,9170. O modelo treinado com esta amostra, também obteve suas métricas aumentadas no geral, na maioria dos tratamentos utilizados.

O modelo de regressão logística apresentou uma certa variação significativa de seus resultados de acordo com o tratamento utilizado. O tratamento 1, isto é, sem aplicação da validação cruzada ou pré-processamento, obtiveram resultados baixos.

Ao analisar a amostra 4 , estes resultados melhoraram e no teste binário, o melhor desempenho foi F_M de 0,7862 e acurácia de 0,8402. Já no multiclasse, os melhores desempenhos obtiveram F_M de 0,7944 e acurácia de 0,8103.

Ao analisar o modelo de regressão logística, é possível perceber que resultados altos quando submetidos a dados categóricos (amostras 1 e 3). Em contra partida, o desempenho

para o mesmo modelo utilizando as amostras 2 e 4, foram baixas.

4.2.2 Gaussian e Bernoulli Naive Bayes

O modelo *Naive Bayes* parte do pressuposto de que as probabilidades das variáveis são independentes entre si. Isto não é totalmente verdade para o problema em questão, uma vez que variáveis como raio, área e perímetro estão correlacionadas. Entretanto, os modelos NBG e NBB obtiveram bom desempenho em algumas amostras e menor em outras.

Para o modelo treinado sob a amostra 1, o teste binário obteve acurácia de 0,9250 e F_M de 0,9077 para o modelo NBG e 0,9250 e F_M de 0,9017 para o NBB. Já o teste multiclasse resultou em métricas de desempenho para o NBG de 0,9250 de acurácia e F_M de 0,9492 enquanto o modelo NBB alcançou 0,8750 e F_M de 0,9017.

Ao analisar os melhores desempenhos do modelo para a amostra 2 teste binário, o NBG obteve 0,9000 e 0,8779 de acurácia e F_M , respectivamente. O modelo NBB nesta etapa conseguiu valores de 0,875 de acurácia e 0,8514 de F_M . Já os mesmos modelos para a análise multiclasse, obtiveram 0,6250 e 0,6714 de acurácia e F_M , respectivamente, para o NBG e 0,6750 e 0,7322 de acurácia e F_M , respectivamente, para o NBB.

Na amostra 3, contendo a junção das variáveis categóricas e qualitativas do conjunto PH2, os modelos obtiveram no teste binário: 0,9000 de acurácia e 0,8796 e F_M para o NBG e 0,9000 de acurácia e 0,8731 e F_M para o NBB. Quanto aos resultados dos modelos no multiclasse: 0,9000 de acurácia e 0,9127 e F_M para o NBG e 0,8750 de acurácia e 0,9020 e F_M para o NBB.

Para a amostra 4, o modelo NBG desempenhou em seu teste binário uma acurácia de 0,7139 e F_M de 0,6751. Neste mesmo teste, o modelo NBB alcançou um desempenho de 0,7718 e 0,6813 de acurácia e F_M , respectivamente. Para o teste multiclasse, alcançou-se um acurácia de 0,5217 e F_M 0,6962 com o NBG e acurácia de 0,5047 e F_M 0,6579.

Analizando o desempenho deste modelo individualmente, observa-se que o modelo NBG obteve um desempenho um pouco melhor do que o NBB. Ademais, ressalta-se que ambos modelos obtiveram resultados melhores nos testes binários em comparação com os testes multiclasse.

4.2.3 Árvore de Decisões

Primeiramente, este modelo obteve uma acurácia de 0,9000 e F_M de 0,8734 para o teste binário da amostra 1. Quanto o multiclasse, o mesmo modelo obteve 0,9250 e 0,9402, demonstrando uma melhora na separação com balanceamento das classes.

Já o desempenho do mesmo para a amostra 2 é de 0,9250 de acurácia e F_M de

0,9402 no teste binário. No outro caso, multiclasse, os valores de acurácia e F_M são 0,7500 e 0,7809 respectivamente. É notável neste caso um decaimento na capacidade do modelo em prever as mais de 2 classes utilizando somente variáveis qualitativas.

Durante a validação dos resultados para a amostra 3, o modelo de árvores de decisão obteve para o teste binário: 0,9250 de acurácia e 0,9035 de F_M . O teste multiclasse neste caso apresentou valores de 0,9000 de acurácia e 0,9157 de F_M . O modelo analisado, para a amostra 4, obteve 0,8457 de acurácia e 0,7302 de F_M para o teste binário e 0,7179 e 0,6650 para o multiclasse.

No âmbito geral, o modelo apresentou resultados semelhantes de desempenho independente do tratamento utilizado. Ademais, os resultados obtidos por ele são satisfatórios.

4.2.4 Floresta Aleatória

No modelo Floresta Aleatória (*Random Tree*), uma adaptação do modelo de Árvores de Decisões, obteve-se para o teste binário da amostra 1 uma acurácia de 0,9000 e F_M de 0,8g82. Enquanto para o caso multiclasse, obteve-se 0,9000 e 0,9249.

Já para o conjunto contendo somente as variáveis quantitativas do *dataset* PH2, o modelo alcançou uma acurácia de 0,9250 e F_M de 0,8964 no teste binário. No multiclasse, o modelo alcançou uma acurácia de 0,8250 e F_M de 0,8581.

Na amostra 3, o modelo obteve para o teste binário: 0,9250 de acurácia e 0,9035 de F_M . Já o teste multiclasse obteve 0,9000 de acurácia e 0,9157 de F_M .

No conjunto 4, utilizando os dados provenientes do conjunto HAM10000, este modelo desempenhou uma acurácia de 0,8802 e F_M de 0,8141. Já no caso multiclasse, o mesmo modelo alcançou acurácia de 0,7973 e F_M de 0,7559.

É interessante ressaltar que o desempenho deste modelo ultrapassou os resultados obtidos pelo floresta de decisão. Isto mostra que a combinações de modelos, isto é, o método *ensemble* é um artifício poderoso no aprendizado de máquina.

4.2.5 Árvores Extremamente Aleatória

O método de árvores extremamente aleatórias (*Extra Tree*), outra variante do método de árvores de decisões, apresentou para amostra 1 teste binário: acurácia de 0,9500 e F_M de 0,9307. Para o teste multiclasse: acurácia de 0,9250 e F_M de 0,9406.

Analizando os melhores desempenhos do modelo para a amostra 2 teste binário, obteve-se 0,0,8750 e 0,8318 de acurácia e F_M , respectivamente. Já o mesmo modelo para o teste multiclasse conseguiu valores de 0,9000 de acurácia e 0,8629 de F_M .

Para a amostra 3, análise binária, alcançou 0,9000 e 0,8175 de acurácia e F_M , respectivamente. No caso do multiclasse, 0,9250 e 0,9447 de acurácia e F_M , respectivamente.

O modelo submetido a amostra 4 conseguiu chegar ao desempenho de 0,8737 de acurácia e 0,7870 de F_M para o teste binário e 0,8038 de acurácia e 0,7416 de F_M para o teste multiclasse.

4.2.6 KNN e RNN

Os modelos baseados em distância KNN e RNN em geral apresentaram desempenhos piores se comparado aos outros métodos em alguns tratamentos, principalmente nas amostras 2, 3 e 4.

Para o modelo treinado sob a amostra 1, o teste binário obteve acurácia de 0,9000 e F_M de 0,8734 para o modelo KNN e 0,9000 e F_M de 0,8656 para o RNN. Já o teste multiclasse resultou em métricas de desempenho para o KNN de 0,9000 de acurácia e F_M de 0,9143 enquanto o modelo RNN alcançou 0,8750 e F_M de 0,8911.

Ao analisar os melhores desempenhos do modelo para a amostra 2 teste binário, o KNN obteve 0,9000 e 0,8175 de acurácia e F_M , respectivamente. O modelo RNN nesta etapa conseguiu valores de 0,8750 de acurácia e 0,8117 de F_M . Já os mesmos modelos para a análise multiclasse, obtiveram 0,6000 e 0,6846 de acurácia e F_M , respectivamente, para o KNN e 0,6000 e 0,6778 de acurácia e F_M , respectivamente, para o RNN.

Na amostra 3, criada a partir do conjunto PH2, os modelos obtiveram no teste binário: 0,9000 de acurácia e 0,8175 e F_M para o KNN e 0,9000 de acurácia e 0,8473 e F_M para o NBB. Quanto aos resultados dos modelos no multiclasse: 0,6250 de acurácia e 0,6769 e F_M para o KNN e 0,6000 de acurácia e 0,6743 e F_M para o RNN.

Para a amostra 4, o modelo KNN desempenhou em seu teste binário uma acurácia de 0,8422 e F_M de 0,7101. Neste mesmo teste, o modelo RNN alcançou um desempenho de 0,6875 e 0,6738 de acurácia e F_M , respectivamente. Para o teste multiclasse, alcançou-se um acurácia de 0,7449 e F_M 0,6422 com o KNN e acurácia de 0,5212 e F_M 0,5424 com o RNN.

4.2.7 *Adaptative Boosting*

Os métodos *boosting*, sendo o *Adaptative Boosting* um deles, é uma vertente do método *ensemble* que utiliza a combinação de diversos estimadores para criar um modelo em conjunto.

Para este modelo, na validação da amostra 1, obteve-se 0,9250 de acurácia e 0,8981 de F_M para o teste binário. No outro teste, para a mesma amostra, alcançou-se 0,8500 e 0,8512 de acurácia e F_M respectivamente.

Quanto a análise utilizando a amostra 2, os resultados foram: 0,9250 de acurácia e 0,8952 de F_M para o teste binário e 0,6750 de acurácia e 0,7270 de F_M .

Para a amostra 3, análise binária, alcançou 0,9500 e 0,9247 de acurácia e F_M , respectivamente. No caso do multiclasse, 0,9000 e 0,9126 de acurácia e F_M , respectivamente.

No que cerca os resultados obtidos para a amostra 4, este modelo obteve no teste binário, acurácia e F_M de 0,8662 e 0,7783, respectivamente. No caso multiclasse, 0,5642 de acurácia e 0,6158.

Nota-se com os estes dados, que o modelo teve uma queda de desempenho quando avaliado o problema multiclasse.

4.2.8 Gradient Boosting

O algoritmo de *Gradient Boosting* obteve uma acurácia de 0,9500 e F_M de 0,9390 para o teste binário da amostra 1. Analisando a mesma amostra, no caso multiclasse, o modelo alcançou 0,9250 de acurácia e F_M de 0,9382.

Quanto ao mesmo modelo para a amostra 2, seu desempenho foi de 0,8500 e 0,8067 de acurácia e F_M respectivamente para a análise binária. Para o desempenho no multiclasse, o modelo chegou a conseguir 0,6500 de acurácia e F_M 6948.

Analizando o resultado deste modelo para a amostra 3, as melhores métricas alcançadas para o teste binário foram 0,9000 para acurácia e 0,8602 para F_M . Os valores dessas métricas para o teste multiclasse foram 0,9000 e 0,9126.

Para a amostra 4, contendo as variáveis provenientes do conjunto HAM10000, o modelo obteve no teste binário 0,8702 de acurácia e 0,7875 de F_M . Para o multiclasse, 0,8048 e 0,7607 de acurácia e F_M respectivamente.

4.2.9 SVM

No processo de avaliação do modelo SVM, foram analisadas diferentes kernels utilizados, a saber: Linear, Polinomial, RBF e Sigmoidal. Então cada modelo com seu kernel em particular foi submetido a teste e seu desempenho foi analisado. A Tabela 8 mostra um resumo dos resultados obtidos para este modelo em cada kernel.

Observando a Tabela 8 e as tabelas do Anexo A, nota-se uma boa variação dos desempenhos dos modelos SVM dependendo do tratamento utilizado. Conforme SCIKIT-LEARN (2020j), os modelos SVM tendem a ser dependentes dos dados, por isso é recomendável padronizá-los antes de submetê-los a treinamento.

Outro fator importante a ser ressaltado é o desempenho menos dos kernel linear e sigmoidal para a amostra 4. Isto pode ser ocasionado devido a distribuição dos dados ou estes não serem linearmente separáveis, como foi comentado na Seção 2.4.2.7.

Tabela 8 – Resumo dos resultados obtidos para o modelo SVM

Kernel	Amostra	Teste	<i>AC</i>	<i>F_M</i>
Linear	1	Binário	0,9250	0,8861
	1	Multiclasse	0,9250	0,9353
	2	Binário	0,9500	0,9270
	2	Multiclasse	0,6750	0,7133
	3	Binário	0,9250	0,9041
	3	Multiclasse	0,9250	0,9419
	4	Binário	0,7369	0,6153
	4	Multiclasse	0,6370	0,6558
Polinomial	1	Binário	0,9500	0,9294
	1	Multiclasse	0,9500	0,9590
	2	Binário	0,9250	0,8880
	2	Multiclasse	0,7000	0,7454
	3	Binário	0,9250	0,8856
	3	Multiclasse	0,9250	0,9420
	4	Binário	0,8337	0,7739
	4	Multiclasse	0,7713	0,7411
RBF	1	Binário	0,9250	0,9004
	1	Multiclasse	0,9250	0,9438
	2	Binário	0,9000	0,8645
	2	Multiclasse	0,6750	0,7283
	3	Binário	0,9000	0,8614
	3	Multiclasse	0,7250	0,7929
	4	Binário	0,8662	0,7813
	4	Multiclasse	0,8153	0,7871
Sigmoidal	1	Binário	0,9000	0,8645
	1	Multiclasse	0,9250	0,9421
	2	Binário	0,8500	0,8040
	2	Multiclasse	0,6750	0,7283
	3	Binário	0,8750	0,8407
	3	Multiclasse	0,7500	0,7939
	4	Binário	0,6345	0,5622
	4	Multiclasse	0,5726	0,5623

Fonte: Autor.

4.2.10 Redes Neurais Artificiais

Para o modelo de Redes Neurais Artificiais, foram treinados e avaliados 3 possíveis formas de variação do MLP (*Multi Layer Perceptron*). Nesta monografia foram utilizadas uma RNA-MLP com 1, 2 e 3 camadas ocultas, avaliando o desempenho em cada caso. A Tabela 9 mostra um resumo dos resultados obtidos para este modelo em conforme o número de camadas ocultas.

Tabela 9 – Resumo dos resultados obtidos para os RNA-MLP

Camadas	Amostra	Teste	<i>AC</i>	<i>F_M</i>
1	1	Binário	0,9500	0,9343
	1	Multiclasse	0,9250	0,9355
	2	Binário	0,9250	0,8994
	2	Multiclasse	0,7000	0,7411
	3	Binário	0,9000	0,8693
	3	Multiclasse	0,7500	0,7774
	4	Binário	0,8842	0,8179
	4	Multiclasse	0,8084	0,7996
2	1	Binário	0,9250	0,8981
	1	Multiclasse	0,9250	0,9548
	2	Binário	0,9000	0,8704
	2	Multiclasse	0,6750	0,7252
	3	Binário	0,9000	0,8704
	3	Multiclasse	0,8000	0,8160
	4	Binário	0,8643	0,7959
	4	Multiclasse	0,8244	0,8160
3	1	Binário	0,9250	0,8873
	1	Multiclasse	0,9250	0,9541
	2	Binário	0,9000	0,8696
	2	Multiclasse	0,6750	0,7124
	3	Binário	0,9000	0,8681
	3	Multiclasse	0,7750	0,8272
	4	Binário	0,8907	0,8207
	4	Multiclasse	0,8224	0,8154

Fonte: Autor.

Avaliando os resultados da tabela acima, é possível perceber que o aumento dos números de camadas ocultas não implica necessariamente no aumento do desempenho do modelo.

4.3 Discussão geral dos modelos

Para esta Seção, na análise binária considera-se como classe positiva (1) a presença de câncer, independente da amostra analisada. No caso do teste multiclasse, para as amostras 1,2 e 3 é considerado os seguintes valores para cada classe: Melanoma (0), Nevo Comum (1) e Nevo Atípico (2). Já para a amostra 4 teste multiclasse, considera-se as classes: Nevos melanocíticos (0), Lesões ceratose benigna (1), Carcinoma basocelular (2), Melanoma (3), Carcinoma epidermoide (4), Lesões vasculares (5) e Dermatofibroma (6).

Com base nos resultados apresentados, nas seções anteriores deste capítulo, cada

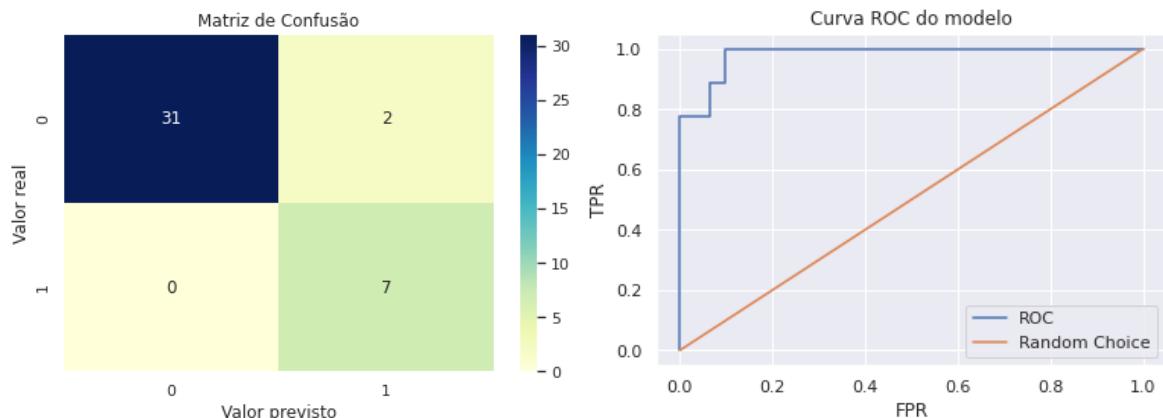
modelo foi avaliado perante uma amostra diferente. Estas por sua vez tem um intuito de avaliar o modelo na predição de câncer de pele utilizando variáveis distintas, ora categóricas, ora quantitativas.

Para avaliar cada modelo perante uma amostra diferente, com dados diferentes, utilizou-se técnicas de tratamento, preparação e treinamento com os dados afim de obter os melhores resultados de cada modelo.

A amostra 1 foi constituída totalmente com os dados categóricos do conjunto PH2 ([MENDONÇA et al., 2013](#)). Para esta situação, foi avaliado o teste binário, isto é, o modelo conseguir prever a presença ou não de câncer. Para o teste multiclasse, predizer a qual classe aquele dado pertence: Melanoma, Nevo Atípico e Nevo Típico.

Neste primeiro caso, o modelo *Gradient Boosting* mostrou-se como o melhor. Este por sua vez apresentou uma acurácia de 95% e F_M 0,9390. O treinamento utilizou o processo de validação cruzada, sendo os melhores hiperparâmetros: 100 estimadores e uma taxa de aprendizado $\eta = 0.1$.

Figura 77 – Matriz de confusão e curva ROC obtida para o o modelo *Gradient Boosting* amostra 1 teste binário



Fonte: Autor.

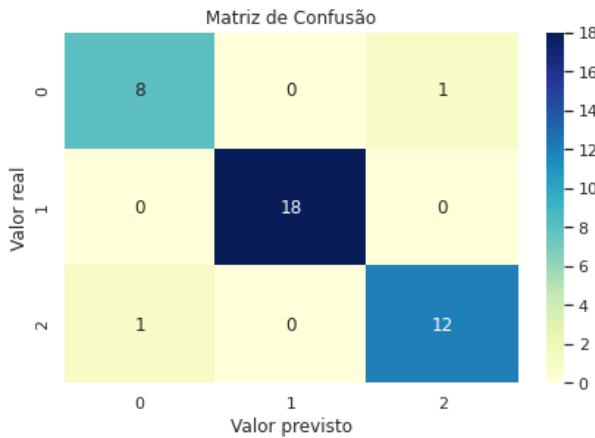
A Figura 77 mostra a matriz de confusão e curva ROC para o modelo obtidas na fase de teste. Analisando primeiramente a curva ROC é possível perceber que este modelo conseguiu quase que uma curva perfeita, o que indica que o modelo conseguiu prever bem os dados gerados.

Ademais, também é possível observar esta qualidade nos resultados dentro da matriz de confusão. Além de apresentar ótimos resultados, predizer mais casos como câncer do que deveria é algo importante no diagnóstico de doenças. Isto implica um pensamento médico que favorece a utilização deste modelo na medicina.

Para a avaliação multiclasse, contendo 3 classes distintas, o melhor modelo foi o

SVM com kernel polinomial e utilização apenas da validação cruzada, alcançando uma acurácia de 0,9500 e F_M de 0,9590. Para este modelo, os melhores hiperparâmetros foram: $C = 5$, grau 5 e ϵ -sensitive = 0.5. A Figura 78 mostra as previsões feitas por este modelo no conjunto de teste.

Figura 78 – Matriz de confusão obtida para o o modelo SVM - Linear para amostra 1 teste multiclasse



Fonte: Autor.

Para esta amostra 1, é importante salientar que os resultados de todos os modelos, em especial estes dois, foram altos para ambos os testes (binário e multiclasse). Isto se dá pois a amostra foi gerada com dados categóricos provenientes de avaliações médicas sobre as imagens. Estas avaliações têm um grande peso decisivo na hora de treinar o modelo o modelo.

Já a amostra 2 foi gerada a partir das informações extraídas das imagens fornecidas pelo conjunto PH2. Estas informações são grandezas pertinentes ao problema destacado nesta monografia, câncer de pele.

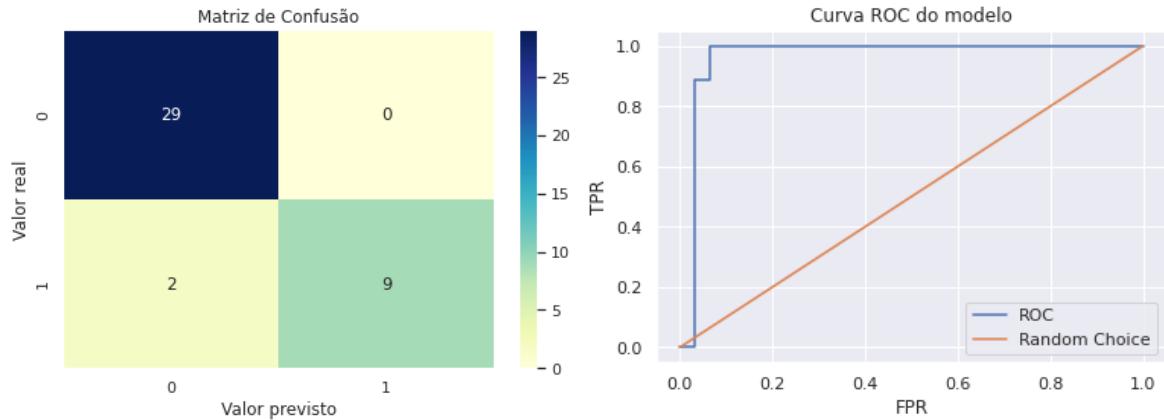
Durante o teste binário, o melhor modelo foi o SVM de kernel linear, utilizando como hiperparâmetros: $C = 1$ e ϵ -sensitive = 0.01. Este por sua vez obteve uma acurácia de 95% e F_M de 0,9270. Os resultados obtidos com este modelo estão expostos através da Figura 79.

O melhor resultado deste modelo foi obtido no tratamento 5, isto é, com a normalização dos dados, aplicação da técnica SMOTE e validação cruzada.

Avaliando a matriz de confusão, que representa seu desempenho, é possível perceber que o modelo não conseguiu prever todos os casos melanoma corretamente.

Já na etapa de análise multiclasse desta amostra, a maioria dos modelos apresentaram certa dificuldade e tiveram uma grande queda de desempenho quando comparado com os outros testes.

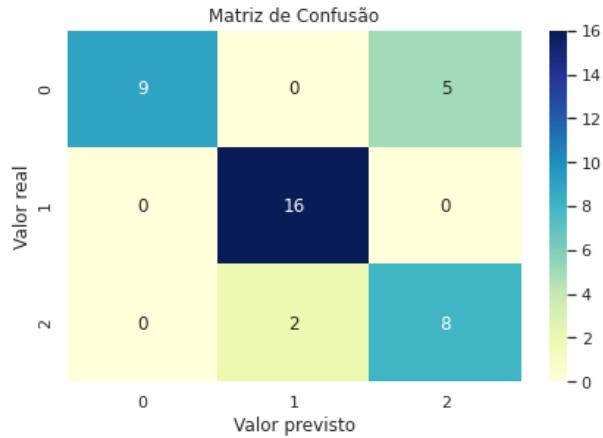
Figura 79 – Matriz de confusão e curva ROC obtida para o o modelo SVM - Linear amostra 2 teste binário



Fonte: Autor.

O melhor modelo neste caso, foi a Floresta Aleatória (*Random Forest*). Obtendo rendimento com acurácia de 92,5% e F_M de 0,8963, quando submetido a normalização dos dados e aplicação do SMOTE, sendo os hiperparâmetros do modelo: critério de separação de índice de Gini, 10 estimadores e profundidade de 50.

Figura 80 – Matriz de confusão obtida para o o modelo *Random Forest* para amostra 2 teste multiclasse



Fonte: Autor.

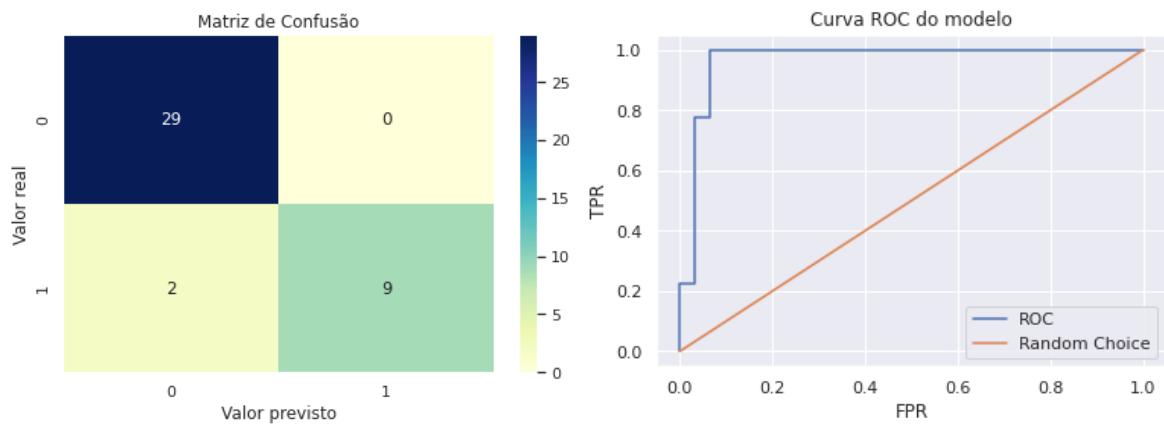
A Figura 80 mostra os resultados obtidos por este modelo para a análise em questão. Nela, ressalva-se que o modelo apresentou boa predição para o caso Nevo Típico. No caso de predizer as outras duas situações: Melanoma e Atípico, cometem erros, principalmente para Melanoma, onde o modelo errou 5 de 14.

A amostra 3, uma combinação entre as variáveis extraídas e categóricas fornecidas, foi obtido através do conjunto PH2. O objetivo desta amostra é avaliar se os modelos

conseguiam uma melhora quando estes dados fossem agrupados. De fato, alguns modelos obtiveram um aumento no desempenho, porém outros não se adaptaram tão bem.

Para o teste binário, o melhor modelo foi a Regressão Logística, com uma acurácia de 95% e F_M de 0,9282. Para este caso, o melhor modelo foi obtido utilizando a aplicação do SMOTE e validação cruzada somente, com os hiperparâmetros $C = 500$ e solver liblinear que utiliza uma otimização baseada em *gradient descent*.

Figura 81 – Matriz de confusão e curva ROC obtida para o o modelo Regressão Logística amostra 3 teste binário



Fonte: Autor.

A Figura 81 ilustra o desempenho obtido por esse modelo. Comparando este resultado com os outros métodos, conforme mostra as Tabelas do Anexo A, a junção das variáveis categóricas com as qualitativas corroboram para o aumento do desempenho se comparado com a amostra 2 e 3.

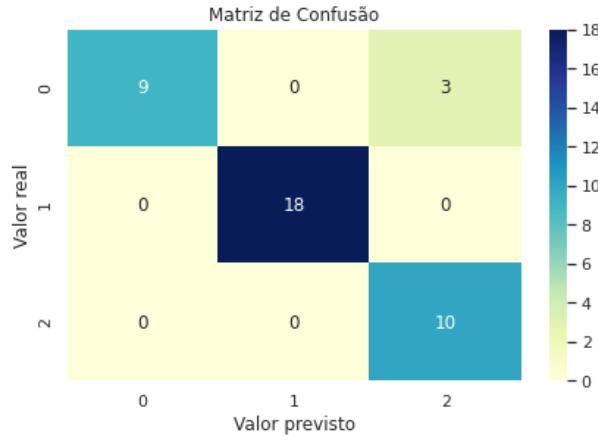
Para o caso multiclasse, o melhor modelo obteve 92,50% de acurácia e 0,9447 de F_M , sendo este modelo o Árvores Extremamente Aleatórias (*Extra Tree*). O melhor desempenho deste modelo foi alcançado aplicando a normalização sobre os dados e validação cruzada. Os hiperparâmetros utilizados foram índice de Gini, 10 estimadores e profundidade de 50. A Figura 82 mostra o resultado obtido pelo modelo para o caso multiclasse.

Diferente das amostras 1,2 e 3, a amostra 4 foi obtida a partir do conjunto HAM10000. Conforme discutido na Seção 3.2, o conjunto apresenta variáveis como idade, gênero do paciente e localização. Sendo estas duas últimas de caráter categórico.

A amostra 4 foi montada utilizando estes dados e a extração das características das 10015 imagens do conjunto HAM10000 e suas 7 classes distintas. Sendo as classes Carcinoma basocelular, Melanoma e Carcinoma epidermoide tipos cânceres que são análise desta monografia. No problema de teste binário, o foco do modelo é encontrar e classificar estas classes como câncer e o restante como normal.

Neste quesito, o melhor modelo para esse teste binário foi a RNA-MLP com 3

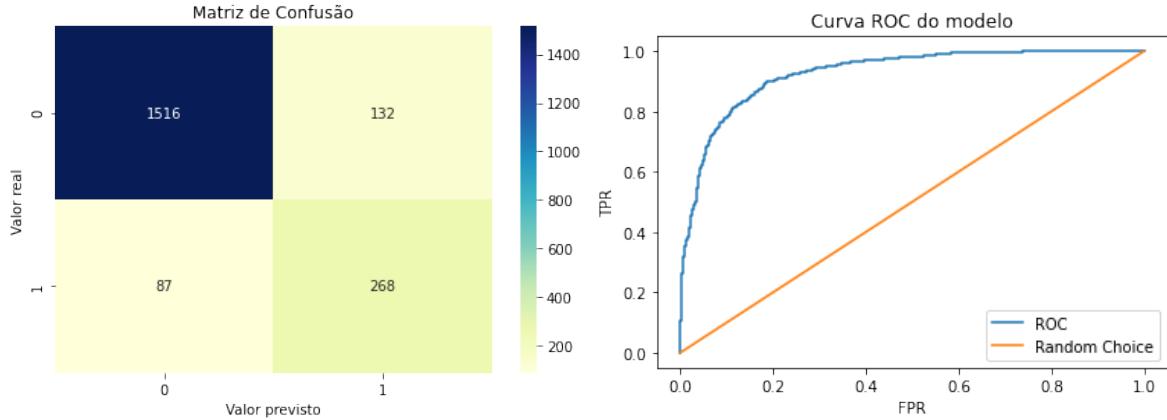
Figura 82 – Matriz de confusão obtida para o o modelo Árvores Extremamente Aleatórias para amostra 3 teste multiclasse



Fonte: Autor.

camadas ocultas. Este modelo obteve acurácia de 89,07% e F_M de 0,8207 quando sujeito a normalização dos dados e aplicação da técnica SMOTE. Os hiperparâmetros selecionados neste caso foram a função de ativação ReLU e 3 camadas ocultas, sendo a primeira com 500, a segunda com 300 e terceira com 100.

Figura 83 – Matriz de confusão e curva ROC obtida para o o modelo RNA-MLP com 3 camadas ocultas amostra 4 teste binário



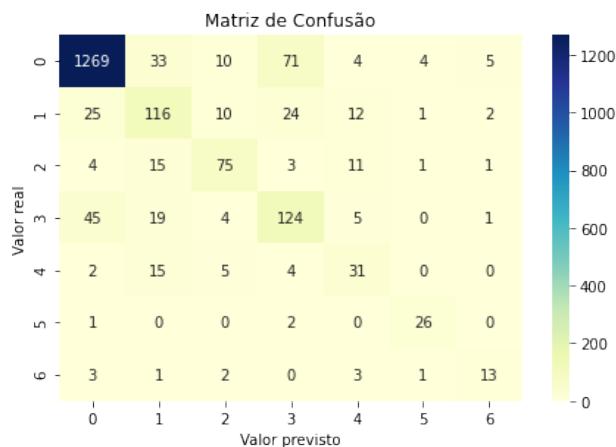
Fonte: Autor.

A Figura 83 mostra o resultado obtido por este modelo. Neste, é possível observar um alto desempenho, apesar da grande disparidade das amostras em cada classe.

No caso multiclasse para esta amostra, o melhor modelo também foi um rede neural. Neste caso, o melhor desempenho foi obtido quando os dados foram submetidos a normalização e ao SMOTE pelo modelo RNA-MLP com duas camadas ocultas. Os hiperparâmetros utilizados foram a função de ativação ReLU e 2 camadas ocultas, sendo

a primeira com 500 neurônio e a segunda com 300. Neste caso, o modelo alcançou uma acurácia de 82,44% e F_M de 0,8160.

Figura 84 – Matriz de confusão obtida para o o modelo RNA-MLP com 2 camadas ocultas para amostra 4 teste multiclasse



Fonte: Autor.

Apesar desta monografia seguir uma metodologia própria, é possível fazer uma comparação com resultados obtidos em outros trabalhos. Cada trabalho utiliza uma metodologia distinta, isto é, avalia características distintas para introduzir no treinamento do modelo visando uma melhora na predição da doença ou utilizam métricas diferentes para avaliar o melhor modelo.

[Maia \(2015\)](#) resolveu a mesma problemática apresentada neste trabalho, analisando o conjunto PH2 porém com outras etapas e outras características destacadas. No caso, seu trabalho utilizou de outros modelos e de atributos diferentes. Entretanto, é interessante ressaltar que em seus treinamentos com seleção de atributos de forma, cor e textura os melhores modelos analisados por ele foram *RandomForest* e o *FuzzyDT*. Sendo seus erros neste trabalho 8% e 5% respectivamente.

Outra artigo importante foi realizada por [Barata et al. \(2013\)](#), sobre um conjunto contendo 176 imagens dermatoscópicas (25 melanomas e 151 nevos). Os modelos avaliados nesta monografia foram SVM e KNN. A Abordagem tomada utiliza uma diferenciação do problema em duas variáveis: globais e locais. Nesta primeira, as já mencionadas textura, forma e cor e a segunda considerando porções pequenas da imagem. Os melhores modelos obtiveram para as variáveis globais uma revocação de 0,96 e especificidade de 0,80. Para o caso das variáveis locais, o modelo apresentou uma melhora na detecção de casos melanoma com revocação de 1,0 e especificidade de 0,75.

No caso do artigo criado por [Barros, Morais e Fernandes \(2017\)](#) que utiliza somente 9 descritores, contendo características de forma e cor, foi treinado um modelo RNA-MLP. O autor utilizou também o conjunto PH2 para teste. Nesta monografia, foi utilizada 2

camadas ocultas sendo a primeira contendo 10 e a segunda 24. Ao submeter 30 imagens a treinamento, sendo elas 22 nevos e 8 melanomas. Neste caso, [Barros, Morais e Fernandes \(2017\)](#) obteve uma precisão de 90%, uma especificidade de 90.9% e revocação de 87.5%.

Contudo, os resultados apresentados por [Maia \(2015\)](#), [Barata et al. \(2013\)](#) e [Barros, Morais e Fernandes \(2017\)](#) são distintos dos obtidos nesta monografia. Isto ocorre pela utilização de metodologias diferentes como atributos e conjunto de dados. A diferença nos resultados, podendo encontrar modelhos melhores, reforça a importância da utilização do aprendizado de máquina na predição de câncer de pele.

Aliás, pode-se encontrar semelhanças entre os resultados apontados no Anexo A e estes mostrados acima. Entretanto, na literatura não foi possível encontrar resultados que apresentassem os hiperparâmetros atingidos para cada monografia. Além disso, não foi encontrado nenhum trabalho que utiliza-se as mesmas informações extraídas das imagens em relação a esta monografia.

5 Conclusão

Pode-se concluir que o presente trabalho conseguiu uma gama de resultados para cada modelo em diversas situações distintas. Para um conjunto de dados categóricos, o melhor modelo foi o *Gradient Boosting* quando necessário analisar a presença de câncer (problema binário). Apresentando uma acurácia de 95% e F_M de 0,9390. Para a análise das 3 classes com os mesmos dados, o melhor modelo foi SVM com kernel polinomial, apresentando uma acurácia de 95% e F_M de 0,9590.

Para o conjunto contendo somente as variáveis extraídas das imagens, os melhores modelos foram SVM com kernel linear (acurácia de 95% e F_M de 0,9270) e Floresta aleatória (acurácia 92,5% e F_M de 0,8963) para os testes binários e multiclasse respectivamente. Ao fundir ambas as amostras anteriores, o melhor modelo foi a Regressão Logística que obteve uma acurácia de 95% e F_M de 0,9282 para análise de duas classes. No caso multiclasse, o modelo com melhor desempenho foi as Árvores Extremamente Aleatórias, alcançando um acurácia de 92,50% de acurácia e 0,9447 de F_M .

No caso da amostra 4, contendo muito mais exemplares e mais classes, além de contemplar outros tipos de cânceres de pele além do melanoma, os melhores modelos utilizaram o algoritmo RNA-MLP. No caso do problema de identificar a ocorrência de câncer, o melhor modelo foi a RNA-MLP com 3 camadas ocultas, cujo desempenho foi 89,07% e 0,8207 de acurácia e F_M respectivamente. Já no problema multiclasse, onde o modelo deveria identificar 7 classes diferentes, o melhor modelo foi uma RNA-MLP com 2 camadas ocultas. Tal algoritmo obteve acurácia de 82,44% e F_M de 0,8160.

Ademais, ressalva-se o fato dos modelos apresentarem resultados muito distintos dependo do tratamento utilizado para seu treinamento. O desempenho destes modelos ocasionalmente apresentou-se significa melhora quando os dados eram submetidos à normalização ou aplicação do SMOTE. Sendo então, tais técnicas de pré-processamento são úteis para generalizar o problema proposto.

Na tentativa de predizer a ocorrência de câncer entre os conjuntos de dados analisados, é notável a diferença na precisão em geral dos modelos quando se avalia dados categóricos e quantitativos. Isto se dá devido ao peso e importância dos dados categóricos por apresentarem uma resposta binária.

Por fim, apesar dos conceitos de ML serem antigos, eles veem sendo utilizados cada vez mais por conta da capacidade de processamento e informações disponíveis. Por esse motivo, novas teorias e novos modelos estão sendo criados. Com esse avanço, a medicina se beneficia de novos modelos que visam diagnosticar pacientes de maneira mais rápida e fácil. Por mais que a utilização destes modelos não substitua a necessidade de um profissional da

saúde, a utilização destas algoritmos podem ser uma ferramenta que auxilia no processo de triagem de pacientes.

Referências

- BARATA, C. et al. Two systems for the detection of melanomas in dermoscopy images using texture and color features. *IEEE SYSTEMS JOURNAL*, FCT, 2013. Citado 3 vezes nas páginas 79, 95 e 96.
- BARROS, W.; MORAIS, D.; FERNANDES, M. C. Detecção de câncer de pele com redes neurais artificiais. *CBIC: XIII Brazilian Congress on Computational Intelligence*, 2017. Acessado em: 27 de Outubro de 2020. Disponível em: <<http://cbic2017.org/papers/cbic-paper-108.pdf>>. Citado 8 vezes nas páginas 14, 15, 60, 64, 65, 79, 95 e 96.
- BOTELHO, G. M. *Processamento de Imagens no Auxílio à Detecção de Câncer de Pele utilizando Redes Neurais Artificiais*. Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Goiás, Campus Catalão como requisito parcial para obtenção do título de Bacharel em Ciência da Computação, Catalão, 2007. Citado 8 vezes nas páginas 1, 3, 4, 6, 14, 15, 16 e 60.
- BURGER, W.; BURGE, M. *Digital Image Processing: An Algorithmic Introduction Using Java*. Second edition. London: Springer, 2016. ISBN 9781447166849. Citado 7 vezes nas páginas 14, 15, 16, 17, 18, 60 e 61.
- DERMNETNZ. *Dermoscopy: Dermatoscopic features*. 2008. Acessado em: 22 de Outubro de 2020. Disponível em: <<https://dermnetnz.org/cme/dermoscopy-course/dermoscopic-features/>>. Citado 6 vezes nas páginas 7, 8, 9, 10, 11 e 12.
- DEVORE, J. *Probability and Statistics for Engineering and the Sciences*. Ninth edition. [S.l.]: Cengage Learning, 2016. ISBN 9781305251809. Citado na página 61.
- FACELI, K. et al. *Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina*. Rio de Janeiro: LTC, 2011. ISBN 9788521618805. Citado 34 vezes nas páginas 1, 18, 19, 20, 21, 22, 27, 28, 29, 30, 31, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 55, 59, 61, 62, 69 e 71.
- GANZELI, H. et al. Skan: Skin scanner – system for skin cancer detection using adaptive techniques. *IEEE LATIN AMERICA TRANSACTIONS*, v. 9, p. 205–212, April 2011. Citado 4 vezes nas páginas 60, 65, 66 e 67.
- GATTO, R. C. *Simulation based data augmentation approach for Machine Learning models on audio under uncontrolled sound environments*. Thesis of Doctor of Science in Electronic Engineering and Computing - Informatics - Instituto Tecnológico de Aeronáutica, São José dos Campos, 2020. Citado 2 vezes nas páginas 29 e 30.
- GBM. *O Melanoma*. 2020. Acessado em: 21 de Outubro de 2020. Disponível em: <<https://gbm.org.br/o-melanoma/#nevos-e-melanoma>>. Citado 3 vezes nas páginas 4, 5 e 6.
- GÉRON, A. *Mãos à Obra Aprendizado de Máquina com Scikit-Learn e TensorFlow: Conceitos, ferramentas e técnicas para construção de sistemas inteligentes*. [S.l.]: Alta Books, 2019. Traduzido por Rafael Contatori. ISBN 9788550803814. Citado 28 vezes nas

páginas 19, 20, 21, 22, 24, 25, 26, 27, 29, 31, 32, 33, 34, 35, 36, 41, 42, 44, 45, 46, 47, 50, 51, 52, 53, 55, 69 e 79.

IBM. *Over-sampling*. 2020. Acessado em: 07 de Novembro de 2020. Disponível em: <https://imbalanced-learn.readthedocs.io/en/stable/over_sampling.html#smote-adasyn>. Citado 2 vezes nas páginas 20 e 21.

INCA. *Câncer de Pele Melanoma*. 2020. Acessado em: 14 de Outubro de 2020. Disponível em: <<https://www.inca.gov.br/tipos-de-cancer/cancer-de-pele-melanoma>>. Citado 6 vezes nas páginas 3, 4, 6, 7, 75 e 78.

KITTLER, H. et al. Standardization of terminology in dermoscopy/dermatoscopy: Results of the third consensus conference of the international society of dermoscopy. *Journal of American Academy of Dermatology*, American Academy of Dermatology, v. 74, p. 1903–1106, 2016. Disponível em: <[https://www.jaad.org/article/S0190-9622\(15\)02615-8/fulltext#secsectitle0040](https://www.jaad.org/article/S0190-9622(15)02615-8/fulltext#secsectitle0040)>. Citado na página 10.

LOPES, T. G. G. *Implementação computacional dos modelos de machine learning para predição de câncer de pele*. 2020. Acessado em: 20 de Novembro de 2020. Disponível em: <<https://drive.google.com/drive/folders/1GYyQH9HTQSggLeqBYqQUiOhnvo6ipIFN?usp=sharing>>. Citado na página 81.

MAIA, H. W. *UMA PROPOSTA PARA DETECÇÃO AUTOMÁTICA DE CÂNCER DE PELE BASEANDO-SE EM CARACTERÍSTICAS DE FORMA, COR E TEXTURA*. Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação - associação ampla entre a Universidade do Estado do Rio Grande do Norte e a Universidade Federal Rural do Semi-Árido, para a obtenção do título de Mestre em Ciência da Computação. Mossoró, 2015. Citado 2 vezes nas páginas 95 e 96.

MALACARA, D. *Color vision and colorimetry : theory and applications*. Second edition. [S.l.]: SPIE, 2011. ISBN 9780819483973. Citado 2 vezes nas páginas 16 e 18.

MASSI, G.; LEBOIT, P. *Histological Diagnosis of Nevi and Melanoma*. Second edition. [S.l.]: Springer, 2014. ISBN 9783642373114. Citado 6 vezes nas páginas 4, 6, 7, 8, 10 e 11.

MENDONÇA, T. et al. Ph2: A dermoscopic image database for research and benchmarking. *International Conference of the IEEE Engineering in Medicine and Biology Society*, v. 589, p. 1–15, 2013. Citado 7 vezes nas páginas 1, 7, 13, 24, 57, 58 e 90.

OPENCV. *Color conversions*. 2020. Acessado em: 05 de Novembro de 2020. Disponível em: <https://docs.opencv.org/4.2.0/de/d25/imgproc_color_conversions.html#color_convert_rgb_hls>. Citado 3 vezes nas páginas 17, 18 e 63.

RUSSO, T. et al. Dermoscopy pathology correlation in melanoma. *Journal of Dermatology*, Japanese Dermatological Association, v. 44, p. 507–514, 2017. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1111/1346-8138.13629>>. Citado 3 vezes nas páginas 7, 10 e 11.

SCIKIT-LEARN. *Cross-validation: evaluating estimator performance*. 2020. Acessado em: 07 de Novembro de 2020. Disponível em: <https://scikit-learn.org/stable/modules/cross_validation.html>. Citado 3 vezes nas páginas 22, 23 e 69.

- SCIKIT-LEARN. *Decision Trees*. 2020. Acessado em: 09 de Novembro de 2020. Disponível em: <<https://scikit-learn.org/stable/modules/tree.html>>. Citado na página 31.
- SCIKIT-LEARN. *Ensemble methods*. 2020. Acessado em: 09 de Novembro de 2020. Disponível em: <<https://scikit-learn.org/stable/modules/ensemble.html>>. Citado 5 vezes nas páginas 33, 34, 35, 36 e 37.
- SCIKIT-LEARN. *Linear Models*. 2020. Acessado em: 07 de Novembro de 2020. Disponível em: <https://scikit-learn.org/stable/modules/linear_model.html>. Citado 2 vezes nas páginas 24 e 25.
- SCIKIT-LEARN. *Metrics and scoring: quantifying the quality of predictions*. 2020. Acessado em: 14 de Novembro de 2020. Disponível em: <https://scikit-learn.org/stable/modules/model_evaluation.html#regression-metrics>. Citado 6 vezes nas páginas 50, 51, 52, 53, 54 e 55.
- SCIKIT-LEARN. *Naive Bayes*. 2020. Acessado em: 07 de Novembro de 2020. Disponível em: <https://scikit-learn.org/stable/modules/naive_bayes.html>. Citado 2 vezes nas páginas 28 e 29.
- SCIKIT-LEARN. *Nearest Neighbors*. 2020. Acessado em: 10 de Novembro de 2020. Disponível em: <<https://scikit-learn.org/stable/modules/neighbors.html>>. Citado 3 vezes nas páginas 37, 38 e 39.
- SCIKIT-LEARN. *Neural network models (supervised)*. 2020. Acessado em: 13 de Novembro de 2020. Disponível em: <https://scikit-learn.org/stable/modules/neural_networks_supervised.html>. Citado 2 vezes nas páginas 45 e 49.
- SCIKIT-LEARN. *Preprocessing data*. 2020. Acessado em: 07 de Novembro de 2020. Disponível em: <<https://scikit-learn.org/stable/modules/preprocessing.html>>. Citado 3 vezes nas páginas 20, 21 e 71.
- SCIKIT-LEARN. *Support Vector Machines*. 2020. Acessado em: 11 de Novembro de 2020. Disponível em: <<https://scikit-learn.org/stable/modules/svm.html>>. Citado 3 vezes nas páginas 42, 43 e 87.
- SCIKIT-LEARN. *Tuning the hyper-parameters of an estimator*. 2020. Acessado em: 07 de Novembro de 2020. Disponível em: <https://scikit-learn.org/stable/modules/grid_search.html#grid-search>. Citado 2 vezes nas páginas 22 e 23.
- SILVA, I. N. d.; SPATTI, D. H.; FLAUZINO, R. A. *Redes Neurais Artificiais para engenharia e ciências aplicadas: fundamentos teóricos e aspectos práticos*. 2. ed. São Paulo: Artliber, 2010. Citado 2 vezes nas páginas 44 e 46.
- SZELISK, R. *Computer Vision: Algorithms and applications*. London: Springer, 2011. ISBN 9781848829343. Citado na página 14.
- TSCHANDL, P. *The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions*. Harvard Dataverse, 2018. Disponível em: <<https://doi.org/10.7910/DVN/DBW86T>>. Citado 6 vezes nas páginas 1, 13, 24, 57, 58 e 59.

ZHOU S. K. AND RUECKERT, D.; FICHTINGER, G. *Handbook of Medical Image Computing and Computer assisted intervention*. [S.l.]: Academic Press, 2020. ISBN 9780128161760. Citado 3 vezes nas páginas [12](#), [13](#) e [57](#).

Anexos

ANEXO A – Resultados para cada modelo

Tabela 10 – Resultados obtidos para amostra 1 teste binário

Modelo	T	<i>Ac</i>	<i>Pr</i>	<i>Rec</i>	<i>F</i> ₁	<i>F</i> _β	<i>B</i> _S	<i>ROC</i> _S	<i>MCC</i>	<i>F</i> _M
Regressão Logística	1	0,8750	0,8333	0,5556	0,6667	0,7576	0,7616	0,9785	0,6120	0,8326
	2	0,9250	0,8750	0,7778	0,8235	0,8537	0,8728	0,9642	0,7783	0,8969
	3	0,9250	0,8750	0,7778	0,8235	0,8537	0,8728	0,9642	0,7783	0,8969
	4	0,8500	0,6667	0,6667	0,6667	0,6667	0,7849	0,9140	0,5699	0,7885
	5	0,9250	0,8000	0,8889	0,8421	0,8163	0,9122	0,9749	0,7950	0,9011
N BG	1	0,7750	0,5000	1,0000	0,6667	0,5556	0,8548	0,9821	0,5957	0,7975
	2	0,9000	0,6923	1,0000	0,8182	0,7377	0,9355	0,9821	0,7765	0,8851
	3	0,9000	0,6923	1,0000	0,8182	0,7377	0,9355	0,9821	0,7765	0,8851
	4	0,9250	0,7500	1,0000	0,8571	0,7895	0,9516	0,9821	0,8231	0,9077
	5	0,9000	0,6923	1,0000	0,8182	0,7377	0,9355	0,9821	0,7765	0,8851
N BB	1	0,8750	0,7000	0,7778	0,7368	0,7143	0,8405	0,9570	0,6567	0,8373
	2	0,8750	0,7000	0,7778	0,7368	0,7143	0,8405	0,9570	0,6567	0,8373
	3	0,8750	0,7000	0,7778	0,7368	0,7143	0,8405	0,9570	0,6567	0,8373
	4	0,9250	0,7500	1,0000	0,8571	0,7895	0,9516	0,9642	0,8231	0,9017
	5	0,9000	0,7273	0,8889	0,8000	0,7547	0,8961	0,9642	0,7408	0,8716
Decision Tree	1	0,9250	0,8750	0,7778	0,8235	0,8537	0,8728	0,8728	0,7783	0,8664
	2	0,8500	0,7143	0,5556	0,6250	0,6757	0,7455	0,9032	0,5397	0,7748
	3	0,9000	0,7778	0,7778	0,7778	0,7778	0,8566	0,9642	0,7133	0,8662
	4	0,9000	0,7273	0,8889	0,8000	0,7547	0,8961	0,9695	0,7408	0,8734
	5	0,9000	0,7778	0,7778	0,7778	0,7778	0,8566	0,9642	0,7133	0,8662
Random Forest	1	0,8750	0,8333	0,5556	0,6667	0,7576	0,7616	0,9534	0,6120	0,8242
	2	0,9000	0,8571	0,6667	0,7500	0,8108	0,8172	0,9767	0,6972	0,8682
	3	0,8750	0,7500	0,6667	0,7059	0,7317	0,8011	0,9534	0,6286	0,8287
	4	0,8750	0,7500	0,6667	0,7059	0,7317	0,8011	0,9570	0,6286	0,8299
	5	0,8750	0,7500	0,6667	0,7059	0,7317	0,8011	0,9534	0,6286	0,8287
Extra Tree	1	0,9000	0,8571	0,6667	0,7500	0,8108	0,8172	0,9642	0,6972	0,8641
	2	0,8750	0,8333	0,5556	0,6667	0,7576	0,7616	0,9677	0,6120	0,8290
	3	0,8750	0,7500	0,6667	0,7059	0,7317	0,8011	0,9534	0,6286	0,8287
	4	0,9000	0,8571	0,6667	0,7500	0,8108	0,8172	0,9588	0,6972	0,8623
	5	0,9500	0,8889	0,8889	0,8889	0,8889	0,9283	0,9749	0,8566	0,9307
KNN	1	0,9000	1,0000	0,5556	0,7143	0,8621	0,7778	0,9803	0,7015	0,8734
	2	0,8750	0,8333	0,5556	0,6667	0,7576	0,7616	0,9552	0,6120	0,8248
	3	0,8750	0,8333	0,5556	0,6667	0,7576	0,7616	0,9552	0,6120	0,8248
	4	0,9000	0,7778	0,7778	0,7778	0,7778	0,8566	0,9731	0,7133	0,8692
	5	0,8750	0,7500	0,6667	0,7059	0,7317	0,8011	0,8011	0,6286	0,7780

Continuação

Modelo	T	Ac	Pr	Rec	F_1	F_β	B_S	ROC_S	MCC	F_M
RNN	1	0,8750	1,0000	0,4444	0,6154	0,8000	0,7222	0,9677	0,6186	0,8300
	2	0,9000	1,0000	0,5556	0,7143	0,8621	0,7778	0,9570	0,7015	0,8656
	3	0,9000	1,0000	0,5556	0,7143	0,8621	0,7778	0,9570	0,7015	0,8656
	4	0,8500	1,0000	0,3333	0,5000	0,7143	0,6667	0,6864	0,5285	0,6891
	5	0,8500	1,0000	0,3333	0,5000	0,7143	0,6667	0,6864	0,5285	0,6891
Ada Boost	1	0,8500	0,6364	0,7778	0,7000	0,6604	0,8244	0,9355	0,6067	0,8067
	2	0,9250	0,8750	0,7778	0,8235	0,8537	0,8728	0,9677	0,7783	0,8981
	3	0,9250	0,8750	0,7778	0,8235	0,8537	0,8728	0,9677	0,7783	0,8981
	4	0,9000	0,7778	0,7778	0,7778	0,7778	0,8566	0,9677	0,7133	0,8674
	5	0,9000	0,8571	0,6667	0,7500	0,8108	0,8172	0,9570	0,6972	0,8617
Gradient Boosting	1	0,9250	0,8750	0,7778	0,8235	0,8537	0,8728	0,9785	0,7783	0,9016
	2	0,9500	1,0000	0,7778	0,8750	0,9459	0,8889	0,9821	0,8548	0,9390
	3	0,9250	0,8750	0,7778	0,8235	0,8537	0,8728	0,9642	0,7783	0,8969
	4	0,9000	0,8571	0,6667	0,7500	0,8108	0,8172	0,9570	0,6972	0,8617
	5	0,9250	0,8750	0,7778	0,8235	0,8537	0,8728	0,9892	0,7783	0,9052
SVM - Linear	1	0,8750	0,8333	0,5556	0,6667	0,7576	0,7616	0,9319	0,6120	0,8170
	2	0,9000	0,8571	0,6667	0,7500	0,8108	0,8172	0,9498	0,6972	0,8593
	3	0,9000	0,8571	0,6667	0,7500	0,8108	0,8172	0,9498	0,6972	0,8593
	4	0,9000	0,8571	0,6667	0,7500	0,8108	0,8172	0,9534	0,6972	0,8605
	5	0,9250	0,8750	0,7778	0,8235	0,8537	0,8728	0,9319	0,7783	0,8861
SVM - Polinomial	1	0,8750	0,8333	0,5556	0,6667	0,7576	0,7616	0,9857	0,6120	0,8350
	2	0,9000	0,8571	0,6667	0,7500	0,8108	0,8172	0,9534	0,6972	0,8605
	3	0,9000	0,8571	0,6667	0,7500	0,8108	0,8172	0,9534	0,6972	0,8605
	4	0,9000	0,7778	0,7778	0,7778	0,7778	0,8566	0,9606	0,7133	0,8650
	5	0,9500	1,0000	0,7778	0,8750	0,9459	0,8889	0,9534	0,8548	0,9294
SVM - RBF	1	0,8500	0,7143	0,5556	0,6250	0,6757	0,7455	0,9427	0,5397	0,7879
	2	0,8500	0,7143	0,5556	0,6250	0,6757	0,7455	0,9498	0,5397	0,7903
	3	0,8500	0,7143	0,5556	0,6250	0,6757	0,7455	0,9498	0,5397	0,7903
	4	0,9250	0,8750	0,7778	0,8235	0,8537	0,8728	0,9749	0,7783	0,9004
	5	0,8750	0,8333	0,5556	0,6667	0,7576	0,7616	0,9642	0,6120	0,8278
SVM - Sigmoidal	1	0,9000	0,7273	0,8889	0,8000	0,7547	0,8961	0,9427	0,7408	0,8645
	2	0,8500	0,6667	0,6667	0,6667	0,6667	0,7849	0,8566	0,5699	0,7694
	3	0,8500	0,6667	0,6667	0,6667	0,6667	0,7849	0,8566	0,5699	0,7694
	4	0,8500	0,6364	0,7778	0,7000	0,6604	0,8244	0,8566	0,6067	0,7805
	5	0,8500	0,7143	0,5556	0,6250	0,6757	0,7455	0,9498	0,5397	0,7903
RNA - 1 Camada	1	0,9250	0,8750	0,7778	0,8235	0,8537	0,8728	0,9749	0,7783	0,9004
	2	0,9250	0,8750	0,7778	0,8235	0,8537	0,8728	0,9606	0,7783	0,8957
	3	0,9000	0,7778	0,7778	0,7778	0,7778	0,8566	0,9677	0,7133	0,8674
	4	0,9500	1,0000	0,7778	0,8750	0,9459	0,8889	0,9677	0,8548	0,9342
	5	0,9250	0,8750	0,7778	0,8235	0,8537	0,8728	0,9606	0,7783	0,8957

Continuação

Modelo	T	Ac	Pr	Rec	F_1	F_β	B_S	ROC_S	MCC	F_M
RNA - 2 Camadas	1	0,9000	0,8571	0,6667	0,7500	0,8108	0,8172	0,9606	0,6972	0,8629
	2	0,8500	0,6667	0,6667	0,6667	0,6667	0,7849	0,9355	0,5699	0,7957
	3	0,9250	0,8750	0,7778	0,8235	0,8537	0,8728	0,9677	0,7783	0,8981
	4	0,9000	0,8571	0,6667	0,7500	0,8108	0,8172	0,9713	0,6972	0,8664
	5	0,9000	0,8571	0,6667	0,7500	0,8108	0,8172	0,9534	0,6972	0,8605
RNA - 3 Camadas	1	0,9000	0,7778	0,7778	0,7778	0,7778	0,8566	0,9606	0,7133	0,8650
	2	0,9000	0,8571	0,6667	0,7500	0,8108	0,8172	0,9606	0,6972	0,8629
	3	0,9250	0,8750	0,7778	0,8235	0,8537	0,8728	0,9355	0,7783	0,8873
	4	0,8750	0,7000	0,7778	0,7368	0,7143	0,8405	0,9319	0,6567	0,8289
	5	0,9000	0,8571	0,6667	0,7500	0,8108	0,8172	0,9606	0,6972	0,8629

Fonte: Autor.

Legenda: Ac = Acurácia / Pr = Precisão / Rec = Revocação / F_1 = Medida F para $\beta = 1$ / F_β = Medida F / B_S = Acurácia Balanceada / ROC_S Pontuação ROC / MCC = Coeficiente de Correlação de Matthews / F_M Métrica Final.

Tabela 11 – Resultados obtidos para amostra 1 teste multiclasse

Modelo	T	Ac	Pr	Rec	F_1	F_β	B_S	ROC_S	MCC	F_M
Regressão Logística	1	0,9250	0,9254	0,9250	0,9242	0,9247	0,9003	0,9887	0,8835	0,9379
	2	0,9250	0,9254	0,9250	0,9242	0,9247	0,9003	0,9809	0,8835	0,9353
	3	0,9250	0,9254	0,9250	0,9242	0,9247	0,9003	0,9809	0,8835	0,9353
	4	0,8750	0,8741	0,8750	0,8736	0,8737	0,8376	0,9376	0,8051	0,8830
	5	0,9250	0,9279	0,9250	0,9255	0,9267	0,9117	0,9880	0,8844	0,9421
NBG	1	0,7000	0,8714	0,7000	0,6314	0,6545	0,6923	0,9896	0,6421	0,7788
	2	0,9000	0,9308	0,9000	0,9000	0,9145	0,8974	0,9887	0,8596	0,9335
	3	0,9000	0,9308	0,9000	0,9000	0,9145	0,8974	0,9887	0,8596	0,9335
	4	0,9250	0,9438	0,9250	0,9255	0,9342	0,9231	0,9904	0,8921	0,9492
	5	0,9000	0,9308	0,9000	0,9000	0,9145	0,8974	0,9939	0,8596	0,9353
NBB	1	0,8750	0,8783	0,8750	0,8758	0,8771	0,8490	0,9791	0,8066	0,9017
	2	0,8750	0,8783	0,8750	0,8758	0,8771	0,8490	0,9764	0,8066	0,9008
	3	0,8750	0,8783	0,8750	0,8758	0,8771	0,8490	0,9764	0,8066	0,9008
	4	0,8750	0,8783	0,8750	0,8758	0,8771	0,8490	0,9790	0,8066	0,9017
	5	0,8750	0,8783	0,8750	0,8758	0,8771	0,8490	0,9772	0,8066	0,9011
Decision Tree	1	0,9250	0,9254	0,9250	0,9242	0,9247	0,9003	0,9454	0,8835	0,9235
	2	0,8750	0,8813	0,8750	0,8690	0,8737	0,8262	0,9764	0,8101	0,8921
	3	0,8000	0,8071	0,8000	0,8029	0,8053	0,7564	0,9483	0,6903	0,8367
	4	0,9250	0,9438	0,9250	0,9255	0,9342	0,9231	0,9632	0,8921	0,9402
	5	0,8500	0,8500	0,8500	0,8500	0,8500	0,8120	0,9809	0,7661	0,8810
Random Forest	1	0,9000	0,9029	0,9000	0,8973	0,8996	0,8632	0,9808	0,8462	0,9145
	2	0,9000	0,9000	0,9000	0,9000	0,9000	0,8746	0,9878	0,8441	0,9208
	3	0,9000	0,9029	0,9000	0,8973	0,8996	0,8632	0,9887	0,8462	0,9172
	4	0,9000	0,9029	0,9000	0,8973	0,8996	0,8632	0,9787	0,8462	0,9138
	5	0,9000	0,9091	0,9000	0,9008	0,9049	0,8860	0,9838	0,8486	0,9249

Continuação

Modelo	T	Ac	Pr	Rec	F_1	F_β	B_S	ROC_S	MCC	F_M
<i>Extra Tree</i>	1	0,8750	0,8866	0,8750	0,8749	0,8804	0,8447	0,9483	0,8101	0,8912
	2	0,9250	0,9279	0,9250	0,9255	0,9267	0,9117	0,9834	0,8844	0,9406
	3	0,9250	0,9254	0,9250	0,9242	0,9247	0,9003	0,9856	0,8835	0,9369
	4	0,8750	0,8813	0,8750	0,8690	0,8737	0,8262	0,9760	0,8101	0,8920
	5	0,9000	0,9091	0,9000	0,9008	0,9049	0,8860	0,9896	0,8486	0,9268
KNN	1	0,9000	0,9113	0,9000	0,8964	0,9025	0,8632	0,9770	0,8475	0,9143
	2	0,9000	0,8982	0,9000	0,8981	0,8979	0,8746	0,9241	0,8435	0,8989
	3	0,9000	0,8982	0,9000	0,8981	0,8979	0,8746	0,9241	0,8435	0,8989
	4	0,7750	0,7623	0,7750	0,7667	0,7636	0,7236	0,9696	0,6463	0,8189
	5	0,8250	0,8201	0,8250	0,8218	0,8206	0,7863	0,9777	0,7265	0,8615
RNN	1	0,8000	0,8762	0,8000	0,7846	0,8175	0,7407	0,9084	0,7225	0,8222
	2	0,8750	0,8951	0,8750	0,8675	0,8781	0,8262	0,9690	0,8120	0,8911
	3	0,8750	0,8951	0,8750	0,8675	0,8781	0,8262	0,9690	0,8120	0,8911
	4	0,8250	0,8201	0,8250	0,8218	0,8206	0,7863	0,9723	0,7265	0,8597
	5	0,8250	0,8201	0,8250	0,8218	0,8206	0,7863	0,9741	0,7265	0,8604
Adap. Boosting	1	0,8500	0,8490	0,8500	0,8460	0,8469	0,8006	0,9060	0,7673	0,8512
	2	0,8500	0,8490	0,8500	0,8460	0,8469	0,8006	0,9060	0,7673	0,8512
	3	0,8500	0,8490	0,8500	0,8460	0,8469	0,8006	0,9060	0,7673	0,8512
	4	0,8500	0,8490	0,8500	0,8460	0,8469	0,8006	0,9060	0,7673	0,8512
	5	0,8500	0,8490	0,8500	0,8460	0,8469	0,8006	0,9060	0,7673	0,8512
Gradient Boosting	1	0,9000	0,9029	0,9000	0,8973	0,8996	0,8632	0,9853	0,8462	0,9160
	2	0,9000	0,9029	0,9000	0,8973	0,8996	0,8632	0,9731	0,8462	0,9120
	3	0,9250	0,9254	0,9250	0,9242	0,9247	0,9003	0,9896	0,8835	0,9382
	4	0,9000	0,9029	0,9000	0,8973	0,8996	0,8632	0,9670	0,8462	0,9099
	5	0,8750	0,8813	0,8750	0,8690	0,8737	0,8262	0,9836	0,8101	0,8945
SVM - Linear	1	0,8750	0,8813	0,8750	0,8690	0,8737	0,8262	0,9775	0,8101	0,8925
	2	0,9500	0,9500	0,9500	0,9500	0,9500	0,9373	0,9819	0,9220	0,9564
	3	0,9500	0,9500	0,9500	0,9500	0,9500	0,9373	0,9811	0,9220	0,9561
	4	0,9250	0,9254	0,9250	0,9242	0,9247	0,9003	0,9809	0,8835	0,9353
	5	0,9000	0,9000	0,9000	0,9000	0,9000	0,8746	0,9719	0,8441	0,9155
SVM - Polinomial	1	0,9250	0,9254	0,9250	0,9242	0,9247	0,9003	0,9740	0,8835	0,9330
	2	0,9000	0,9029	0,9000	0,8973	0,8996	0,8632	0,9827	0,8462	0,9152
	3	0,9000	0,9029	0,9000	0,8973	0,8996	0,8632	0,9827	0,8462	0,9152
	4	0,9500	0,9500	0,9500	0,9500	0,9500	0,9373	0,9896	0,9220	0,9590
	5	0,9000	0,9029	0,9000	0,8973	0,8996	0,8632	0,9835	0,8462	0,9154
SVM - RBF	1	0,9000	0,9029	0,9000	0,8973	0,8996	0,8632	0,9844	0,8462	0,9157
	2	0,8750	0,8813	0,8750	0,8690	0,8737	0,8262	0,9791	0,8101	0,8930
	3	0,8750	0,8813	0,8750	0,8690	0,8737	0,8262	0,9782	0,8101	0,8927
	4	0,9250	0,9254	0,9250	0,9242	0,9247	0,9003	0,9913	0,8835	0,9388
	5	0,9250	0,9279	0,9250	0,9255	0,9267	0,9117	0,9931	0,8844	0,9438
SVM - Sigmoidal	1	0,8750	0,8813	0,8750	0,8690	0,8737	0,8262	0,9774	0,8101	0,8924
	2	0,8750	0,8741	0,8750	0,8736	0,8737	0,8376	0,9603	0,8051	0,8905
	3	0,8750	0,8741	0,8750	0,8736	0,8737	0,8376	0,9620	0,8051	0,8911
	4	0,9250	0,9279	0,9250	0,9255	0,9267	0,9117	0,9880	0,8844	0,9421
	5	0,8250	0,8412	0,8250	0,8261	0,8334	0,7977	0,9592	0,7366	0,8635

Continuação

Modelo	T	Ac	Pr	Rec	F_1	F_β	B_S	ROC_S	MCC	F_M
RNA - 1 Camada	1	0,9250	0,9254	0,9250	0,9242	0,9247	0,9003	0,9808	0,8835	0,9353
	2	0,9250	0,9254	0,9250	0,9242	0,9247	0,9003	0,9816	0,8835	0,9355
	3	0,9000	0,9029	0,9000	0,8973	0,8996	0,8632	0,9799	0,8462	0,9142
	4	0,8750	0,8745	0,8750	0,8714	0,8723	0,8376	0,9782	0,8051	0,8960
	5	0,8750	0,8745	0,8750	0,8714	0,8723	0,8376	0,9807	0,8051	0,8969
RNA - 2 Camadas	1	0,9250	0,9391	0,9250	0,9214	0,9289	0,8889	0,9852	0,8897	0,9343
	2	0,9000	0,9029	0,9000	0,8973	0,8996	0,8632	0,9800	0,8462	0,9143
	3	0,9000	0,9029	0,9000	0,8973	0,8996	0,8632	0,9790	0,8462	0,9139
	4	0,9500	0,9567	0,9500	0,9487	0,9522	0,9259	0,9861	0,9251	0,9548
	5	0,9250	0,9254	0,9250	0,9242	0,9247	0,9003	0,9887	0,8835	0,9379
RNA - 3 Camadas	1	0,9250	0,9391	0,9250	0,9214	0,9289	0,8889	0,9818	0,8897	0,9332
	2	0,9250	0,9254	0,9250	0,9242	0,9247	0,9003	0,9861	0,8835	0,9370
	3	0,9500	0,9567	0,9500	0,9487	0,9522	0,9259	0,9843	0,9251	0,9541
	4	0,9000	0,9000	0,9000	0,9000	0,9000	0,8746	0,9887	0,8441	0,9211
	5	0,9000	0,9000	0,9000	0,9000	0,9000	0,8746	0,9714	0,8441	0,9153

Fonte: Autor.

Legenda: Ac = Acurácia / Pr = Precisão / Rec = Revocação / F_1 = Medida F para $\beta = 1$ / F_β = Medida F / B_S = Acurácia Balanceada / ROC_S Pontuação ROC / MCC = Coeficiente de Correlação de Matthews / F_M Métrica Final.

Tabela 12 – Resultados obtidos para amostra 2 teste binário

Modelo	T	Ac	Pr	Rec	F_1	F_β	B_S	ROC_S	MCC	F_M
Regressão Logística	1	0,9000	0,7273	0,8889	0,8000	0,7547	0,8961	0,9498	0,7408	0,8669
	2	0,8750	0,7000	0,7778	0,7368	0,7143	0,8405	0,9211	0,6567	0,8253
	3	0,8750	0,6667	0,8889	0,7619	0,7018	0,8799	0,9140	0,6924	0,8319
	4	0,8750	0,7000	0,7778	0,7368	0,7143	0,8405	0,9140	0,6567	0,8229
	5	0,9000	0,6923	1,0000	0,8182	0,7377	0,9355	0,9606	0,7765	0,8779
N BG	1	0,8250	0,5714	0,8889	0,6957	0,6154	0,8477	0,9068	0,6088	0,7900
	2	0,8250	0,5833	0,7778	0,6667	0,6140	0,8082	0,9462	0,5618	0,7895
	3	0,8250	0,5625	1,0000	0,7200	0,6164	0,8871	0,9032	0,6599	0,8023
	4	0,8500	0,6000	1,0000	0,7500	0,6522	0,9032	0,8925	0,6956	0,8160
	5	0,8500	0,6000	1,0000	0,7500	0,6522	0,9032	0,8925	0,6956	0,8160
NBB	1	0,7750	0,0000	0,0000	0,0000	0,0000	0,5000	0,5000	0,0000	0,3333
	2	0,8250	0,5714	0,8889	0,6957	0,6154	0,8477	0,9158	0,6088	0,7929
	3	0,8750	0,6429	1,0000	0,7826	0,6923	0,9194	0,9427	0,7343	0,8514
	4	0,8500	0,6000	1,0000	0,7500	0,6522	0,9032	0,9140	0,6956	0,8231
	5	0,8500	0,6000	1,0000	0,7500	0,6522	0,9032	0,9462	0,6956	0,8339

Continuação

Modelo	T	Ac	Pr	Rec	F_1	F_β	B_S	ROC_S	MCC	F_M
<i>Decision Tree</i>	1	0,7750	0,5000	0,6667	0,5714	0,5263	0,7366	0,7366	0,4311	0,6665
	2	0,9250	0,5000	1,0000	0,8571	0,7895	0,9516	0,9409	0,8231	0,8940
	3	0,8250	0,5833	0,7778	0,6667	0,6140	0,8082	0,9301	0,5618	0,7841
	4	0,8250	0,6250	0,5556	0,5882	0,6098	0,7294	0,9283	0,4789	0,7558
	5	0,7750	0,5000	0,6667	0,5714	0,5263	0,7366	0,7168	0,4311	0,6599
<i>Random Forest</i>	1	0,9250	0,8000	0,8889	0,8421	0,8163	0,9122	0,9606	0,7950	0,8963
	2	0,8750	0,7000	0,7778	0,7368	0,7143	0,8405	0,9552	0,6567	0,8367
	3	0,9250	0,8750	0,7778	0,8235	0,8537	0,8728	0,9624	0,7783	0,8964
	4	0,8750	0,6667	0,8889	0,7619	0,7018	0,8799	0,9480	0,6924	0,8432
	5	0,9000	0,6923	1,0000	0,8182	0,7377	0,9355	0,9570	0,7765	0,8767
<i>Extra Tree</i>	1	0,8750	0,7500	0,6667	0,7059	0,7317	0,8011	0,9355	0,6286	0,8228
	2	0,8750	0,7000	0,7778	0,7368	0,7143	0,8405	0,9050	0,6567	0,8199
	3	0,8500	0,6667	0,6667	0,6667	0,6667	0,7849	0,9409	0,5699	0,7975
	4	0,8750	0,7500	0,6667	0,7059	0,7317	0,8011	0,9455	0,6286	0,8318
	5	0,8500	0,6667	0,6667	0,6667	0,6667	0,7849	0,9265	0,5699	0,7927
<i>KNN</i>	1	0,8750	0,8333	0,5556	0,6667	0,7576	0,7616	0,8262	0,6120	0,7818
	2	0,9000	0,8571	0,6667	0,7500	0,8108	0,8172	0,8244	0,6972	0,8175
	3	0,8250	0,6667	0,4444	0,5333	0,6061	0,6900	0,9337	0,4443	0,7432
	4	0,8250	0,6000	0,6667	0,6316	0,6122	0,7688	0,8065	0,5185	0,7292
	5	0,8250	0,6000	0,6667	0,6316	0,6122	0,7688	0,9176	0,5185	0,7662
<i>RNN</i>	1	0,8750	0,8333	0,5556	0,6667	0,7576	0,7616	0,9158	0,6120	0,8117
	2	0,8750	0,8333	0,5556	0,6667	0,7576	0,7616	0,9158	0,6120	0,8117
	3	0,7750	0,0000	0,0000	0,0000	0,0000	0,5000	0,5000	0,0000	0,3333
	4	0,8750	0,8333	0,5556	0,6667	0,7576	0,7616	0,9104	0,6120	0,8099
	5	0,7750	0,0000	0,0000	0,0000	0,0000	0,5000	0,5000	0,0000	0,3333
<i>Adap. Boosting</i>	1	0,9250	0,8000	0,8889	0,8421	0,8163	0,9122	0,9570	0,7950	0,8952
	2	0,8500	0,6667	0,6667	0,6667	0,6667	0,7849	0,9211	0,5699	0,7909
	3	0,8250	0,5833	0,7778	0,6667	0,6140	0,8082	0,9158	0,5618	0,7793
	4	0,8500	0,6364	0,7778	0,7000	0,6604	0,8244	0,9355	0,6067	0,8067
	5	0,9000	0,7273	0,8889	0,8000	0,7547	0,8961	0,9355	0,7408	0,8621
<i>Gradient Boosting</i>	1	0,8500	0,6364	0,7778	0,7000	0,6604	0,8244	0,9032	0,6067	0,7960
	2	0,7750	0,5000	0,5556	0,5263	0,5102	0,6971	0,7724	0,3802	0,6599
	3	0,8500	0,6364	0,7778	0,7000	0,6604	0,8244	0,9194	0,6067	0,8014
	4	0,8250	0,5833	0,7778	0,6667	0,6140	0,8082	0,9140	0,5618	0,7788
	5	0,8500	0,6364	0,7778	0,7000	0,6604	0,8244	0,9355	0,6067	0,8067
<i>SVM - Linear</i>	1	0,8500	0,6364	0,7778	0,7000	0,6604	0,8244	0,9032	0,6067	0,7960
	2	0,8250	0,6250	0,5556	0,5882	0,6098	0,7294	0,8244	0,4789	0,7212
	3	0,9500	0,8182	1,0000	0,9000	0,8491	0,9677	0,9642	0,8749	0,9270
	4	0,8750	0,6667	0,8889	0,7619	0,7018	0,8799	0,9462	0,6924	0,8426
	5	0,9500	0,8182	1,0000	0,9000	0,8491	0,9677	0,9642	0,8749	0,9270
<i>SVM - Polinomial</i>	1	0,8750	0,8333	0,5556	0,6667	0,7576	0,7616	0,9211	0,6120	0,8135
	2	0,8750	0,7500	0,6667	0,7059	0,7317	0,8011	0,9570	0,6286	0,8299
	3	0,8750	0,7000	0,7778	0,7368	0,7143	0,8405	0,9211	0,6567	0,8253
	4	0,8750	0,6667	0,8889	0,7619	0,7018	0,8799	0,9462	0,6924	0,8426
	5	0,9250	0,8000	0,8889	0,8421	0,8163	0,9122	0,9355	0,7950	0,8880

Continuação

Modelo	T	Ac	Pr	Rec	F_1	F_β	B_S	ROC_S	MCC	F_M
SVM - RBF	1	0,8750	0,8333	0,5556	0,6667	0,7576	0,7616	0,7993	0,6120	0,7728
	2	0,8500	0,6667	0,6667	0,6667	0,6667	0,7849	0,9176	0,5699	0,7897
	3	0,9000	0,7778	0,7778	0,7778	0,7778	0,8566	0,9498	0,7133	0,8614
	4	0,8500	0,6667	0,6667	0,6667	0,6667	0,7849	0,9462	0,5699	0,7993
	5	0,9000	0,7273	0,8889	0,8000	0,7547	0,8961	0,9427	0,7408	0,8645
SVM - Sigmoidal	1	0,7750	0,0000	0,0000	0,0000	0,0000	0,5000	0,7527	0,0000	0,4176
	2	0,7750	0,0000	0,0000	0,0000	0,0000	0,5000	0,8100	0,0000	0,4367
	3	0,8500	0,6364	0,7778	0,7000	0,6604	0,8244	0,9247	0,6067	0,8032
	4	0,7750	0,0000	0,0000	0,0000	0,0000	0,5000	0,5000	0,0000	0,3333
	5	0,8500	0,6154	0,8889	0,7273	0,6557	0,8638	0,8925	0,6487	0,8040
RNA - 1 Camada	1	0,8500	0,6364	0,7778	0,7000	0,6604	0,8244	0,8996	0,6067	0,7948
	2	0,8500	0,6364	0,7778	0,7000	0,6604	0,8244	0,9211	0,6067	0,8020
	3	0,9000	0,7273	0,8889	0,8000	0,7547	0,8961	0,9534	0,7408	0,8681
	4	0,7500	0,4737	1,0000	0,6429	0,5294	0,8387	0,8548	0,5665	0,7410
	5	0,9250	0,7500	1,0000	0,8571	0,7895	0,9516	0,9570	0,8231	0,8994
RNA - 2 Camadas	1	0,7750	0,5000	0,8889	0,6400	0,5479	0,8154	0,8548	0,5377	0,7394
	2	0,8750	0,6667	0,8889	0,7619	0,7018	0,8799	0,9283	0,6924	0,8367
	3	0,8750	0,6667	0,8889	0,7619	0,7018	0,8799	0,9319	0,6924	0,8379
	4	0,8500	0,8000	0,4444	0,5714	0,6897	0,7061	0,9462	0,5204	0,7807
	5	0,9000	0,7273	0,8889	0,8000	0,7547	0,8961	0,9606	0,7408	0,8704
RNA - 3 Camadas	1	0,8750	0,7500	0,6667	0,7059	0,7317	0,8011	0,9211	0,6286	0,8180
	2	0,8500	0,6364	0,7778	0,7000	0,6604	0,8244	0,9211	0,6067	0,8020
	3	0,9000	0,7273	0,8889	0,8000	0,7547	0,8961	0,9534	0,7408	0,8681
	4	0,9000	0,6923	1,0000	0,8182	0,7377	0,9355	0,9355	0,7765	0,8696
	5	0,9000	0,7273	0,8889	0,8000	0,7547	0,8961	0,9462	0,7408	0,8657

Fonte: Autor.

Legenda: Ac = Acurácia / Pr = Precisão / Rec = Revocação / F_1 = Medida F para $\beta = 1$ / F_β = Medida F / B_S = Acurácia Balanceada / ROC_S Pontuação ROC / MCC Coeficiente de Correlação de Matthews / F_M Métrica Final.

Tabela 13 – Resultados obtidos para amostra 2 teste multiclasse

Modelo	T	Ac	Pr	Rec	F_1	F_β	B_S	ROC_S	MCC	F_M
Regressão Logística	1	0,6500	0,6379	0,6500	0,6404	0,6381	0,6282	0,7972	0,4472	0,6878
	2	0,6000	0,5662	0,6000	0,5763	0,5686	0,6140	0,7454	0,3853	0,6427
	3	0,5750	0,5683	0,5750	0,5711	0,5693	0,5655	0,7819	0,3401	0,6389
	4	0,6250	0,6179	0,6250	0,6208	0,6189	0,6282	0,7461	0,4179	0,6644
	5	0,6500	0,6260	0,6500	0,6302	0,6257	0,6581	0,8029	0,4637	0,6956

Continuação

Modelo	T	Ac	Pr	Rec	F_1	F_β	B_S	ROC_S	MCC	F_M
NBG	1	0,6250	0,5975	0,6250	0,5683	0,5676	0,6439	0,8017	0,4482	0,6711
	2	0,5750	0,5259	0,5750	0,5334	0,5232	0,5698	0,8057	0,3449	0,6329
	3	0,6250	0,5975	0,6250	0,5683	0,5676	0,6439	0,8027	0,4482	0,6714
	4	0,5750	0,5259	0,5750	0,5334	0,5232	0,5698	0,8033	0,3449	0,6321
	5	0,5500	0,4753	0,5500	0,5005	0,4818	0,5442	0,7637	0,3103	0,5966
NBB	1	0,6750	0,6797	0,6750	0,6642	0,6706	0,6838	0,7762	0,5155	0,7102
	2	0,6500	0,6932	0,6500	0,6511	0,6721	0,6724	0,7846	0,4957	0,7097
	3	0,6750	0,6624	0,6750	0,6597	0,6594	0,7023	0,8350	0,5097	0,7322
	4	0,6500	0,6575	0,6500	0,6373	0,6462	0,6838	0,7671	0,4865	0,6990
	5	0,6750	0,6703	0,6750	0,6703	0,6698	0,6909	0,8286	0,5010	0,7297
Decision Tree	1	0,5000	0,6151	0,5000	0,5171	0,5613	0,5242	0,6398	0,2775	0,5751
	2	0,4750	0,5379	0,4750	0,4866	0,5112	0,4872	0,6817	0,2223	0,5600
	3	0,7500	0,7570	0,7500	0,7503	0,7534	0,7308	0,8585	0,6070	0,7809
	4	0,5750	0,6112	0,5750	0,5823	0,5966	0,5983	0,7775	0,3645	0,6575
	5	0,5750	0,5946	0,5750	0,5780	0,5862	0,5983	0,7026	0,3590	0,6290
Random Forest	1	0,6250	0,6429	0,6250	0,6252	0,6337	0,6538	0,8274	0,4383	0,7050
	2	0,7250	0,7196	0,7250	0,7202	0,7193	0,7165	0,8463	0,5724	0,7607
	3	0,5500	0,5500	0,5500	0,5442	0,5464	0,5840	0,7452	0,3190	0,6252
	4	0,5750	0,5515	0,5750	0,5520	0,5496	0,6140	0,8153	0,3626	0,6596
	5	0,8250	0,8546	0,8250	0,8257	0,8401	0,8348	0,8993	0,7484	0,8581
Extra Tree	1	0,6000	0,5980	0,6000	0,5962	0,5966	0,6282	0,8317	0,3885	0,6855
	2	0,6750	0,7277	0,6750	0,6825	0,7041	0,6937	0,8559	0,5251	0,7512
	3	0,6750	0,6624	0,6750	0,6597	0,6594	0,7023	0,8021	0,5097	0,7213
	4	0,7000	0,7031	0,7000	0,7008	0,7020	0,6980	0,7854	0,5368	0,7285
	5	0,6500	0,6815	0,6500	0,6577	0,6700	0,6681	0,8449	0,4718	0,7276
KNN	1	0,6000	0,6129	0,6000	0,6030	0,6079	0,5983	0,7272	0,3666	0,6445
	2	0,6250	0,6853	0,6250	0,6363	0,6611	0,6382	0,7315	0,4413	0,6769
	3	0,6000	0,6079	0,6000	0,6013	0,6045	0,5798	0,8300	0,3694	0,6714
	4	0,5000	0,5088	0,5000	0,5008	0,5047	0,5171	0,7271	0,2390	0,5830
	5	0,6000	0,6262	0,6000	0,6010	0,6129	0,6353	0,8056	0,4052	0,6846
RNN	1	0,3250	0,1056	0,3250	0,1594	0,1221	0,3333	0,5000	0,0000	0,3185
	2	0,5000	0,5234	0,5000	0,5091	0,5171	0,5171	0,7177	0,2263	0,5840
	3	0,5000	0,4665	0,5000	0,4528	0,4557	0,4416	0,8158	0,2533	0,5711
	4	0,5000	0,5042	0,5000	0,5007	0,5025	0,5171	0,6732	0,2314	0,5642
	5	0,6000	0,6158	0,6000	0,5914	0,6022	0,6467	0,7843	0,4121	0,6778
Adap. Boosting	1	0,5500	0,6220	0,5500	0,5636	0,5929	0,5570	0,7716	0,3295	0,6405
	2	0,6750	0,6813	0,6750	0,6773	0,6795	0,6610	0,8356	0,4917	0,7253
	3	0,6750	0,7131	0,6750	0,6773	0,6935	0,6382	0,8494	0,4982	0,7270
	4	0,6250	0,5841	0,6250	0,5950	0,5857	0,5954	0,7425	0,4149	0,6412
	5	0,5750	0,5677	0,5750	0,5666	0,5663	0,6026	0,7605	0,3535	0,6431
Gradient Boosting	1	0,6000	0,5848	0,6000	0,5910	0,5869	0,5840	0,7464	0,3767	0,6391
	2	0,6500	0,6713	0,6500	0,6545	0,6632	0,6538	0,7673	0,4723	0,6948
	3	0,5750	0,5920	0,5750	0,5794	0,5860	0,5798	0,7802	0,3542	0,6487
	4	0,5500	0,5821	0,5500	0,5469	0,5630	0,5912	0,7784	0,3407	0,6442
	5	0,6250	0,6329	0,6250	0,6244	0,6284	0,6538	0,6783	0,4310	0,6535

Continuação

Modelo	T	Ac	Pr	Rec	F_1	F_β	B_S	ROC_S	MCC	F_M
SVM - Linear	1	0,4750	0,4829	0,4750	0,4675	0,4735	0,5242	0,6904	0,2162	0,5627
	2	0,4750	0,4829	0,4750	0,4675	0,4735	0,5242	0,6861	0,2162	0,5613
	3	0,6750	0,6704	0,6750	0,6722	0,6710	0,6909	0,7780	0,4912	0,7133
	4	0,5250	0,5593	0,5250	0,5061	0,5261	0,5983	0,7088	0,3214	0,6111
	5	0,6750	0,6704	0,6750	0,6722	0,6710	0,6909	0,7913	0,4912	0,7177
SVM - Polinomial	1	0,6000	0,6164	0,6000	0,6067	0,6121	0,5983	0,7799	0,3779	0,6635
	2	0,6000	0,5869	0,6000	0,5902	0,5875	0,5840	0,7642	0,3671	0,6452
	3	0,7000	0,7031	0,7000	0,7008	0,7020	0,6980	0,8362	0,5368	0,7454
	4	0,6250	0,5313	0,6250	0,5525	0,5297	0,6182	0,6983	0,4333	0,6154
	5	0,6250	0,6154	0,6250	0,6195	0,6169	0,6467	0,7841	0,4179	0,6826
SVM - RBF	1	0,5750	0,6531	0,5750	0,5842	0,6176	0,6011	0,6194	0,3786	0,6127
	2	0,5750	0,6531	0,5750	0,5842	0,6176	0,6011	0,6242	0,3786	0,6143
	3	0,6000	0,6268	0,6000	0,6062	0,6166	0,6168	0,8114	0,3974	0,6816
	4	0,6000	0,6291	0,6000	0,6028	0,6154	0,6239	0,7655	0,4052	0,6683
	5	0,6750	0,6856	0,6750	0,6784	0,6823	0,6795	0,8230	0,5015	0,7283
SVM - Sigmoidal	1	0,3250	0,1056	0,3250	0,1594	0,1221	0,3333	0,0,5	0,0000	0,3185
	2	0,4250	0,5674	0,4250	0,3361	0,3993	0,4074	0,6829	0,2575	0,4965
	3	0,6750	0,6912	0,6750	0,6812	0,6868	0,6610	0,8091	0,5015	0,7189
	4	0,3250	0,1056	0,3250	0,1594	0,1221	0,3333	0,5000	0,0000	0,3185
	5	0,6750	0,6929	0,6750	0,6696	0,6805	0,7094	0,8206	0,5226	0,7368
RNA - 1 Camada	1	0,5750	0,5552	0,5750	0,5571	0,5544	0,6211	0,7190	0,3577	0,6315
	2	0,5500	0,5658	0,5500	0,5546	0,5605	0,5726	0,7803	0,3131	0,6378
	3	0,6500	0,6389	0,6500	0,6319	0,6326	0,6396	0,7532	0,4695	0,6751
	4	0,5500	0,5487	0,5500	0,5476	0,5479	0,5470	0,7275	0,3079	0,6075
	5	0,7000	0,6979	0,7000	0,6939	0,6952	0,7094	0,8277	0,5435	0,7441
RNA - 2 Camadas	1	0,5750	0,6300	0,5750	0,5813	0,6050	0,5983	0,7492	0,3832	0,6508
	2	0,5250	0,5563	0,5250	0,5309	0,5434	0,5541	0,7334	0,2875	0,6103
	3	0,6750	0,6800	0,6750	0,6756	0,6778	0,6980	0,7997	0,5015	0,7252
	4	0,5000	0,5156	0,5000	0,5040	0,5100	0,5100	0,7322	0,2390	0,5841
	5	0,6500	0,6356	0,6500	0,6411	0,6374	0,6652	0,8004	0,4550	0,7010
RNA - 3 Camadas	1	0,5500	0,5254	0,5500	0,5244	0,5218	0,5769	0,6766	0,3193	0,5918
	2	0,6000	0,4219	0,6000	0,4828	0,4420	0,5556	0,7044	0,4305	0,5673
	3	0,6750	0,6625	0,6750	0,6627	0,6612	0,6838	0,7923	0,5034	0,7124
	4	0,5250	0,3833	0,5250	0,4430	0,4051	0,5185	0,6237	0,2646	0,5158
	5	0,5750	0,5486	0,5750	0,5585	0,5517	0,5769	0,7738	0,3415	0,6342

Fonte: Autor.

Legenda: Ac = Acurácia / Pr = Precisão / Rec = Revocação / F_1 = Medida F para $\beta = 1$ / F_β = Medida F / B_S = Acurácia Balanceada / ROC_S Pontuação ROC / MCC Coeficiente de Correlação de Matthews / F_M Métrica Final.

Tabela 14 – Resultados obtidos para amostra 3 teste binário

Modelo	T	Ac	Pr	Rec	F_1	F_β	B_S	ROC_S	MCC	F_M
Regressão Logística	1	0,9500	0,8182	1,0000	0,9000	0,8491	0,9677	0,9642	0,8749	0,9270
	2	0,9250	0,8000	0,8889	0,8421	0,8163	0,9122	0,9462	0,7950	0,8916
	3	0,9250	0,8000	0,8889	0,8421	0,8163	0,9122	0,9749	0,7950	0,9011
	4	0,9500	0,8182	1,0000	0,9000	0,4849	0,9677	0,9677	0,8749	0,9282
	5	0,9000	0,7778	0,7778	0,7778	0,7778	0,8566	0,9427	0,7133	0,8590
NBG	1	0,8250	0,5714	0,0000	0,6957	0,6154	0,8477	0,9068	0,6088	0,7900
	2	0,8250	0,5833	0,7778	0,6667	0,6140	0,8082	0,9462	0,5618	0,7895
	3	0,9000	0,6923	1,0000	0,8182	0,7377	0,9355	0,9355	0,7765	0,8696
	4	0,8500	0,6154	0,8889	0,7273	0,6557	0,8638	0,9427	0,6487	0,8207
	5	0,9000	0,6923	1,0000	0,8182	0,7377	0,9355	0,9755	0,7765	0,8796
NBB	1	0,9000	1,0000	0,5556	0,7143	0,8621	0,7778	0,9570	0,7015	0,8656
	2	0,9000	0,6923	1,0000	0,8182	0,7377	0,9355	0,9462	0,7765	0,8731
	3	0,8750	0,6429	1,0000	0,7826	0,6923	0,9194	0,9534	0,7343	0,8550
	4	0,8750	0,6429	1,0000	0,7826	0,6923	0,9194	0,9462	0,7343	0,8526
	5	0,8750	0,6429	1,0000	0,7826	0,6923	0,9194	0,9373	0,7343	0,8496
Decision Tree	1	0,8500	0,6364	0,7778	0,7000	0,6604	0,8244	0,8244	0,6067	0,7697
	2	0,9250	0,7500	1,0000	0,8571	0,7895	0,9516	0,9695	0,8231	0,9035
	3	0,9000	0,6923	1,0000	0,8182	0,7377	0,9355	0,9444	0,7765	0,8725
	4	0,8250	0,6667	0,4444	0,5333	0,6061	0,6900	0,8172	0,4443	0,7044
	5	0,8750	0,7500	0,6667	0,7059	0,7317	0,8011	0,9158	0,6286	0,8162
Random Forest	1	0,9000	0,7273	0,8889	0,8000	0,7547	0,8961	0,9534	0,7408	0,8681
	2	0,8750	0,7000	0,7778	0,7368	0,7143	0,8405	0,9570	0,6567	0,8373
	3	0,9250	0,8750	0,7778	0,8235	0,8537	0,8728	0,9588	0,7783	0,8951
	4	0,8750	0,7000	0,7778	0,7368	0,7143	0,8405	0,9570	0,6567	0,8373
	5	0,8750	0,7000	0,7778	0,7368	0,7143	0,8405	0,9516	0,6567	0,8355
Extra Tree	1	0,8500	0,6364	0,7778	0,7000	0,6604	0,8244	0,9355	0,6067	0,8067
	2	0,9000	0,7778	0,7778	0,7778	0,7778	0,8566	0,9480	0,7133	0,8608
	3	0,8750	0,7000	0,7778	0,7368	0,7143	0,8405	0,9462	0,6567	0,8337
	4	0,8750	0,6667	0,8889	0,7619	0,7018	0,8799	0,9301	0,6924	0,8373
	5	0,9000	0,8571	0,6667	0,7500	0,8108	0,8172	0,9606	0,6972	0,8629
KNN	1	0,8750	0,8333	0,5556	0,6667	0,7576	0,7616	0,8262	0,6120	0,7818
	2	0,9000	0,8571	0,6667	0,7500	0,8108	0,8172	0,8244	0,6972	0,8175
	3	0,8500	0,7143	0,5556	0,6250	0,6757	0,7455	0,9373	0,5397	0,7862
	4	0,7500	0,4444	0,4444	0,4444	0,4444	0,6416	0,6416	0,2832	0,5759
	5	0,8750	0,7000	0,7778	0,7368	0,7143	0,8405	0,8405	0,6567	0,7984
RNN	1	0,8750	0,8333	0,5556	0,6667	0,7576	0,7616	0,9158	0,6120	0,8117
	2	0,8750	0,8333	0,5556	0,6667	0,7576	0,7616	0,9158	0,6120	0,8117
	3	0,7750	0,0000	0,0000	0,0000	0,0000	0,5000	0,5000	0,0000	0,3333
	4	0,9000	0,8571	0,6667	0,7500	0,8108	0,8172	0,9140	0,6972	0,8473
	5	0,7750	0,0000	0,0000	0,0000	0,0000	0,5000	0,5000	0,0000	0,3333

Continuação

Modelo	T	<i>Ac</i>	<i>Pr</i>	<i>Rec</i>	<i>F</i> ₁	<i>F</i> _β	<i>B</i> _S	<i>ROC</i> _S	<i>MCC</i>	<i>F</i> _M
Adap. Boosting	1	0,8750	0,7000	0,7778	0,7368	0,7143	0,8405	0,9534	0,6567	0,8361
	2	0,8750	0,7000	0,7778	0,7368	0,7143	0,8405	0,9462	0,6567	0,8337
	3	0,8750	0,7500	0,6667	0,7059	0,7317	0,8011	0,9355	0,6286	0,8228
	4	0,8750	0,7000	0,7778	0,7368	0,7143	0,8405	0,9086	0,6567	0,8211
	5	0,9500	0,8889	0,8889	0,8889	0,8889	0,9283	0,9570	0,8566	0,9247
Gradient Boosting	1	0,9000	0,7778	0,7778	0,7778	0,7778	0,8566	0,9391	0,7133	0,8578
	2	0,8250	0,6000	0,6667	0,6316	0,6122	0,7688	0,8996	0,5185	0,7602
	3	0,8750	0,7500	0,6667	0,7059	0,7317	0,8011	0,9283	0,6286	0,8204
	4	0,8750	0,7500	0,6667	0,7059	0,7317	0,8011	0,9176	0,6286	0,8168
	5	0,9000	0,7778	0,7778	0,7778	0,7778	0,8566	0,9462	0,7133	0,8602
SVM - Linear	1	0,8500	0,6364	0,7778	0,7000	0,6604	0,8244	0,9032	0,6067	0,7960
	2	0,8250	0,6250	0,5556	0,5882	0,6098	0,7294	0,8172	0,4789	0,7188
	3	0,9250	0,7500	1,0000	0,8571	0,7895	0,9516	0,9713	0,8231	0,9041
	4	0,8750	0,7000	0,7778	0,7368	0,7143	0,8405	0,8961	0,6567	0,8169
	5	0,9250	0,7500	1,0000	0,8571	0,7895	0,9516	0,9713	0,8231	0,9041
SVM - Polinomial	1	0,8750	0,8333	0,5556	0,6667	0,7576	0,7616	0,9211	0,6120	0,8135
	2	0,9000	0,7778	0,7778	0,0000	0,7778	0,8566	0,9462	0,7133	0,8602
	3	0,8750	0,7000	0,7778	0,7368	0,7143	0,8405	0,9498	0,6567	0,8349
	4	0,6000	0,2667	0,4444	0,3333	0,2899	0,5448	0,4875	0,7729	0,4407
	5	0,9250	0,8000	0,8889	0,8421	0,8163	0,9122	0,9283	0,7950	0,8856
SVM - RBF	1	0,8750	0,8333	0,5556	0,6667	0,7576	0,7616	0,7993	0,6120	0,7728
	2	0,8500	0,6667	0,6667	0,6667	0,6667	0,7849	0,9176	0,5699	0,7897
	3	0,9000	0,7778	0,7778	0,7778	0,7778	0,8566	0,9498	0,7133	0,8614
	4	0,8750	0,7000	0,7778	0,7368	0,7143	0,8405	0,9606	0,6567	0,8385
	5	0,8750	0,7000	0,7778	0,7368	0,7143	0,8405	0,9462	0,6567	0,8337
SVM - Sigmoidal	1	0,7750	0,0000	0,0000	0,0000	0,0000	0,5000	0,7527	0,0000	0,4176
	2	0,7750	0,0000	0,0000	0,0000	0,0000	0,5000	0,8100	0,0000	0,4367
	3	0,8500	0,6364	0,7778	0,7000	0,6604	0,8244	0,9355	0,6067	0,8067
	4	0,7750	0,0000	0,0000	0,0000	0,0000	0,5000	0,5000	0,0000	0,3333
	5	0,8750	0,6429	1,0000	0,7826	0,6923	0,9194	0,9104	0,7343	0,8407
RNA - 1 Camada	1	0,9000	0,7273	0,8889	0,8000	0,7547	0,8961	0,9462	0,7408	0,8657
	2	0,8750	0,6667	0,8889	0,7619	0,7018	0,8799	0,9283	0,6924	0,8367
	3	0,8750	0,6667	0,8889	0,7619	0,7018	0,8799	0,9570	0,6924	0,8462
	4	0,8500	0,6000	1,0000	0,7500	0,6522	0,9032	0,9032	0,6956	0,8195
	5	0,9000	0,7273	0,8889	0,8000	0,7547	0,8961	0,9570	0,7408	0,8693
RNA - 2 Camadas	1	0,8750	0,7000	0,7778	0,7368	0,7143	0,8405	0,9337	0,6567	0,8295
	2	0,8750	0,6429	1,0000	0,7826	0,6923	0,9194	0,9194	0,7343	0,8437
	3	0,9000	0,7273	0,8889	0,8000	0,7547	0,8961	0,9606	0,7408	0,8704
	4	0,9000	0,6923	1,0000	0,8182	0,7377	0,9355	0,9355	0,7765	0,8696
	5	0,9000	0,7273	0,8889	0,8000	0,7547	0,8961	0,9498	0,7408	0,8669

Continuação

Modelo	T	Ac	Pr	Rec	F_1	F_β	B_S	ROC_S	MCC	F_M
RNA - 3 Camadas	1	0,8750	0,6429	1,0000	0,7826	0,6923	0,9194	0,9283	0,7343	0,8467
	2	0,8500	0,6364	0,7778	0,7000	0,6604	0,8244	0,9211	0,6067	0,8020
	3	0,8500	0,6154	0,8889	0,7273	0,6557	0,8638	0,9211	0,6487	0,8136
	4	0,8750	0,6429	1,0000	0,7826	0,6923	0,9194	0,9194	0,7343	0,8437
	5	0,9000	0,7273	0,8889	0,8000	0,7547	0,8961	0,9534	0,7408	0,8681

Fonte: Autor.

Legenda: Ac = Acurácia / Pr = Precisão / Rec = Revocação / F_1 = Medida F para $\beta = 1$ / F_β = Medida F / B_S = Acurácia Balanceada / ROC_S Pontuação ROC / MCC Coeficiente de Correlação de Matthews / F_M Métrica Final.

Tabela 15 – Resultados obtidos para amostra 3 teste multiclasse

Modelo	T	Ac	Pr	Rec	F_1	F_β	B_S	ROC_S	MCC	F_M
Regressão Logística	1	0,6500	0,6379	0,6500	0,6404	0,6381	0,6282	0,7982	0,4472	0,6881
	2	0,8750	0,8825	0,8750	0,8722	0,8766	0,8604	0,9409	0,8112	0,8926
	3	0,9000	0,9091	0,9000	0,9008	0,9049	0,8860	0,9600	0,8486	0,9170
	4	0,8750	0,9071	0,8750	0,8695	0,8857	0,8718	0,9682	0,8239	0,9086
	5	0,8750	0,8783	0,8750	0,8758	0,8771	0,8490	0,9439	0,8066	0,8900
NBG	1	0,8500	0,8717	0,8500	0,8514	0,8616	0,8604	0,9246	0,7808	0,8822
	2	0,8500	0,8717	0,8500	0,8514	0,8616	0,8604	0,9246	0,7808	0,8822
	3	0,9000	0,9308	0,9000	0,9000	0,9145	0,8974	0,9262	0,8596	0,9127
	4	0,8500	0,8717	0,8500	0,8514	0,8616	0,8604	0,9228	0,7808	0,8816
	5	0,9000	0,9308	0,9000	0,9000	0,9145	0,8974	0,9253	0,8596	0,9124
NBB	1	0,8750	0,8925	0,8750	0,8758	0,8838	0,8604	0,9222	0,8144	0,8888
	2	0,8750	0,8783	0,8750	0,8758	0,8771	0,8490	0,9800	0,8066	0,9020
	3	0,7000	0,6901	0,7000	0,6896	0,6888	0,7279	0,8659	0,5435	0,7609
	4	0,8750	0,8741	0,8750	0,8736	0,8737	0,8376	0,9799	0,8051	0,8971
	5	0,7000	0,6901	0,7000	0,6896	0,6888	0,7279	0,8677	0,5435	0,7615
Decision Tree	1	0,9000	0,9000	0,9000	0,9000	0,9000	0,8746	0,9307	0,8441	0,9018
	2	0,8500	0,8490	0,8500	0,8460	0,8469	0,8006	0,9281	0,7673	0,8585
	3	0,7500	0,7477	0,7500	0,7480	0,7476	0,7236	0,8716	0,6073	0,7809
	4	0,8500	0,8490	0,8500	0,8460	0,8469	0,8006	0,9525	0,7673	0,8667
	5	0,9000	0,9000	0,9000	0,9000	0,9000	0,8746	0,9412	0,8441	0,9053
Random Forest	1	0,8750	0,8983	0,8750	0,8756	0,8867	0,8789	0,9424	0,8192	0,9027
	2	0,8750	0,8783	0,8750	0,8758	0,8771	0,8490	0,9612	0,8066	0,8958
	3	0,8250	0,8485	0,8250	0,8269	0,8376	0,8162	0,9531	0,7418	0,8690
	4	0,8750	0,8955	0,8750	0,8794	0,8877	0,8675	0,9367	0,8137	0,8973
	5	0,9000	0,9091	0,9000	0,9008	0,9049	0,8860	0,9562	0,8486	0,9157

Continuação

Modelo	T	Ac	Pr	Rec	F_1	F_β	B_S	ROC_S	MCC	F_M
<i>Extra Tree</i>	1	0,7750	0,7838	0,7750	0,7728	0,7778	0,7536	0,9166	0,6589	0,8160
	2	0,8750	0,8783	0,8750	0,8758	0,8771	0,8490	0,9625	0,8066	0,8962
	3	0,9250	0,9438	0,9250	0,9255	0,9342	0,9231	0,9769	0,8921	0,9447
	4	0,9000	0,9091	0,9000	0,9008	0,9049	0,8860	0,9519	0,8486	0,9143
	5	0,8750	0,8783	0,8750	0,8758	0,8771	0,8490	0,9451	0,8066	0,8904
KNN	1	0,6000	0,6129	0,6000	0,6030	0,6079	0,5983	0,7272	0,3666	0,6445
	2	0,6250	0,6853	0,6250	0,6363	0,6611	0,6382	0,7315	0,4413	0,6769
	3	0,6000	0,6079	0,6000	0,6013	0,6045	0,5798	0,8354	0,3694	0,6732
	4	0,4750	0,4732	0,4750	0,4735	0,4732	0,4915	0,6573	0,1886	0,5407
	5	0,5750	0,6007	0,5750	0,5743	0,5867	0,6097	0,8140	0,3700	0,6701
RNN	1	0,3250	0,1056	0,3250	0,1594	0,1221	0,3333	0,5000	0,0000	0,3185
	2	0,5000	0,5234	0,5000	0,5091	0,5171	0,5171	0,7177	0,2263	0,5840
	3	0,5250	0,4786	0,5250	0,4760	0,4733	0,4601	0,8369	0,2893	0,5901
	4	0,5000	0,5088	0,5000	0,5008	0,5047	0,5171	0,6698	0,2390	0,5639
	5	0,6000	0,6170	0,6000	0,5980	0,6070	0,6282	0,7877	0,4035	0,6743
Adap. Boosting	1	0,8500	0,0,85	0,8500	0,8500	0,8500	0,8120	0,8989	0,7661	0,8536
	2	0,9000	0,9308	0,9000	0,9000	0,9145	0,8974	0,9554	0,8596	0,9224
	3	0,9000	0,9000	0,9000	0,9000	0,9000	0,8746	0,9569	0,8441	0,9105
	4	0,8500	0,8591	0,8500	0,8513	0,8552	0,8234	0,9422	0,7709	0,8736
	5	0,3750	0,2422	0,3750	0,2768	0,2524	0,4416	0,8764	0,1545	0,5235
Gradient Boosting	1	0,9000	0,9000	0,9000	0,9000	0,9000	0,8746	0,9531	0,8441	0,9092
	2	0,8500	0,8571	0,5000	0,8513	0,8543	0,8191	0,9482	0,7686	0,8739
	3	0,8750	0,8741	0,8750	0,8736	0,8737	0,8376	0,9504	0,8051	0,8872
	4	0,8750	0,8783	0,8750	0,8758	0,8771	0,8490	0,9583	0,8066	0,8948
	5	0,9000	0,9000	0,9000	0,9000	0,9000	0,8746	0,9630	0,8441	0,9126
SVM - Linear	1	0,6750	0,6719	0,6750	0,6600	0,6631	0,6652	0,8165	0,5035	0,7150
	2	0,4500	0,4545	0,4500	0,4393	0,4449	0,4986	0,7389	0,1839	0,5608
	3	0,9250	0,9354	0,9250	0,9229	0,9285	0,9231	0,9741	0,8894	0,9419
	4	0,4750	0,5075	0,4750	0,4767	0,4915	0,5128	0,5512	0,2101	0,5185
	5	0,9250	0,9354	0,9250	0,9229	0,9285	0,9231	0,9704	0,8894	0,9407
SVM - Polinomial	1	0,6000	0,6164	0,6000	0,6067	0,6121	0,5983	0,7823	0,3779	0,6642
	2	0,6000	0,5869	0,6000	0,5902	0,5875	0,5840	0,7671	0,3671	0,6462
	3	0,7500	0,7546	0,7500	0,7515	0,7532	0,7422	0,8873	0,6141	0,7942
	4	0,5500	0,5103	0,5500	0,5213	0,5125	0,5513	0,7107	0,3120	0,5915
	5	0,9250	0,9438	0,9250	0,9255	0,9342	0,9231	0,9685	0,8921	0,9420
SVM - RBF	1	0,5750	0,6531	0,5750	0,5842	0,6176	0,6011	0,7390	0,3786	0,6526
	2	0,5750	0,6531	0,5750	0,5842	0,6176	0,6011	0,7376	0,3786	0,6521
	3	0,7250	0,7582	0,7250	0,7331	0,7459	0,7308	0,8991	0,5883	0,7920
	4	0,6000	0,6291	0,6000	0,6028	0,6154	0,6239	0,7519	0,4052	0,6637
	5	0,7250	0,7582	0,7250	0,7331	0,7459	0,7308	0,9009	0,5883	0,7929
SVM - Sigmoidal	1	0,3250	0,1056	0,3250	0,1594	0,1221	0,3333	0,5000	0,0000	0,3185
	2	0,4500	0,5707	0,4500	0,3717	0,4341	0,4259	0,6416	0,2920	0,5005
	3	0,6750	0,7154	0,6750	0,6870	0,7019	0,6681	0,8379	0,5111	0,7360
	4	0,3250	0,1056	0,3250	0,1594	0,1221	0,3333	0,5000	0,0000	0,3185
	5	0,7500	0,7736	0,7500	0,7497	0,7613	0,7792	0,8414	0,6330	0,7939

Continuação

Modelo	T	<i>Ac</i>	<i>Pr</i>	<i>Rec</i>	<i>F</i> ₁	<i>F</i> _{β}	<i>B</i> _S	<i>ROC</i> _S	<i>MCC</i>	<i>F</i> _M
RNA - 1 Camada	1	0,6750	0,6425	0,6750	0,6350	0,6312	0,6695	0,7609	0,5029	0,6872
	2	0,4000	0,2548	0,4000	0,2977	0,2686	0,4786	0,7443	0,2062	0,4972
	3	0,7250	0,7084	0,7250	0,7031	0,7021	0,7023	0,8514	0,5767	0,7519
	4	0,5250	0,5796	0,5250	0,4870	0,5143	0,6125	0,7745	0,3457	0,6338
	5	0,7500	0,7435	0,7500	0,7250	0,7288	0,7208	0,8825	0,6232	0,7774
RNA - 2 Camadas	1	0,4500	0,2132	0,4500	0,2893	0,2382	0,3333	0,5703	0,2532	0,3806
	2	0,6750	0,6259	0,6750	0,5837	0,5687	0,6738	0,7626	0,5394	0,6684
	3	0,7250	0,7096	0,7250	0,7040	0,7033	0,7208	0,8750	0,5792	0,7664
	4	0,6750	0,4562	0,6750	0,5443	0,4877	0,6667	0,7662	0,5623	0,6402
	5	0,8000	0,8010	0,8000	0,7906	0,7940	0,7721	0,8820	0,6957	0,8160
RNA - 3 Camadas	1	0,4500	0,7341	0,4500	0,3687	0,3974	0,5228	0,7415	0,2958	0,5539
	2	0,5500	0,6700	0,5500	0,5170	0,5643	0,6268	0,7825	0,4016	0,6579
	3	0,7750	0,7701	0,7750	0,7670	0,7674	0,7721	0,9420	0,6546	0,8272
	4	0,4000	0,2548	0,4000	0,2977	0,2686	0,4786	0,7348	0,2062	0,4940
	5	0,8000	0,8103	0,8000	0,7947	0,8012	0,7721	0,8981	0,6990	0,8238

Fonte: Autor.

Legenda: Ac = Acurácia / Pr = Precisão / Rec = Revocação / F_1 = Medida F para $\beta = 1$ / F_β = Medida F / B_S = Acurácia Balanceada / ROC_S Pontuação ROC / MCC Coeficiente de Correlação de Matthews / F_M Métrica Final.

Tabela 16 – Resultados obtidos para amostra 4 teste binário

Modelo	T	<i>Ac</i>	<i>Pr</i>	<i>Rec</i>	<i>F</i> ₁	<i>F</i> _{β}	<i>B</i> _S	<i>ROC</i> _S	<i>MCC</i>	<i>F</i> _M
Regressão Logística	1	0,8043	0,5290	0,1825	0,2714	0,3834	0,5710	0,8192	0,2241	0,5912
	2	0,8507	0,6850	0,4675	0,5557	0,6267	0,7069	0,8968	0,4822	0,7435
	3	0,8652	0,7044	0,5600	0,6240	0,6699	0,7507	0,9166	0,5484	0,7791
	4	0,8213	0,5342	0,8200	0,6469	0,5742	0,8208	0,8992	0,5563	0,7647
	5	0,8402	0,5680	0,8350	0,6761	0,6068	0,8383	0,9135	0,5939	0,7862
NBG	1	0,7054	0,3869	0,8125	0,5242	0,4322	0,7456	0,8174	0,3980	0,6651
	2	0,6835	0,3753	0,8800	0,5262	0,4239	0,7572	0,8203	0,4121	0,6671
	3	0,6985	0,3854	0,8575	0,5318	0,4331	0,7581	0,8122	0,4154	0,6678
	4	0,7139	0,3976	0,8400	0,5398	0,4444	0,7612	0,8196	0,4229	0,6751
	5	0,6965	0,3829	0,8500	0,5280	0,4302	0,7541	0,8096	0,4089	0,6646
NBB	1	0,7998	0,4932	0,0900	0,1522	0,2601	0,5335	0,6869	0,1428	0,4935
	2	0,7718	0,4532	0,6900	0,5471	0,4866	0,7411	0,8161	0,4191	0,6813
	3	0,7269	0,4071	0,8050	0,5407	0,4517	0,7562	0,8221	0,4191	0,6767
	4	0,6945	0,3312	0,5200	0,4047	0,3571	0,6290	0,6790	0,2223	0,5551
	5	0,7199	0,4012	0,8175	0,5383	0,4467	0,7565	0,8196	0,4175	0,6743

Continuação

Modelo	T	Ac	Pr	Rec	F ₁	F _β	B _S	ROC _S	MCC	F _M
Decision Tree	1	0,8103	0,5256	0,5125	0,5190	0,5230	0,6985	0,6985	0,4009	0,6400
	2	0,8457	0,6619	0,4650	0,5463	0,6102	0,7029	0,8771	0,4671	0,7301
	3	0,8457	0,6619	0,4650	0,5463	0,6102	0,7029	0,8776	0,4671	0,7302
	4	0,8173	0,5374	0,6100	0,5714	0,5505	0,7395	0,8588	0,4574	0,7163
	5	0,7983	0,4966	0,7375	0,5936	0,5313	0,7755	0,8625	0,4823	0,7231
Random Forest	1	0,8697	0,7860	0,4775	0,5941	0,6961	0,7225	0,9169	0,5449	0,7785
	2	0,8662	0,7598	0,4825	0,5902	0,6815	0,7222	0,9130	0,5340	0,7722
	3	0,8557	0,7160	0,4600	0,5601	0,6443	0,7072	0,9049	0,4954	0,7521
	4	0,8747	0,6945	0,6650	0,6794	0,6884	0,7960	0,9216	0,6018	0,8020
	5	0,8802	0,6970	0,7075	0,7022	0,6991	0,8154	0,9277	0,6273	0,8141
Extra Tree	1	0,8737	0,8238	0,4675	0,5965	0,7148	0,7213	0,9249	0,5581	0,7870
	2	0,8268	0,7227	0,2150	0,3314	0,4909	0,5972	0,8887	0,3288	0,6589
	3	0,8273	0,7288	0,2150	0,3320	0,4931	0,5975	0,8859	0,3311	0,6589
	4	0,8143	0,5213	0,8550	0,6477	0,5655	0,8296	0,8967	0,5615	0,7639
	5	0,7943	0,4911	0,8275	0,6164	0,5346	0,8068	0,8905	0,5191	0,7439
KNN	1	0,8003	0,0000	0,0000	0,0000	0,0000	0,5000	0,6175	0,0000	0,3725
	2	0,8003	0,0000	0,0000	0,0000	0,0000	0,5000	0,6436	0,0000	0,3812
	3	0,8422	0,7188	0,3450	0,4662	0,5908	0,6557	0,8838	0,4228	0,7101
	4	0,5637	0,2557	0,6200	0,3620	0,2897	0,5848	0,6357	0,1357	0,5034
	5	0,6925	0,3889	0,9450	0,5510	0,4408	0,7872	0,8881	0,4595	0,7054
RNN	1	0,8003	0,0000	0,0000	0,0000	0,0000	0,5000	0,6140	0,0000	0,3713
	2	0,8003	0,0000	0,0000	0,0000	0,0000	0,5000	0,6140	0,0000	0,3713
	3	0,8003	0,0000	0,0000	0,0000	0,0000	0,5000	0,5000	0,0000	0,3333
	4	0,6311	0,2779	0,5300	0,3646	0,3071	0,5931	0,6255	0,1533	0,5086
	5	0,6875	0,3780	0,8750	0,5279	0,4264	0,7578	0,8371	0,4135	0,6738
Adap. Boosting	1	0,8492	0,6571	0,5125	0,5758	0,6220	0,7229	0,8954	0,4914	0,7467
	2	0,8632	0,6944	0,5625	0,6215	0,6633	0,7504	0,9069	0,5436	0,7735
	3	0,8662	0,7037	0,5700	0,6298	0,6722	0,7551	0,9077	0,5538	0,7783
	4	0,8467	0,6183	0,6075	0,6129	0,6161	0,7570	0,8997	0,5174	0,7576
	5	0,8427	0,5914	0,6875	0,6358	0,6084	0,7845	0,9029	0,5387	0,7653
Gradient Boosting	1	0,8632	0,7072	0,5375	0,6108	0,6652	0,7410	0,9157	0,5370	0,7740
	2	0,8702	0,6892	0,6375	0,6623	0,6782	0,7829	0,9015	0,5828	0,7875
	3	0,8632	0,6712	0,6175	0,6432	0,6597	0,7710	0,8998	0,5595	0,7768
	4	0,8502	0,6250	0,6250	0,6250	0,6250	0,7657	0,8867	0,5314	0,7591
	5	0,8662	0,6765	0,6325	0,6537	0,6672	0,7785	0,8815	0,5714	0,7757
SVM - Linear	1	0,4184	0,1576	0,4400	0,2320	0,1808	0,4265	0,5964	-0,1183	0,4012
	2	0,4184	0,1576	0,4400	0,2320	0,1808	0,4265	0,5964	-0,1183	0,4012
	3	0,7369	0,3943	0,5925	0,4735	0,4226	0,6827	0,7406	0,3188	0,6153
	4	0,4119	0,1451	0,3975	0,2126	0,1662	0,4065	0,5554	-0,1502	0,3760
	5	0,4448	0,2170	0,6825	0,3293	0,2513	0,5340	0,4234	0,5627	0,4029
SVM - Polinomial	1	0,8003	0,0000	0,0000	0,0000	0,0000	0,5000	0,6270	0,0000	0,3757
	2	0,7798	0,3787	0,1600	0,2250	0,2974	0,5472	0,6242	0,1359	0,4896
	3	0,8527	0,7869	0,3600	0,4940	0,6360	0,6678	0,8814	0,4657	0,7284
	4	0,7664	0,4124	0,4000	0,4061	0,4098	0,6289	0,6713	0,2607	0,5700
	5	0,8337	0,5573	0,8150	0,6619	0,5949	0,8267	0,9002	0,5745	0,7739

Continuação

Modelo	T	Ac	Pr	Rec	F_1	F_β	B_S	ROC_S	MCC	F_M
SVM - RBF	1	0,8003	0,0000	0,0000	0,0000	0,0000	0,5000	0,7753	0,0000	0,4251
	2	0,8048	0,9091	0,2500	0,4866	0,1126	0,5122	0,7510	0,1319	0,4586
	3	0,8627	0,6761	0,6000	0,6358	0,6593	0,7641	0,9019	0,5530	0,7751
	4	0,7379	0,4198	0,8175	0,5547	0,4650	0,7678	0,8293	0,4392	0,6874
	5	0,8662	0,6793	0,6250	0,6510	0,6677	0,7757	0,9006	0,5692	0,7813
SVM - Sigmoidal	1	0,6810	0,1346	0,1100	0,1210	0,1288	0,4667	0,4100	-0,7198	0,3352
	2	0,6810	0,1346	0,1100	0,1210	0,1288	0,4667	0,5900	-0,7198	0,3952
	3	0,7584	0,3918	0,3800	0,3858	0,3893	0,6164	0,6313	0,2355	0,5457
	4	0,4498	0,1651	0,4325	0,2390	0,1884	0,4433	0,4216	-0,9073	0,3511
	5	0,6345	0,3070	0,6600	0,4190	0,3438	0,6441	0,6987	0,2328	0,5622
RNA - 1 Camada	1	0,8224	0,6029	0,3981	0,4795	0,5467	0,6651	0,8597	0,3896	0,6905
	2	0,8483	0,6098	0,7282	0,6637	0,6303	0,8038	0,8956	0,5705	0,7765
	3	0,8842	0,7419	0,6699	0,7041	0,7263	0,8048	0,9227	0,6336	0,8179
	4	0,8283	0,5630	0,7379	0,6387	0,5910	0,7948	0,8963	0,5371	0,7607
	5	0,8443	0,6214	0,6214	0,6214	0,6214	0,7617	0,9126	0,5234	0,7652
RNA - 2 Camadas	1	0,8144	0,6786	0,1845	0,2901	0,4419	0,5809	0,7694	0,2848	0,5974
	2	0,8124	0,5283	0,8155	0,6412	0,5683	0,8135	0,8795	0,5445	0,7538
	3	0,8643	0,6882	0,6214	0,6531	0,6737	0,7742	0,9268	0,5701	0,7916
	4	0,7884	0,4895	0,6796	0,5691	0,5185	0,7481	0,8539	0,4440	0,7068
	5	0,8643	0,6733	0,6602	0,6667	0,6706	0,7886	0,9284	0,5815	0,7959
RNA - 3 Camadas	1	0,6447	0,3529	0,8738	0,5028	0,4007	0,7296	0,7987	0,3712	0,6430
	2	0,6846	0,3887	0,9320	0,5486	0,4400	0,7763	0,8476	0,4467	0,6880
	3	0,8762	0,7204	0,6505	0,6837	0,7053	0,7926	0,9174	0,6082	0,8051
	4	0,8124	0,5391	0,6019	0,5688	0,5506	0,7344	0,8594	0,4505	0,7148
	5	0,8907	0,7157	0,7087	0,7122	0,7143	0,8179	0,9299	0,6382	0,8207

Fonte: Autor.

Legenda: Ac = Acurácia / Pr = Precisão / Rec = Revocação / F_1 = Medida F para $\beta = 1$ / F_β = Medida F / B_S = Acurácia Balanceada / ROC_S Pontuação ROC / MCC = Coeficiente de Correlação de Matthews / F_M Métrica Final.

Tabela 17 – Resultados obtidos para amostra 4 teste multiclasse

Modelo	T	Ac	Pr	Rec	F_1	F_β	B_S	ROC_S	MCC	F_M
Regressão Logística	1	0,6775	0,5618	0,6775	0,5722	0,5406	0,1623	0,7293	0,1334	0,4774
	2	0,7798	0,7627	0,7798	0,7657	0,7619	0,5032	0,9273	0,5501	0,7308
	3	0,8038	0,7931	0,8038	0,7969	0,7942	0,5817	0,9423	0,6091	0,7727
	4	0,7279	0,7660	0,7279	0,7378	0,7521	0,5236	0,9159	0,4836	0,7305
	5	0,8103	0,8068	0,8103	0,8059	0,8058	0,6451	0,9414	0,6250	0,7974

Continuação

Modelo	T	Ac	Pr	Rec	F_1	F_β	B_S	ROC_S	MCC	F_M
NBG	1	0,5267	0,7456	0,5267	0,5847	0,6658	0,5119	0,8491	0,3518	0,6756
	2	0,5037	0,7550	0,5037	0,5641	0,6585	0,5431	0,8567	0,3454	0,6861
	3	0,4993	0,7468	0,4993	0,5599	0,6522	0,5429	0,8476	0,3385	0,6809
	4	0,5217	0,7540	0,5217	0,5770	0,6654	0,5642	0,8589	0,3596	0,6962
	5	0,5102	0,7457	0,5102	0,5698	0,6576	0,5348	0,8490	0,3437	0,6805
NBB	1	0,6930	0,5212	0,6930	0,5921	0,5467	0,1813	0,7069	0,2325	0,4783
	2	0,5537	0,6923	0,5537	0,6005	0,6500	0,4372	0,8046	0,3224	0,6306
	3	0,5047	0,7377	0,5047	0,5634	0,6503	0,4737	0,8498	0,3326	0,6579
	4	0,5227	0,6942	0,5227	0,5799	0,6409	0,4467	0,7916	0,3028	0,6264
	5	0,5042	0,7397	0,5042	0,5640	0,6519	0,4802	0,8499	0,3315	0,6607
<i>Decision Tree</i>	1	0,7124	0,7209	0,7124	0,7156	0,7185	0,4608	0,7583	0,4549	0,6459
	2	0,7424	0,7177	0,7424	0,7253	0,7189	0,3788	0,8944	0,4694	0,6640
	3	0,7424	0,7177	0,7424	0,7253	0,7189	0,3788	0,8934	0,4694	0,6637
	4	0,7179	0,7033	0,7179	0,7034	0,7009	0,4177	0,8763	0,4319	0,6650
	5	0,7014	0,7023	0,7014	0,6987	0,6998	0,3833	0,8775	0,4241	0,6536
<i>Random Forest</i>	1	0,7983	0,7802	0,7983	0,7762	0,7729	0,5043	0,9382	0,5757	0,7384
	2	0,7968	0,7790	0,7968	0,7754	0,7714	0,4964	0,9379	0,5741	0,7352
	3	0,7903	0,7614	0,7903	0,7681	0,7617	0,4695	0,9361	0,5596	0,7225
	4	0,7968	0,7797	0,7968	0,7781	0,7756	0,5671	0,9370	0,5769	0,7599
	5	0,7973	0,7779	0,7973	0,7772	0,7742	0,5552	0,9381	0,5756	0,7559
<i>Extra Tree</i>	1	0,8038	0,7890	0,8038	0,7833	0,7809	0,5009	0,9431	0,5884	0,7416
	2	0,7214	0,7059	0,7214	0,6485	0,6275	0,2581	0,8986	0,3532	0,5947
	3	0,7204	0,6881	0,7204	0,6444	0,6171	0,2652	0,8964	0,3517	0,5929
	4	0,6790	0,6830	0,6790	0,6313	0,6292	0,3273	0,8883	0,3129	0,6149
	5	0,6790	0,7414	0,6790	0,6334	0,6324	0,3255	0,8888	0,3141	0,6156
KNN	1	0,6650	0,5307	0,6650	0,5561	0,5184	0,1626	0,6312	0,6532	0,4374
	2	0,6735	0,4536	0,6735	0,5421	0,4853	0,1429	0,6509	0,0000	0,4264
	3	0,7449	0,7070	0,7449	0,7023	0,6931	0,3295	0,9040	0,4363	0,6422
	4	0,4513	0,4596	0,4513	0,4533	0,4570	0,1532	0,5939	0,1299	0,4014
	5	0,6291	0,7716	0,6291	0,6296	0,6615	0,3256	0,8755	0,3103	0,6209
RNN	1	0,6735	0,4536	0,6735	0,5421	0,4853	0,1429	0,6107	0,0000	0,4129
	2	0,6735	0,4536	0,6735	0,5421	0,4853	0,1429	0,6107	0,0000	0,4129
	3	0,6735	0,4536	0,6735	0,5421	0,4853	0,1429	0,5000	0,0000	0,3760
	4	0,3485	0,5062	0,3485	0,4067	0,4611	0,1889	0,6212	0,7524	0,4237
	5	0,5212	0,5559	0,5212	0,5325	0,5462	0,2511	0,8299	0,2029	0,5424
Adap. Boosting	1	0,7139	0,6802	0,7139	0,6818	0,6733	0,4288	0,7295	0,3997	0,6105
	2	0,6530	0,6825	0,6530	0,6609	0,6723	0,4440	0,7118	0,3530	0,6094
	3	0,5642	0,7192	0,5642	0,6065	0,6659	0,4764	0,7050	0,3526	0,6158
	4	0,4348	0,6657	0,4348	0,4895	0,5748	0,4588	0,6960	0,2401	0,5765
	5	0,3380	0,6411	0,3380	0,3879	0,4963	0,4214	0,7155	0,1759	0,5444

Continuação

Modelo	T	Ac	Pr	Rec	F_1	F_β	B_S	ROC_S	MCC	F_M
Gradient Boosting	1	0,8048	0,7892	0,8048	0,7945	0,7907	0,5479	0,9434	0,6030	0,7607
	2	0,7379	0,7264	0,7379	0,7302	0,7273	0,4440	0,8429	0,4777	0,6714
	3	0,7239	0,7176	0,7239	0,7177	0,7169	0,4354	0,8375	0,4550	0,6633
	4	0,5627	0,6877	0,5627	0,5989	0,6436	0,3427	0,7058	0,2947	0,5640
	5	0,8088	0,3499	0,8088	0,1078	0,1799	0,7042	0,4428	-0,1110	0,2310
SVM - Linear	1	0,2561	0,4548	0,2561	0,2889	0,3628	0,2257	0,5383	-0,3658	0,3756
	2	0,2561	0,4548	0,2561	0,2889	0,3628	0,2257	0,4884	-0,3658	0,3590
	3	0,6370	0,6561	0,6370	0,6452	0,6513	0,4814	0,8348	0,3189	0,6558
	4	0,1917	0,4315	0,1917	0,2322	0,3136	0,2680	0,4630	0,1782	0,3482
	5	0,6375	0,6573	0,6375	0,6459	0,6523	0,4740	0,8330	0,3222	0,6531
SVM - Polinomial	1	0,6735	0,4536	0,6735	0,5421	0,4853	0,1429	0,6280	0,0000	0,4187
	2	0,6475	0,5015	0,6475	0,5471	0,5048	0,1733	0,5327	0,7354	0,4036
	3	0,7813	0,7581	0,7813	0,7527	0,7496	0,4857	0,9251	0,5316	0,7201
	4	0,6066	0,4682	0,6066	0,5266	0,4899	0,2117	0,6229	0,6607	0,4415
	5	0,7713	0,7613	0,7713	0,7497	0,7508	0,5479	0,9245	0,5191	0,7411
SVM - RBF	1	0,6735	0,4536	0,6735	0,5421	0,4853	0,1429	0,8383	0,0000	0,4888
	2	0,6855	0,5933	0,6855	0,5774	0,5443	0,1677	0,8407	0,1710	0,5176
	3	0,8153	0,8046	0,8153	0,8077	0,8052	0,6103	0,9460	0,6288	0,7871
	4	0,6216	0,6199	0,6216	0,5748	0,5678	0,2560	0,8280	0,1954	0,5506
	5	0,8148	0,8043	0,8148	0,8073	0,8048	0,6086	0,9447	0,6282	0,7861
SVM - Sigmoidal	1	0,6116	0,4496	0,6116	0,5182	0,4747	0,1359	0,4333	-0,3371	0,3480
	2	0,5891	0,4463	0,5891	0,5077	0,4690	0,1312	0,4257	-0,4557	0,3419
	3	0,6460	0,6063	0,6460	0,6189	0,6085	0,2828	0,7682	0,2598	0,5531
	4	0,4019	0,4344	0,4019	0,4155	0,4266	0,1363	0,4492	-0,3614	0,3374
	5	0,5726	0,6344	0,5726	0,5853	0,6056	0,3337	0,7476	0,2288	0,5623
RNA - 1 Camada	1	0,6627	0,6332	0,6627	0,5917	0,5757	0,3933	0,7731	0,2932	0,5807
	2	0,6966	0,7778	0,6966	0,7184	0,7474	0,5571	0,9114	0,4909	0,7387
	3	0,8084	0,8045	0,8084	0,8058	0,8048	0,6498	0,9442	0,6412	0,7996
	4	0,5629	0,7678	0,5629	0,6239	0,6873	0,4215	0,8250	0,3651	0,6446
	5	0,7665	0,7788	0,7665	0,7668	0,7720	0,5179	0,9344	0,5802	0,7414
RNA - 2 Camadas	1	0,6327	0,7050	0,6327	0,6488	0,6760	0,4739	0,8661	0,3961	0,6720
	2	0,6467	0,6550	0,6467	0,6005	0,5973	0,3429	0,7913	0,3082	0,5772
	3	0,8224	0,8158	0,8224	0,8175	0,8160	0,6547	0,9559	0,6624	0,8089
	4	0,6627	0,6925	0,6627	0,6710	0,6813	0,4840	0,8587	0,3995	0,6747
	5	0,8244	0,8229	0,8244	0,8225	0,8224	0,6702	0,9552	0,6723	0,8160
RNA - 3 Camadas	1	0,7206	0,6836	0,7206	0,6934	0,6832	0,4738	0,8800	0,4347	0,6790
	2	0,5090	0,6832	0,5090	0,5375	0,5959	0,3980	0,7575	0,2647	0,5838
	3	0,8224	0,8177	0,8224	0,8188	0,8178	0,6715	0,9570	0,6653	0,8154
	4	0,4291	0,6935	0,4291	0,5022	0,5982	0,4155	0,7910	0,2549	0,6016
	5	0,8204	0,8208	0,8204	0,8196	0,8200	0,6539	0,9553	0,6684	0,8097

Fonte: Autor.

Legenda: Ac = Acurácia / Pr = Precisão / Rec = Revocação / F_1 = Medida F para $\beta = 1$ / F_β = Medida F / B_S = Acurácia Balanceada / ROC_S Pontuação ROC / MCC Coeficiente de Correlação de Matthews / F_M Métrica Final.