



**VANCLEY OLIVEIRA SIMÃO**

**CLASSIFICAÇÃO DE EMOÇÕES EM SINAIS DE EEG  
UTILIZANDO TÉCNICAS DE APRENDIZADO DE MÁQUINA**

**LAVRAS – MG**

**2017**

**VANCLEY OLIVEIRA SIMÃO**

**CLASSIFICAÇÃO DE EMOÇÕES EM SINAIS DE EEG UTILIZANDO TÉCNICAS  
DE APRENDIZADO DE MÁQUINA**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Engenharia de Sistemas e Automação, área de concentração em Engenharia de Sistemas e Automação, para a obtenção do título de Mestre.

Prof. DSc. Wilian Soares Lacerda  
Orientador

**LAVRAS – MG**

**2017**

**Ficha catalográfica elaborada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca  
Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a).**

Simão, Vancley Oliveira

Classificação de emoções em sinais de EEG utilizando técnicas de aprendizado de máquina / Vancley Oliveira Simão.

– Lavras : UFLA, 2017.

90 p. : il.

Dissertação (Mestrado Acadêmico) –Universidade Federal de Lavras, 2017.

Orientador: Prof. DSc. Wilian Soares Lacerda.

Bibliografia.

1. Emoções. 2. EEG. 3. Aprendizado de Máquina. I. Lacerda, Wilian Soares. . II. Título.


VANCLEY OLIVEIRA SIMÃO

**CLASSIFICAÇÃO DE EMOÇÕES EM SINAIS DE EEG UTILIZANDO TÉCNICAS  
DE APRENDIZADO DE MÁQUINA  
EMOTION CLASSIFICATION IN EEG USING MACHINE LEARNING  
TECHNIQUES**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Engenharia de Sistemas e Automação, área de concentração em Engenharia de Sistemas e Automação, para a obtenção do título de Mestre.

APROVADA em 29 de setembro de 2017.

Prof. Dr. Wilian Soares Lacerda UFLA  
Prof. Dr. Demóstenes Zegarra UFLA  
Prof. Dr. Paulo Fernando Seixas UFLA

  
Prof. DSc. Wilian Soares Lacerda  
Orientador

LAVRAS – MG  
2017

*Dedico esta dissertação de mestrado ao mundo.*

## AGRADECIMENTOS

Agradeço, primeiramente, a Deus, por tudo.

Aos meus pais, **Valdenir dos Reis Simão** e **Maria de Lourdes Oliveira Simão**, por me apoiarem nos momentos difíceis e por acreditarem em mim.

A minha irmã, **Viviany Oliveira Simão**, e ao meu primo, **Manoel Alexandre de Oliveira**, pela amizade e companheirismo.

À **Letícia Machado Leão Paes da Rosa**, pelo carinho, amor, paciência e companheirismo.

Ao Prof. Dr. **Wilian Soares Lacerda**, pela excelente orientação e amizade.

Aos meus amigos e a todos que contribuíram de alguma forma para a conclusão do meu mestrado.

*Become who you are. (Friedrich Nietzsche)*

## RESUMO

As emoções desempenham um papel fundamental no cotidiano humano, influenciando as decisões e até mesmo na forma de comunicação. Entender como são caracterizadas as emoções e como identificá-las são de extrema importância para entendermos melhor como os seres humanos se comportam. Vários métodos de classificação de emoções já foram propostos, porém as pesquisas por padrões existentes nas atividades cerebrais que correlacionam com emoções discretas ainda estão sendo investigados. Assim, esta dissertação de mestrado apresenta uma metodologia para classificar emoções discretas em sinais de EEG utilizando algoritmos de aprendizado de máquina, sendo eles: Redes Neurais Artificiais, Florestas Aleatórias e Máquinas de Vetor Suporte.

Vários modelos foram criados utilizando os algoritmos de aprendizado de máquina, onde métricas de acurácia foram coletadas com a finalidade de realizar uma comparação e assim denotar qual foi o melhor modelo de classificação. Para o desenvolvimento dos modelos foi utilizado a base de dados DEAP, a qual apresentava um desbalanceamento nos dados. Deste modo, foram investigadas técnicas de balanceamento de dados, sendo elas os algoritmos SMOTE e ADASYN.

Como contribuições principais desta dissertação de mestrado estão: o desenvolvimento de modelos de classificação de emoções discretas em sinais de EEG; a avaliação dos algoritmos de aprendizado de máquina e dos algoritmos de balanceamento de dados para o desenvolvimento dos modelos de classificação.

Após realizados o desenvolvimento dos modelos, pela análise das métricas coletadas, observou-se uma acurácia média de 89,22% para os modelos utilizando o algoritmo de balanceamento de dados ADASYN e Florestas Aleatórias como técnica de aprendizado de máquina. Foram observados também, uma acurácia média de 87,36% para os modelos utilizando o algoritmo ADASYN e Máquinas de Vetor Suporte. E por fim, uma acurácia média de 68,56% para os modelos utilizando o algoritmo de balanceamento SMOTE e como classificador, Redes Neurais Artificiais.

Desta forma, foi observado que para o problema de classificação de emoções discretas em sinais de EEG, os modelos utilizando o algoritmo ADASYN para a etapa de balanceamento dos dados e o algoritmo de Florestas Aleatórias como técnica de aprendizado de máquina se mostraram superiores quando comparados aos demais modelos testados.

**Palavras-chave:** Emoções. EEG. Aprendizado de Máquina. Redes Neurais Artificiais. SVM. Florestas Aleatórias



## ABSTRACT

Emotions play a fundamental role in human daily lives, having influences over decisions and even communications. Understanding how emotions are characterized and how they can be identified are of utmost importance to understand how humans behave. Various methods have been proposed for emotion classification, however researches that correlates discrete emotions and brain activity patterns are still being investigated. Thus this master's dissertation presents a methodology for discrete emotion classification in EEG signals using machine learning algorithms.

During the development of this work various models have been created using machine learning algorithms where accuracy metrics have been noted to be compared and thus select the best classification model. For the development of the models, the DEAP dataset was used. Once the dataset was unbalanced, the balancing algorithms SMOTE and ADASYN were evaluated in their capacity to balance data and contribute to the improvement of the classification models.

The main contributions of this dissertation are: the development of the discrete emotion classification models in EEG signals and the evaluation of the machine learning and the data balancing algorithms.

After developed the models, the results showed that the Random Forest model with the ADASYN algorithm for data balancing had the best results with an average accuracy of 89.22%. The models built with the SVM algorithm and the ADASYN technique for balancing the data also showed good results with an average accuracy of 87.36%. Finally, the models with Neural Networks and SMOTE algorithms showed the worst results with an average accuracy of 68.56%.

Therefore, the results showed that for the discrete emotion classification in EEG signals, the models using the ADASYN algorithm for data balancing with the Random Forest algorithm for classification were superior when compared to the other tested models.

**Keywords:** Emotions. EEG. Machine Learning. Artificial Neural Networks. SVM. Random Forest

## LISTA DE FIGURAS

Figura 1.1 – Artigos publicados relacionados a classificação de emoções nos últimos 70 anos. . . . .	15
Figura 2.1 – Diagrama de uma sistema de gravação de EEG. . . . .	18
Figura 2.2 – Eletrodos em um EEG. . . . .	18
Figura 2.3 – Sistema Internacional 10-20. . . . .	20
Figura 2.4 – Modelo emocional proposto por Plutchik. . . . .	23
Figura 2.5 – Modelo emocional circumplexo proposto por Russell. . . . .	24
Figura 2.6 – Neurônio do tipo Perceptron. . . . .	26
Figura 2.7 – Arquitetura de uma Rede Neural <i>Feedforward</i> de camada única. . . . .	28
Figura 2.8 – Arquitetura de uma Rede Neural <i>Feedforward</i> de múltiplas camadas. . . . .	29
Figura 2.9 – Arquitetura de uma Rede Neural <i>Feedforward</i> Recorrente. . . . .	30
Figura 2.10 – Exemplo de SVM com diferentes kernels. . . . .	36
Figura 2.11 – Exemplo da técnica Oversampling. . . . .	37
Figura 2.12 – Exemplo da técnica Undersampling. . . . .	38
Figura 2.13 – Exemplo da técnica SMOTE. . . . .	39
Figura 2.14 – Exemplo da técnica ADASYN. . . . .	41
Figura 3.1 – Esquemático da metodologia proposta. . . . .	45
Figura 3.2 – Árvore de decisão para a classificação das amostras em emoções discretas. . . . .	47
Figura 3.3 – Distribuição dos dados em função das classes. . . . .	48
Figura 4.1 – Análise de variância explicada. . . . .	52
Figura 4.2 – MLP - Curva ROC para a Classe 1: Protegido . . . . .	53
Figura 4.3 – MLP - Curva ROC para a Classe 2: Satisfeito . . . . .	54
Figura 4.4 – MLP - Curva ROC para a Classe 3: Surpreso . . . . .	54
Figura 4.5 – MLP - Curva ROC para a Classe 4: Alegre . . . . .	55
Figura 4.6 – MLP - Curva ROC para a Classe 5: Triste . . . . .	55
Figura 4.7 – MLP - Curva ROC para a Classe 6: Despreocupado . . . . .	56
Figura 4.8 – MLP - Curva ROC para a Classe 7: Medo . . . . .	56
Figura 4.9 – MLP - Curva ROC para a Classe 8: Bravo . . . . .	57
Figura 4.10 – SVM - Curva ROC para a Classe 1: Protegido . . . . .	62
Figura 4.11 – SVM - Curva ROC para a Classe 2: Satisfeito . . . . .	62
Figura 4.12 – SVM - Curva ROC para a Classe 3: Surpreso . . . . .	63

Figura 4.13 – SVM - Curva ROC para a Classe 4: Alegre . . . . .	63
Figura 4.14 – SVM - Curva ROC para a Classe 5: Triste . . . . .	64
Figura 4.15 – SVM - Curva ROC para a Classe 6: Despreocupado . . . . .	64
Figura 4.16 – SVM - Curva ROC para a Classe 7: Medo . . . . .	65
Figura 4.17 – SVM - Curva ROC para a Classe 8: Bravo . . . . .	65
Figura 4.18 – Florestas Aleatórias - Curva ROC para a Classe 1: Protegido . . . . .	69
Figura 4.19 – Florestas Aleatórias - Curva ROC para a Classe 2: Satisfeito . . . . .	70
Figura 4.20 – Florestas Aleatórias - Curva ROC para a Classe 3: Surpreso . . . . .	70
Figura 4.21 – Florestas Aleatórias - Curva ROC para a Classe 4: Alegre . . . . .	71
Figura 4.22 – Florestas Aleatórias - Curva ROC para a Classe 5: Triste . . . . .	71
Figura 4.23 – Florestas Aleatórias - Curva ROC para a Classe 6: Despreocupado . . . . .	72
Figura 4.24 – Florestas Aleatórias - Curva ROC para a Classe 7: Medo . . . . .	72
Figura 4.25 – Florestas Aleatórias - Curva ROC para a Classe 8: Bravo . . . . .	73
Figura 4.26 – Acurácia média para o modelo utilizando RNA. . . . .	76
Figura 4.27 – Acurácia média para o modelo utilizando Florestas Aleatórias. . . . .	77
Figura 4.28 – Acurácia média para o modelo utilizando SVM. . . . .	77

## LISTA DE TABELAS

Tabela 2.1 – Classificação de sinais pela frequência. . . . .	19
Tabela 2.2 – Emoções caracterizadas no modelo emocional PAD. . . . .	25
Tabela 2.3 – Matriz de confusão . . . . .	43
Tabela 3.1 – Descrição do experimento psicológico realizado em (KOELSTRA et al., 2012). . . . .	46
Tabela 4.1 – Resultados para o modelo utilizando RNA + SMOTE. . . . .	52
Tabela 4.2 – Resultados para o modelo utilizando RNA + ADASYN. . . . .	53
Tabela 4.3 – MLP + ADASYN - Matriz de confusão para a Classe 1: Protegido . . . . .	57
Tabela 4.4 – MLP + SMOTE - Matriz de confusão para a Classe 1: Protegido . . . . .	57
Tabela 4.5 – MLP + ADASYN - Matriz de confusão para a Classe 2: Satisfeito . . . . .	58
Tabela 4.6 – MLP + SMOTE - Matriz de confusão para a Classe 2: Satisfeito . . . . .	58
Tabela 4.7 – MLP + ADASYN - Matriz de confusão para a Classe 3: Surpreso . . . . .	58
Tabela 4.8 – MLP + SMOTE - Matriz de confusão para a Classe 3: Surpreso . . . . .	58
Tabela 4.9 – MLP + ADASYN - Matriz de confusão para a Classe 4: Alegre . . . . .	58
Tabela 4.10 – MLP + SMOTE - Matriz de confusão para a Classe 4: Alegre . . . . .	58
Tabela 4.11 – MLP + ADASYN - Matriz de confusão para a Classe 5: Triste . . . . .	59
Tabela 4.12 – MLP + SMOTE - Matriz de confusão para a Classe 5: Triste . . . . .	59
Tabela 4.13 – MLP + ADASYN - Matriz de confusão para a Classe 6: Despreocupado . . . . .	59
Tabela 4.14 – MLP + SMOTE - Matriz de confusão para a Classe 6: Despreocupado . . . . .	59
Tabela 4.15 – MLP + ADASYN - Matriz de confusão para a Classe 7: Medo . . . . .	59
Tabela 4.16 – MLP + SMOTE - Matriz de confusão para a Classe 7: Medo . . . . .	59
Tabela 4.17 – MLP + ADASYN - Matriz de confusão para a Classe 8: Bravo . . . . .	60
Tabela 4.18 – MLP + SMOTE - Matriz de confusão para a Classe 8: Bravo . . . . .	60
Tabela 4.19 – Resultados para o modelo utilizando SVM + SMOTE. . . . .	61
Tabela 4.20 – Resultados para o modelo utilizando SVM + ADASYN. . . . .	61
Tabela 4.21 – SVM + ADASYN - Matriz de confusão para a Classe 1: Protegido . . . . .	66
Tabela 4.22 – SVM + SMOTE - Matriz de confusão para a Classe 1: Protegido . . . . .	66
Tabela 4.23 – SVM + ADASYN - Matriz de confusão para a Classe 2: Satisfeito . . . . .	66
Tabela 4.24 – SVM + SMOTE - Matriz de confusão para a Classe 2: Satisfeito . . . . .	66
Tabela 4.25 – SVM + ADASYN - Matriz de confusão para a Classe 3: Surpreso . . . . .	66
Tabela 4.26 – SVM + SMOTE - Matriz de confusão para a Classe 3: Surpreso . . . . .	67

Tabela 4.27 – SVM + ADASYN - Matriz de confusão para a Classe 4: Alegre . . . . .	67
Tabela 4.28 – SVM + SMOTE - Matriz de confusão para a Classe 4: Alegre . . . . .	67
Tabela 4.29 – SVM + ADASYN - Matriz de confusão para a Classe 5: Triste . . . . .	67
Tabela 4.30 – SVM + SMOTE - Matriz de confusão para a Classe 5: Triste . . . . .	67
Tabela 4.31 – SVM + ADASYN - Matriz de confusão para a Classe 6: Despreocupado . .	67
Tabela 4.32 – SVM + SMOTE - Matriz de confusão para a Classe 6: Despreocupado . . .	68
Tabela 4.33 – SVM + ADASYN - Matriz de confusão para a Classe 7: Medo . . . . .	68
Tabela 4.34 – SVM + SMOTE - Matriz de confusão para a Classe 7: Medo . . . . .	68
Tabela 4.35 – SVM + ADASYN - Matriz de confusão para a Classe 8: Bravo . . . . .	68
Tabela 4.36 – SVM + SMOTE - Matriz de confusão para a Classe 8: Bravo . . . . .	68
Tabela 4.37 – Resultados para o modelo utilizando Florestas Aleatórias + SMOTE. . . . .	69
Tabela 4.38 – Resultados para o modelo utilizando Florestas Aleatórias + ADASYN. . .	69
Tabela 4.39 – Florestas Aleatórias + ADASYN - Matriz de confusão para a Classe 1:	
Protegido . . . . .	71
Tabela 4.40 – Florestas Aleatórias + SMOTE - Matriz de confusão para a Classe 1: Pro-	
tegido . . . . .	72
Tabela 4.41 – Florestas Aleatórias + ADASYN - Matriz de confusão para a Classe 2:	
Satisfeito . . . . .	73
Tabela 4.42 – Florestas Aleatórias + SMOTE - Matriz de confusão para a Classe 2: Satisfeito	73
Tabela 4.43 – Florestas Aleatórias + ADASYN - Matriz de confusão para a Classe 3:	
Surpreso . . . . .	73
Tabela 4.44 – Florestas Aleatórias + SMOTE - Matriz de confusão para a Classe 3: Surpreso	74
Tabela 4.45 – Florestas Aleatórias + ADASYN - Matriz de confusão para a Classe 4: Alegre	74
Tabela 4.46 – Florestas Aleatórias + SMOTE - Matriz de confusão para a Classe 4: Alegre	74
Tabela 4.47 – Florestas Aleatórias + ADASYN - Matriz de confusão para a Classe 5: Triste	74
Tabela 4.48 – Florestas Aleatórias + SMOTE - Matriz de confusão para a Classe 5: Triste	74
Tabela 4.49 – Florestas Aleatórias + ADASYN - Matriz de confusão para a Classe 6:	
Despreocupado . . . . .	74
Tabela 4.50 – Florestas Aleatórias + SMOTE - Matriz de confusão para a Classe 6: Des-	
preocupado . . . . .	75
Tabela 4.51 – Florestas Aleatórias + ADASYN - Matriz de confusão para a Classe 7: Medo	75
Tabela 4.52 – Florestas Aleatórias + SMOTE - Matriz de confusão para a Classe 7: Medo	75

Tabela 4.53 – Florestas Aleatórias + ADASYN - Matriz de confusão para a Classe 8: Bravo	75
Tabela 4.54 – Florestas Aleatórias + SMOTE - Matriz de confusão para a Classe 1: Bravo	75
Tabela 4.55 – Tabela comparativa com a acurácia média dos modelos. . . . .	76

## SUMÁRIO

<b>1</b>	<b>Introdução</b>	14
<b>1.1</b>	<b>Objetivos</b>	15
<b>1.1.1</b>	<b>Objetivos Gerais</b>	15
<b>1.1.2</b>	<b>Objetivos específicos</b>	15
<b>1.2</b>	<b>Justificativa</b>	16
<b>1.3</b>	<b>Organização do Texto</b>	16
<b>2</b>	<b>Referencial Teórico</b>	17
<b>2.1</b>	<b>Eletroencefalograma</b>	17
<b>2.2</b>	<b>Parâmetros de Hjorth</b>	20
<b>2.3</b>	<b>Psicologia das Emoções</b>	21
<b>2.4</b>	<b>Redes Neurais Artificiais</b>	25
<b>2.4.1</b>	<b>Neurônio Artificial</b>	25
<b>2.4.2</b>	<b>Arquitetura de uma Rede Neural Artificial</b>	27
<b>2.4.3</b>	<b>Formas de aprendizado</b>	30
<b>2.4.4</b>	<b><i>Back-Propagation</i> - Aprendizado por retropropagação</b>	31
<b>2.5</b>	<b>Florestas Aleatórias</b>	32
<b>2.6</b>	<b>Máquinas de Vetor Suporte</b>	33
<b>2.7</b>	<b>Aprendizado de Máquina em Dados Não-balanceados</b>	35
<b>2.7.1</b>	<b><i>Random Oversampling</i> e <i>Undersampling</i></b>	37
<b>2.7.2</b>	<b>SMOTE</b>	39
<b>2.7.3</b>	<b>ADASYN</b>	40
<b>2.8</b>	<b>Análise de Componentes Principais</b>	41
<b>2.9</b>	<b>Matriz de Confusão</b>	42
<b>2.9.1</b>	<b><i>Receiver Operating Characteristic</i></b>	43
<b>3</b>	<b>Materiais e Métodos</b>	44
<b>3.1</b>	<b>Base de Dados DEAP</b>	44
<b>3.1.1</b>	<b>Modelagem das classes conforme o modelo emocional PAD</b>	44
<b>3.2</b>	<b>Etapa de Balanceamento dos Dados</b>	48
<b>3.3</b>	<b>Etapa de Extração e Seleção de Características</b>	49
<b>3.4</b>	<b>Implementação dos Classificadores</b>	49
<b>3.5</b>	<b>Métricas de comparação</b>	50

<b>4</b>	<b>Resultados e Discussões . . . . .</b>	<b>51</b>
<b>4.1</b>	<b>Análise de Componentes Principais . . . . .</b>	<b>51</b>
<b>4.2</b>	<b>Implementação do modelo utilizando Rede Neural Artificiais . . . . .</b>	<b>51</b>
<b>4.3</b>	<b>Implementação do modelo utilizando Máquinas de Vetor Suporte . . . . .</b>	<b>60</b>
<b>4.4</b>	<b>Implementação do modelo utilizando Florestas Aleatórias . . . . .</b>	<b>66</b>
<b>4.5</b>	<b>Comparando os resultados . . . . .</b>	<b>75</b>
<b>5</b>	<b>Conclusão . . . . .</b>	<b>78</b>
<b>5.1</b>	<b>Propostas de Continuidade . . . . .</b>	<b>78</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>79</b>
	<b>APENDICE A – Código Fonte . . . . .</b>	<b>84</b>



## 1 INTRODUÇÃO

Entender como as respostas emocionais são geradas no cérebro e como os estados emocionais se associam com padrões específicos com as respostas fisiológicas sempre despertaram interesse na comunidade acadêmica (LEVENSON et al., 1992). Estudos indicam que as respostas fisiológicas das emoções são mais propícias a serem encontradas no sistema nervoso central (BRAZIER, 2003).

Em geral, a metodologia para o desenvolvimento de um sistema que utilizam uma Interface Cérebro-Computador (ICC), que são conjuntos de hardware e software que fornecem um meio de comunicação entre o cérebro e um computador, tem por fases:

- a coleta dos sinais do cérebro;
- a extração e seleção de características provenientes desses sinais;
- e por fim, a implementação de classificadores para a identificação de padrões.

Dentre as principais formas de se obter os sinais elétricos produzidos pela atividade cerebral, tem-se a utilização de um eletroencefalograma (EEG). Estudos que se baseiam em EEGs demonstram uma atividade assimétrica na parte frontal do cérebro ligada as respostas emocionais (COAN; ALLEN; HARMON-JONES, 2001).

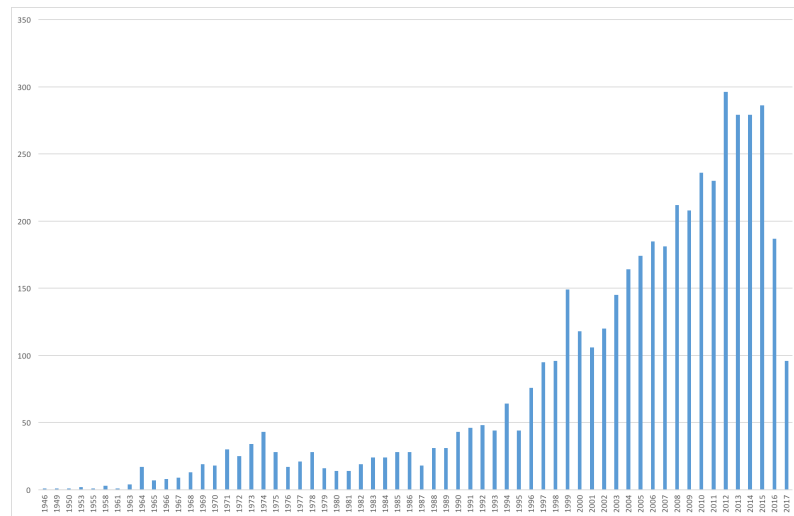
Um dos principais problemas encontrados em classificar emoções através dos sinais obtidos por um EEG está em achar as principais características que melhor representem esse sinal, assim como achar as especificidades de cada emoção. Portanto, experimentos pelos quais indivíduos são submetidos a estímulos audiovisuais vem sendo desenvolvidos, como em (LEE; HSIEH, 2014) e (LI et al., 2014).

Classificar uma emoção é um processo complexo. A utilização de técnicas de aprendizado de máquina, como Redes Neurais Artificiais e Florestas Aleatórias, dentre outras, podem vir a apresentar bons resultados devido a sua alta capacidade de generalização em reconhecimento de padrões.

A forma com que os seres humanos respondem a determinados estímulos pode dizer muito sobre sua personalidade, desse modo verificar se através da análise da resposta fisiológica pode-se acelerar uma eventual classificação de transtornos de personalidade.

Nos últimos anos houve um grande número de pesquisas e publicações no tema de classificação de emoções, conforme pode ser visualizado na Figura 1.1, onde é apresentado o número de artigos relacionados a classificação de emoções nos últimos anos.

Figura 1.1 – Artigos publicados relacionados a classificação de emoções nos últimos 70 anos.



Fonte: PubMed, acessado no dia 03 de setembro de 2017.

Assim, o estudo das melhores abordagens para a classificação das emoções, além de um estudo das melhores características a serem utilizadas nestes modelos, se faz relevante para a comunidade acadêmica.

## 1.1 Objetivos

Esta seção apresenta os objetivos gerais e específicos do projeto.

### 1.1.1 Objetivos Gerais

Essa dissertação de mestrado tem por objetivo geral conseguir identificar padrões nas respostas fisiológicas do cérebro por meio da análise de padrões existentes nos sinais de EEG e fazer uma correlação entre esses e as emoções discretas mapeadas pelo modelo PAD (*Pleasure, Arousal e Dominance*).

### 1.1.2 Objetivos específicos

Como objetivos específicos, encontram-se:

- classificar emoções através dos sinais de EEG provenientes da base de dados DEAP;
- identificar os melhores algoritmos de balanceamento de dados para os dados de EEG;
- propor um modelo geral para classificação de emoções através da análise de sensores específicos em um EEG.

## 1.2 Justificativa

Emoções permitem uma análise compreensiva dos sentimentos, pensamentos e comportamentos de uma pessoa (BAGHDADI; ARIBI; ALIM, 2017), influenciando diretamente a rotina de uma pessoa. Na área médica por exemplo, se for possível determinar em tempo real as alterações do estado emocional de um paciente, intervenções no tratamento do paciente poderão ser realizadas (XIN et al., 2016).

Nos últimos anos o problema de classificação de emoções ganhou destaque na academia, sendo abordado de diversas formas. Modelos foram propostos para classificar emoções através de diversas fontes de dados, como em (ZHANG; MENG; LI, 2016) onde são classificadas emoções em músicas. Entretanto, avanços na área da neurociência mostraram a existência de correlações entre as emoções e o sistema nervoso central (IZARD, 2007). O que viabilizou buscar padrões nos sinais de EEG que correlacionavam com emoções.

Não existem ainda um consenso por parte dos pesquisadores a respeito de quais características são mais apropriadas para classificar emoções em sinais de EEG (JENKE; PEER; BUSS, 2014) e nem da melhor metodologia de classificação. Técnicas de aprendizado de máquina vem sendo empregadas para proporcionar uma maior precisão na classificação dos dados, como em (CHUNG; YOON, 2012) onde é utilizado um classificador Bayesiano. Em (KOTTI; STYLIANOU, 2017) é realizada uma comparação de algoritmos de agrupamentos comparados, no caso o KNN, a técnicas consolidadas como SVM e Florestas Aleatórias. Desse modo, ainda são necessárias pesquisas que visem definir a melhor metodologia e algoritmos para caracterização e identificação desses padrões e suas correlações com as emoções.

## 1.3 Organização do Texto

Essa dissertação de mestrado está organizada da seguinte forma: no Capítulo 2 será apresentada uma revisão da literatura sobre Emoções, dos algoritmos de balanceamento de dados SMOTE e ADASYN, do algoritmo de seleção de atributos PCA, e dos algoritmos de aprendizado de máquina Redes Neurais Artificiais, Máquinas Vetor Suporte e Florestas Aleatórias. No Capítulo 3 serão apresentados os métodos utilizados para o desenvolvimento dessa dissertação, bem como os materiais utilizados. Por fim, nos Capítulos 4 e 5 serão apresentados os resultados e conclusões obtidas.

## 2 REFERENCIAL TEÓRICO

Neste capítulo serão apresentados os conceitos, em forma de referencial teórico. A priori foi feita uma revisão sistemática sobre a psicologia das emoções, onde são apresentadas as principais teorias sobre a origem e caracterização das emoções. Em seguida, são revisados os conceitos dos algoritmos de aprendizado de máquina utilizados nesta dissertação de mestrado: Redes Neurais Artificiais, Florestas Aleatórias e Máquinas de Vetor Suporte. Também são fornecidas as formulações dos algoritmos de balanceamento de dados SMOTE e ADASYN além da técnica de Análise de Componentes Principais (PCA), que é utilizada para seleção de características e visualização de base de dados de alta dimensão.

### 2.1 Eletroencefalograma

Um eletroencefalograma (EEG) é o processo no qual a atividade cerebral pode ser monitorada através do sensoriamento do cérebro (NICOLAS-ALONSO; GOMEZ-GIL, 2012).

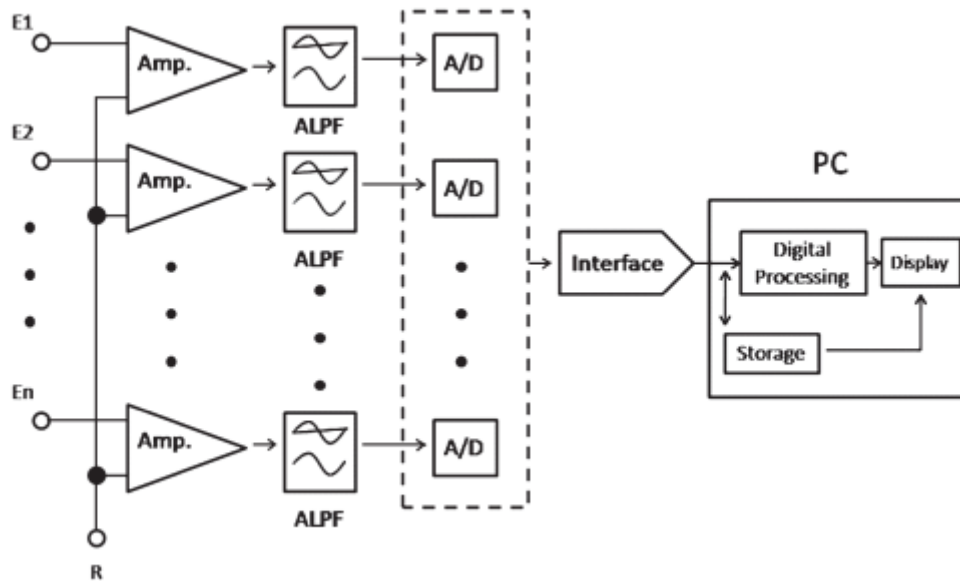
Um sinal de EEG é composto pela soma dos pequenos impulsos elétricos emitidos pelas centenas de milhões de neurônios presentes no cérebro humano (ZHANG; GAO; CHEN, 2011). Os sinais de EEG podem ser facilmente medidos de maneira não invasiva através de eletrodos colocados no escalpo de uma pessoa, o que justifica a sua popularidade. No entanto, os sinais de EEG são altamente complexos e extremamente sensíveis a ruídos. Essa sensibilidade dos sinais influencia diretamente na qualidade do sinal monitorado, além de tornar indispensável o processamento desses sinais (KRISNANDHIKA et al., 2017).

Aplicações que utilizam sinais de EEG são abundantes, abrangendo áreas da Psiquiatria, Psicologia, Pedagogia, entre outras. A análise de um sinal de EEG pode ajudar a fornecer informações importantes para a identificação de doenças, por exemplo.

Em geral um sistema de gravação de EEG é formado por eletrodos, amplificadores, conversores analógico-digital e um aparelho de gravação. Os sinais são obtidos pelos eletrodos espalhados pela superfície do crânio, e em seguida são processados por amplificadores aumentando a amplitude para que em seguida, sejam convertidos para um sinal digital pelo conversor analógico-digital. Por fim, são gravados pelo dispositivo de gravação.

A Figura 2.1 denota um diagrama geral com os elementos que compõem um sistema de gravação de EEG.

Figura 2.1 – Diagrama de uma sistema de gravação de EEG.



Fonte: (VELARDE-REYES; MÁRQUEZ-BOCALANDRO; MARTÍNEZ-MONTES, 2012).

O sinal de um EEG é obtido pela diferença potencial em função do tempo entre um eletrodo ativo para um eletrodo de referência. Há ainda um terceiro eletrodo chamado de eletrodo terra, que é responsável por medir a diferença de voltagem entre os eletrodos ativos e seus eletrodos de referência. Dessa forma, para um EEG é necessário que pelo menos 3 eletrodos sejam utilizados: um para medição, um para referência e um para aterramento (USAKLI, 2010). Existem ainda EEGs de configurações multicanais, que podem ter até 256 eletrodos ativos. A Figura 2.2 mostra uma grande quantidade de eletrodos espalhados pelo escalpo de uma pessoa.

Figura 2.2 – Eletrodos em um EEG.



Fonte: <<https://goo.gl/bXcWds>>, acessado em 11/09/2017.

Os eletrodos, geralmente, são construídos por um sal Cloreto de Prata (AgCl) (TALLGREN et al., 2005). De acordo com (ZHANG; CHASE, 2016) a impedância do contato entre o escalpo e o eletrodo deve ser entre 1 k $\Omega$  e 10 k $\Omega$ , e a interface de contato entre a pele e o eletrodo deve ser resistiva e capacitiva para se comportar como um filtro passa-baixo. Geralmente é utilizado um gel para criar um caminho condutivo entre a pele e o eletrodo reduzindo a impedância desse contato. Existem ainda eletrodos "secos", que não utilizam gel e geralmente são feitos de titânio ou aço inoxidável.

Os sinais de um EEG podem ser classificados de acordo com a sua frequência. A Tabela 2.1 apresenta os tipos de sinais, bem como o intervalo de frequência que caracterizam cada tipo.

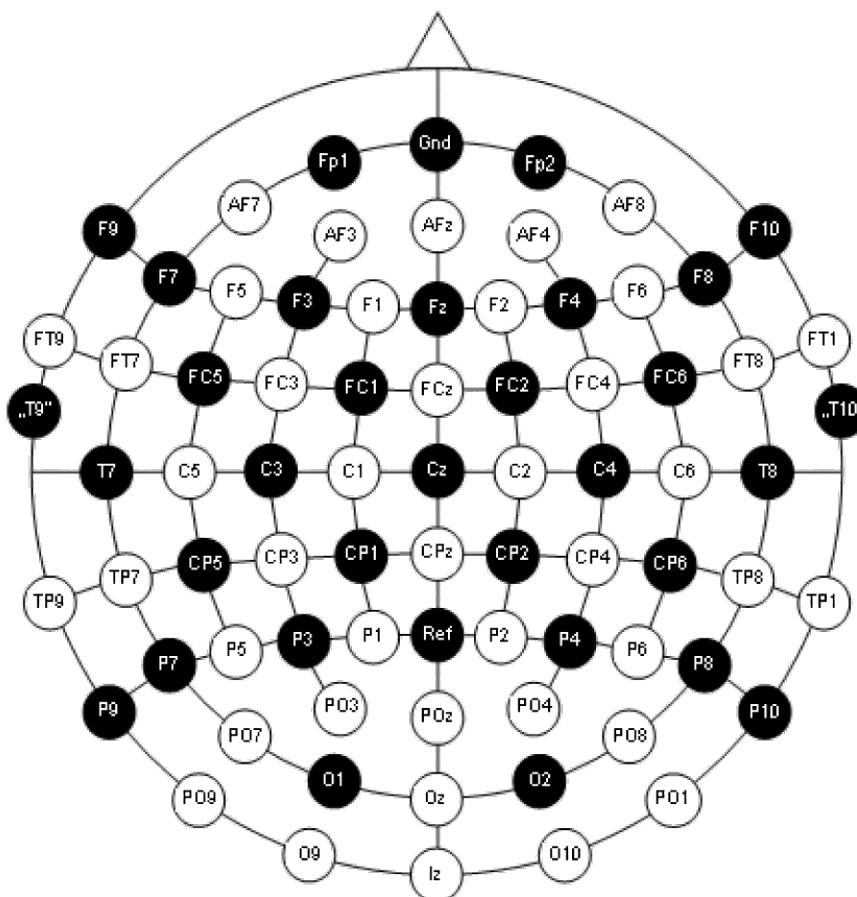
Tabela 2.1 – Classificação de sinais pela frequência.

<b>Tipo de Sinal</b>	<b>Frequência</b>
Delta	1 Hz - 4 Hz
Theta	4 Hz - 8Hz
Alpha	8 Hz - 12 Hz
Beta	12 Hz - 30 Hz
Gamma	30 Hz - 100 Hz

A fim de se padronizar o posicionamento dos sensores no crânio humano durante um EEG, foi criado o Sistema Internacional 10-20. O posicionamento de cada sensor foi definido pela relação entre a área do córtex cerebral e a localização dos sensores.

A Figura 2.3 apresenta a localização de cada sensor no Sistema Internacional 10-20.

Figura 2.3 – Sistema Internacional 10-20.



Fonte: <<http://www.fieldtriptoolbox.org/template/layout>>, acessado em 11/09/2017.

## 2.2 Parâmetros de Hjorth

Bo Hjorth em (HJORTH, 1970), em função da necessidade de métodos quantitativos que descrevessem os sinais de EEG, desenvolveu um sistema de parâmetros para caracterização de padrões encontrados em sinais de EEG tanto pela análise do domínio da frequência quanto no domínio do tempo.

Os parâmetros definidos por ele foram: Atividade, Mobilidade e Complexidade. O parâmetro Atividade é calculado pela variância do sinal em função do tempo, e pode indicar a superfície do espectro de potência no domínio da frequência. O parâmetro Mobilidade é calculado como a raiz quadrada da variância da primeira derivada do sinal dividida pela variância do sinal. Já o parâmetro Complexidade indica o quão similar o sinal é de um sinal senoidal.

Abaixo são apresentadas as definições matemáticas para o cálculo de cada um dos parâmetros de Hjorth, onde  $\xi$  representa um sinal e  $\mu$  representa a média do sinal.

$$\text{Atividade} : A_{\xi} = \frac{\sum_{t=1}^T (\xi(t) - \mu)^2}{T} \quad (2.1)$$

$$\text{Mobilidade} : M_{\xi} = \sqrt{\frac{\text{var}(\xi'(t))}{\text{var}(\xi(t))}} \quad (2.2)$$

$$\text{Complexidade} : C_{\xi} = \frac{M(\xi'(t))}{M(\xi(t))} \quad (2.3)$$

### 2.3 Psicologia das Emoções

O que é uma emoção? Essa é uma das perguntas mais antigas feitas pelo homem, e remete a pensadores antigos como Platão. O estudo das emoções existe em várias áreas do conhecimento, entretanto, defini-la não é uma tarefa fácil, o que faz com que pesquisadores não cheguem em uma definição única e absoluta (DREVER, 1932). O que se tem certeza é da importância do estudo e entendimento das emoções, e do seu papel fundamental em modelos científicos da mente e do comportamento humano (LERNER; KELTNER, 2000).

Muitos cientistas afirmam que emoções básicas, como alegria, tristeza, raiva e medo, tem uma grande influência no dia-a-dia das pessoas. Ações, tomada de decisões e até mesmo o pensamento são fortemente influenciados pelo estado emocional de uma pessoa. Emoções são vistas como causas ou efeitos de outros processos psicológicos e cognitivos do ser humano, como a percepção do que se encontra ao seu redor, atenção e memória (BARRETT, 2006).

Vários modelos foram teorizados para caracterizar cada emoção, além de tentar entender a correlação entre as respostas emocionais a determinados estímulos, em conjunto com o estudo das respostas fisiológicas do cérebro.

Cientistas influenciados por Darwin e pelos recentes estudos na área da neurociência, acreditam que as emoções mais básicas são fenômenos neurofisiológicos moldados pela seleção natural ao longo do tempo. Assim, o conjunto de padrões de respostas aos estímulos e cenários aos quais os indivíduos foram submetidos, evoluiu juntamente com os humanos (IZARD, 1992).

Uma emoção básica pode ser considerada como um conjunto de componentes neurais, comportamentais e emocionais gerados de forma natural, automática, e de forma não consciente como resposta a um estímulo (IZARD, 2007).

Para Ekman (SANDER, 2008), algumas características são importantes na identificação de emoções, como parâmetros relacionados ao interior de um indivíduo (ex. planos e me-



mórias), histórico de situações passadas e a percepção do que se pode ocorrer em um futuro próximo. As emoções básicas identificadas e propostas por Ekman foram: nojo, raiva, medo, tristeza, alegria, surpresa e desprezo.

Para Levenson em (LEVENSON, 2011) as emoções básicas devem apresentar alguns elementos, onde devem ser:

- claras e distintas, para que possam serem diferenciadas umas das outras;
- *hard-wiredness*, ou seja, estarem embutidas no sistema nervoso de forma primitiva;
- funcionais, de modo que consigam descrever o que um indivíduo sente durante determinadas situações críticas para sua sobrevivência.

Levenson define como sendo emoções básicas: alegria, raiva, nojo, medo, surpresa e tristeza.

Emoções podem ser expressadas através de diversos canais sensoriais diferentes, como a fala e microexpressões faciais (TAKAGI et al., 2015).

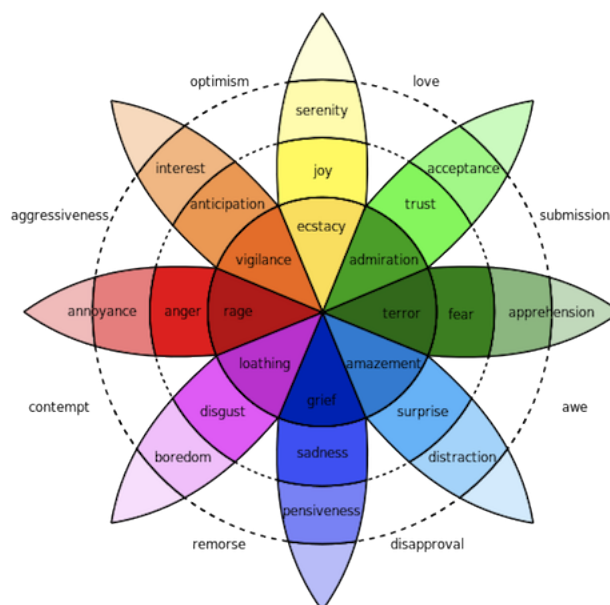
Robert Plutchik (PLUTCHIK, 2001), considera que a percepção de cada emoção é pessoal, e na maior parte das vezes é experimentada ao mesmo tempo com outras emoções, o que torna a experiência confusa. Para Plutchik, algumas emoções são frutos de outras emoções de forma que as emoções que não são primárias podem ser formadas através de misturas de emoções básicas. Na Figura 2.4 é possível ver o modelo de Plutchik para as emoções primárias bem como as suas derivações e combinações em outras emoções.

Outros modelos para caracterizar emoções discretas foram surgindo com o passar do tempo. Entre esses modelos, pode-se destacar os modelos dimensionais. Entre esses estão o modelo emocional circumplexo proposto por Russell em (RUSSELL, 1980) e o modelo PAD proposto por Russel e Mehrabian em (MEHRABIAN, 1995).

No modelo emocional circumplexo de Russel o espaço é composto por duas dimensões, onde todos os estados emocionais podem ser descritos analisando somente dois sistemas neurofisiológicos fundamentais: um relacionado a *valence*, e outro relacionado a *arousal* (POSNER; RUSSELL; PETERSON, 2005). Logo, cada emoção pode ser representada por uma combinação linear dessas duas dimensões.

A Figura 2.5 apresenta o modelo emocional circumplexo de Russell, onde pode ser visualizada a distribuição das emoções mapeadas no espaço emocional.

Figura 2.4 – Modelo emocional proposto por Plutchik.



Fonte: (BANN, 2012).

Entretanto, evidências sugeriram que somente duas dimensões não seriam suficiente para descrever todas as emoções. Desse modo, Mehrabian e Russell propuseram o modelo emocional PAD (*Pleasure, Arousal, Dominance*) em (MEHRABIAN, 1995) e (MEHRABIAN, 1996).

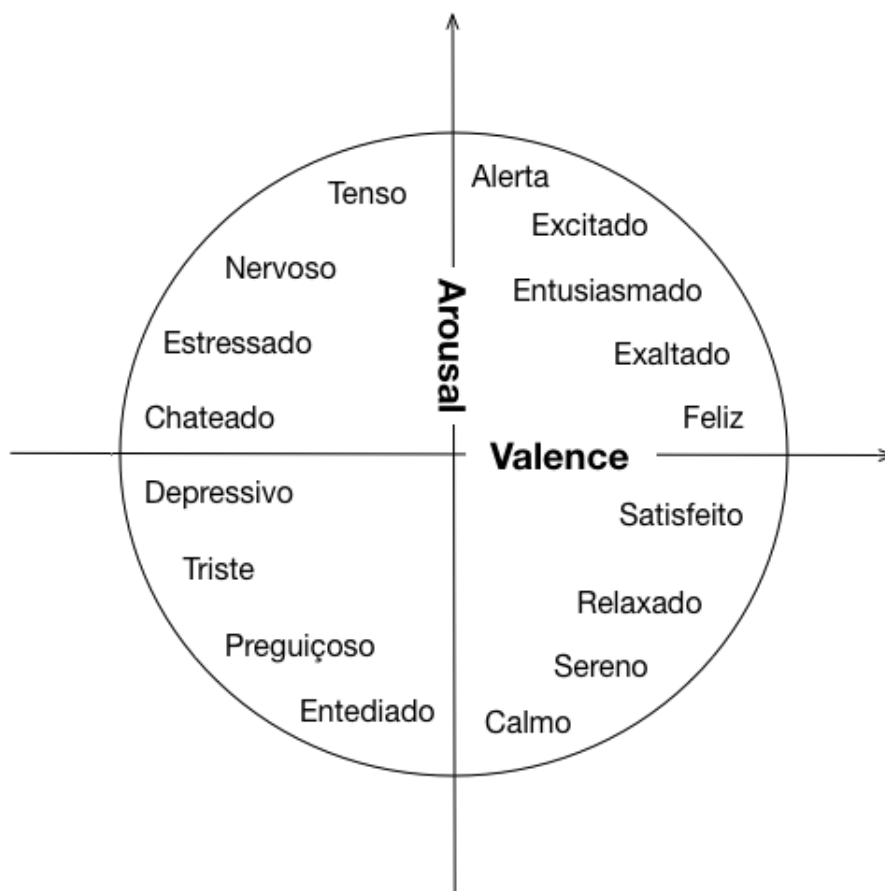
O modelo PAD foi construído levando em consideração 3 escalas ou traços quase ortogonais. A escala *Pleasure* ou *Valence* indica a predominância relativa de estados afetivos positivos ou negativos ao se realizar uma análise das respostas emocionais de um indivíduo para situações cotidianas.

Já na escala *Arousal*, é medido o grau de excitação por estímulos com um alto grau de informação e o quanto tempo demora para que esse retorne ao nível normal.

Por fim, na escala *Dominance* é medida a sensação de se sentir no controle ou de ser controlado e influenciado, por eventos ou por outros, que uma pessoa tem para circunstâncias que possam vir a surgir durante sua vida.

Essas três escalas independentes formam as dimensões do espaço emocional. Portanto, pode-se mapear emoções discretas nesse espaço de acordo com a resposta de uma pessoa para cada uma dessas escalas.

Figura 2.5 – Modelo emocional circumplexo proposto por Russell.



Fonte: Arquivo pessoal.

Pelo modelo PAD 8 emoções discretas podem ser definidas: Protegido, Satisfeito, Surpreso, Alegre, Triste, Despreocupado, Medo e Bravo. Assim, as emoções podem ser caracterizadas por:

- Protegido: um valor positivo para *Pleasure*, e um baixo valor de *Arousal* e *Dominance*;
- Satisfeito: um valor positivo para *Pleasure*, um valor baixo para *Arousal* e alto para *Dominance*;
- Surpreso: um valor positivo de *Pleasure*, alto valor de *Arousal* e baixo valor de *Dominance*;
- Alegre: um valor positivo *Pleasure*, alto valor de *Arousal* e alto valor de *Dominance*;
- Triste: um valor negativo de *Pleasure* e por um baixo valor de *Arousal* e de *Dominance*;
- Despreocupado: um valor negativo de *Pleasure*, alto valor de *Arousal* e por um baixo valor de *Dominance*;

- Medo: um valor negativo de *Pleasure*, alto valor de *Arousal* e baixo valor de *Dominance*;
- Bravo: um valor negativo de *Pleasure*, e por um alto valor de *Arousal* e de *Dominance*.

A tabela 2.2 apresenta a representação de cada emoção no espaço emocional PAD.

Tabela 2.2 – Emoções caracterizadas no modelo emocional PAD.

<b>Emoção</b>	<b>Pleasure</b>	<b>Arousal</b>	<b>Dominance</b>
Protegido	Positivo	Baixo	Baixo
Satisfeito	Positivo	Baixo	Alto
Surpreso	Positivo	Alto	Baixo
Alegre	Positivo	Alto	Alto
Triste	Negativo	Baixo	Baixo
Despreocupado	Negativo	Baixo	Alto
Medo	Negativo	Alto	Baixo
Bravo	Negativo	Alto	Alto

## 2.4 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNA) são representações matemáticas do funcionamento do cérebro humano. Uma RNA também pode ser definida como um massivo processador distribuído que trabalha de forma paralela, composto por unidades básicas de processamento chamadas de neurônios (HAYKIN, 2008).

Entre as principais características de uma RNA está a sua capacidade de generalização, ou seja, a capacidade de aprender representações ou classificar dados que não são previamente conhecidos. Outra característica é a sua não-linearidade, uma vez que os neurônios podem ser lineares ou não-lineares, o que proporciona à RNA solucionar problemas de alta complexidade.

### 2.4.1 Neurônio Artificial

Em (MCCULLOCH; PITTS, 1943), foi proposta a primeira modelagem de um neurônio artificial. Mas foi somente em (ROSENBLATT, 1957) que Frank Rosenblatt desenvolveu o modelo de neurônio artificial utilizado até hoje, chamado de *Perceptron*.

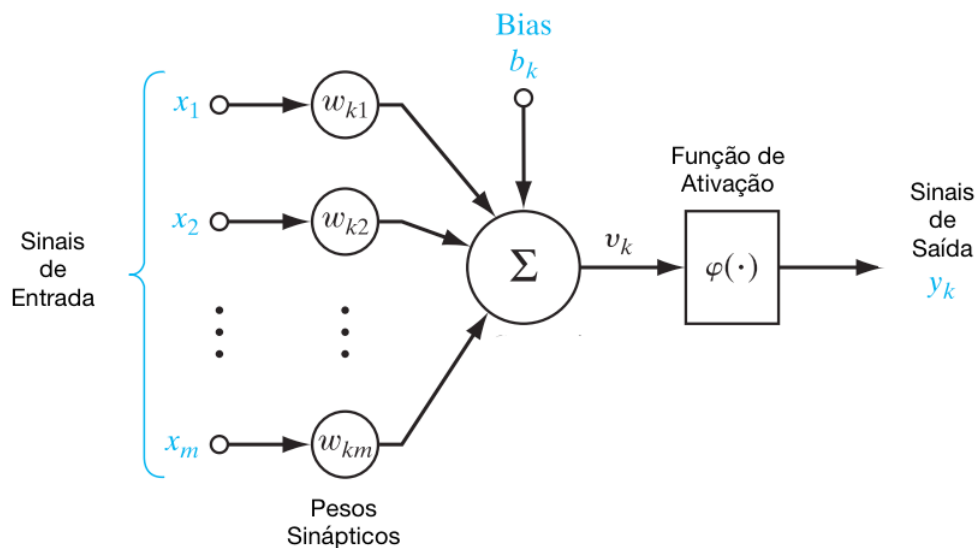
O neurônio artificial apresentado na Figura 2.6 é a unidade básica de uma rede neural. Nela é possível observar os 3 componentes principais que formam um neurônio artificial:

- Uma série de sinapses, multiplicadas por pesos que caracterizam as sinapses.

- Um somatório, que tem a função de somar todos os sinais de entrada, multiplicados pelos seus respectivos pesos para cada neurônio.
- E finalmente, uma função de ativação que tem o objetivo de limitar a amplitude de saída do neurônio.

O modelo neural descrito, também possui um bias, que tem a função de aumentar ou diminuir a entrada líquida da função de ativação (HAYKIN, 2008).

Figura 2.6 – Neurônio do tipo Perceptron.



Fonte: Adaptado de (HAYKIN, 2008).

No *Perceptron* apresentado na Figura 2.6, é possível visualizar a função de ativação  $\phi(v)$ , onde  $v_k$  se refere ao potencial de ativação do neurônio  $k$ ;  $w_{kj}$  ao peso sináptico da sinapse  $j$  pertencente ao neurônio  $k$ . A função de ativação  $\phi(v)$  é definida como uma das 3 formas a seguir:

- Função de Limiar ou função de Heaviside, onde:

$$\phi(v) = \begin{cases} 1, & \text{se } v_k \geq 0 \\ 0, & \text{se } v_k < 0 \end{cases} \quad (2.4)$$

- Função Linear por Partes, onde:

$$\phi(v) = \begin{cases} 1, & \text{se } v \geq +1/2 \\ v, & \text{se } +1/2 > v > -1/2 \\ 0, & \text{se } v \leq -1/2 \end{cases} \quad (2.5)$$

- Função Sigmoidal, onde:

$$\phi(v) = \frac{1}{1 + \exp(-av)} \quad (2.6)$$

O potencial de ativação  $v_k$  pode ser definido por

$$v_k = \sum_{j=1}^m w_{kj}x_j + b_k \quad (2.7)$$

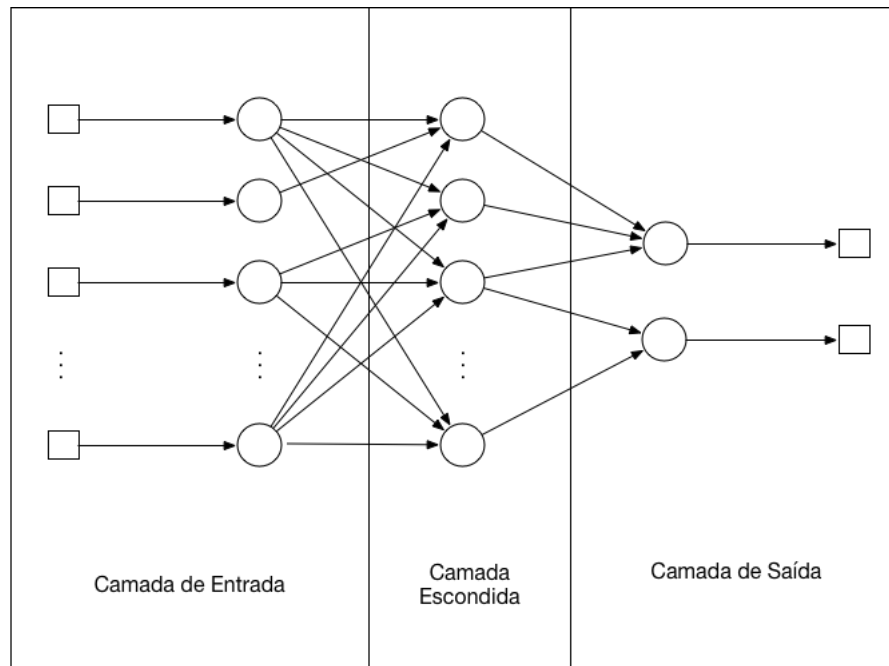
#### 2.4.2 Arquitetura de uma Rede Neural Artificial

A forma com que os neurônios são conectados entre si em uma RNA, definem estritamente como deverão ser implementados os algoritmos de aprendizado de máquina utilizados nessa rede.

Em geral, classificando de acordo com a arquitetura, existem três tipos de Redes Neurais:

- Redes *feedforward* de camada única: nessa arquitetura a RNA possui apenas uma camada, assim os neurônios de entrada são conectados diretamente nos neurônios de saída. O fluxo de informações ou *feedback* desse tipo de arquitetura é somente para frente, por isso o nome de *feedforward*. A Figura 2.7 apresenta um esquemático da arquitetura de uma RNA de camada única.

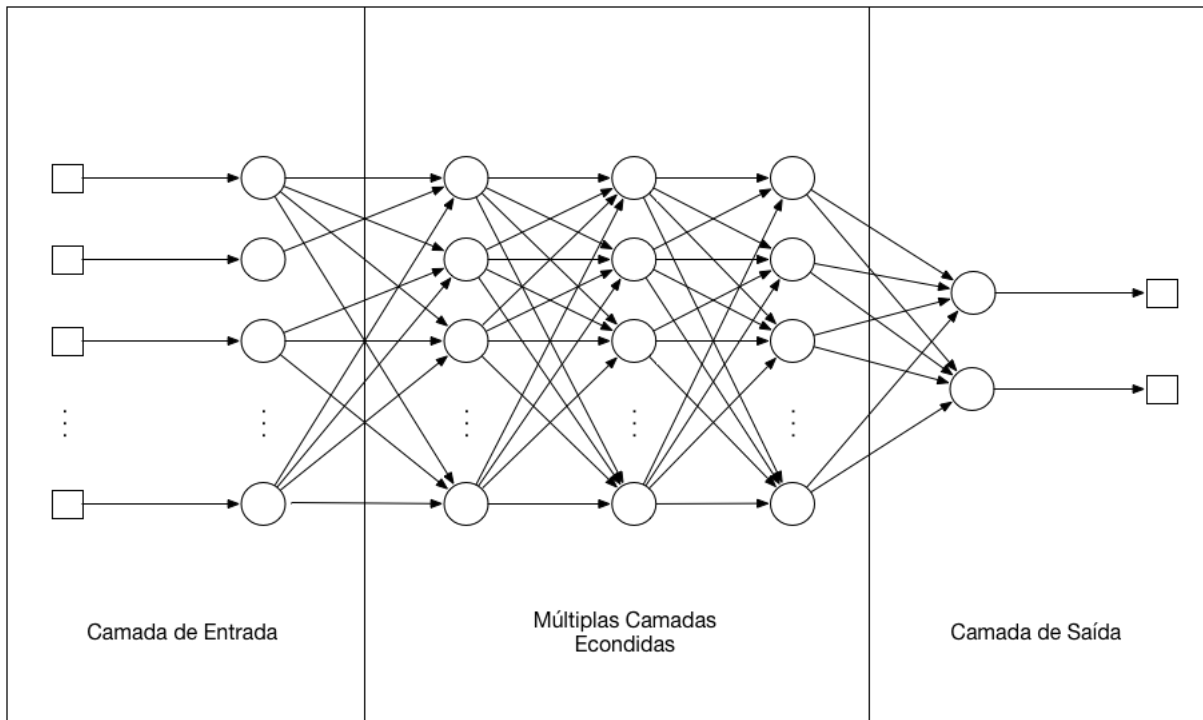
Figura 2.7 – Arquitetura de uma Rede Neural *Feedforward* de camada única.



Fonte: Arquivo pessoal.

- Redes *feedforward* de multicamada (*Multilayer Perceptron* ou MLP): nessa arquitetura, um novo tipo de camada intermediária entre os neurônios de entrada e os neurônios de saída é apresentada, chamada de *hidden layer* ou camada oculta. Portanto, ao se adicionar camadas intermediárias torna-se possível resolver problemas mais complexos. Os neurônios de entrada se conectam aos neurônios da camada oculta, e vão se conectando as  $n$  camadas ocultas que vierem a existir, até se conectarem aos neurônios da camada de saída. Esta arquitetura também apresenta o *feedback* somente para frente. Na Figura 2.8 pode ser observado um esquemático da arquitetura de uma RNA de múltiplas camadas escondidas.

Figura 2.8 – Arquitetura de uma Rede Neural *Feedforward* de múltiplas camadas.

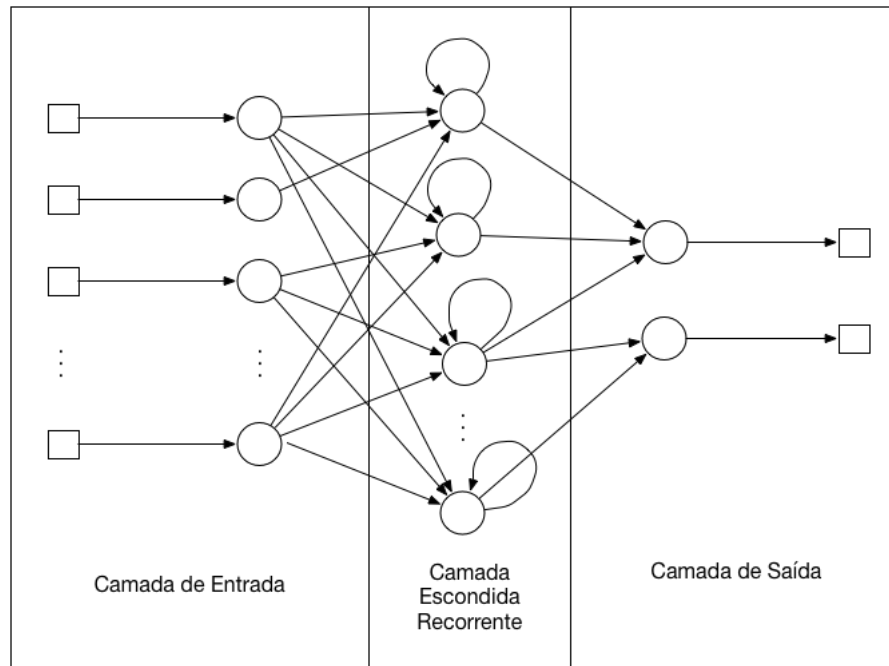


Fonte: Arquivo pessoal.

- **Redes Recorrentes:** nessa arquitetura passa a existir um novo tipo de fluxo de informações na rede, chamado de *feedback loop*. Assim, tanto em uma rede neural de camada única ou com multicamada podem conter neurônios que se conectem com neurônios da camada anterior, ou seja, um neurônio da camada de oculta pode vir a se conectar em outro neurônio da mesma camada ou até nele mesmo. Abaixo, a Figura 2.9 apresenta a arquitetura de uma Rede Neural Recorrente.



Figura 2.9 – Arquitetura de uma Rede Neural *Feedforward* Recorrente.



Fonte: Arquivo pessoal.

### 2.4.3 Formas de aprendizado

As Redes Neurais Artificiais podem ser classificadas de acordo com o tipo do seu algoritmo de aprendizado. Entre os principais tipos de aprendizado de máquina estão:

- **Aprendizado supervisionado:** nesse tipo de aprendizado, existe a figura do professor que funciona como um moderador para os questionamentos feitos ao neurônio. Assim, através da resposta correta da rede apresentada pelo professor em conjunto com a resposta apresentada pela rede, o algoritmo consegue calcular um valor de ajuste para os parâmetros da rede em função do valor obtido e do valor desejado.
- **Aprendizado não-supervisionado:** nessa forma de aprendizado não existe um professor, logo o aprendizado é realizado através da análise do comportamento estatístico dos dados.
- **Aprendizado semi-supervisionado:** neste tipo somente uma pequena porção dos dados apresentam sua classe conhecida previamente.

#### 2.4.4 *Back-Propagation* - Aprendizado por retropropagação

Entre as principais formas de aprendizado supervisionado aplicados em uma RNA, está o algoritmo de retropropagação de erro ou *error backpropagation*, que é baseado na regra de aprendizagem por correção de erro.

Esse algoritmo é formado, basicamente, por um avanço ou propagação e um retrocesso ou retro propagação. Nesse avanço um sinal aplicado na entrada é processado e propagado para as demais camadas, mantendo seu valor fixo para os pesos sinápticos da rede e gerando uma saída como resposta.

No processo de retro propagação, os pesos sinápticos são reajustados de acordo com uma regra de correção de erro, obtida pela subtração da resposta real pela resposta desejada. A partir desse cálculo, esse valor de erro é propagado para as camadas de trás. Esse processo de ajuste dos pesos sinápticos é denominado mecanismo de controle, cujo objetivo é aproximar a resposta atual  $y_k(n)$  da resposta desejada  $d_k(n)$ .

Essa aproximação é obtida através da redução do índice de desempenho  $\xi(n)$ , (HAYKIN, 2008) que é definido por:

$$\xi(n) = \frac{1}{2} e_k^2(n). \quad (2.8)$$

Esses ajustes continuam a acontecer até que a rede chegue a um estado estável e obtenha a resposta desejada, ou um número máximo de épocas ou iterações sejam atingidas, ou ainda, que um valor de resposta mínimo (*threshold*) seja obtido. O processo de redução do índice de desempenho descrito acima,  $\xi(n)$  compõe a regra de aprendizado denominada regra delta (WIDROW; HOFF, 1960). De acordo com a regra delta, o ajuste  $\Delta w_{kj}(n)$  aplicado ao peso sináptico  $w_{kj}$  no tempo  $n$  é definido por:

$$\Delta w_{kj}(n) = \eta e_k(n) x_j(n), \quad (2.9)$$

onde  $\eta$  determina a taxa de aprendizado e  $w_{kj}$  ao peso sináptico da sinapse  $j$  pertencente ao neurônio  $k$ . Uma vez que a regra delta pressupõe que o sinal de erro seja mensurável, é necessário que a resposta desejada seja previamente informada por uma fonte externa e esteja acessível ao neurônio  $k$ . (HAYKIN, 2008)

O ajuste dos pesos a partir da regra delta é denominado Regra Delta Generalizada (HAYKIN, 2008), e pode ser aferido por:

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \alpha[w_{ji}^{(l)}(n-1)] + \eta \delta_j^{(l)}(n) y_i^{(l-1)}(n). \quad (2.10)$$

## 2.5 Florestas Aleatórias

A pesquisa em algoritmos de agrupamento ou *ensemble* possibilitou uma melhora na acurácia dos modelos de classificação (BREIMAN, 2001).

O algoritmo de Florestas Aleatórias tem se tornado popular em vista da sua alta capacidade de generalização, podendo ser utilizado em tarefas de classificação e regressão. Em (KOTTI; STYLIANOU, 2017), o algoritmo é utilizado para a classificação de emoções em trechos de áudio retirados de filmes. Já em (ZHANG; MENG; LI, 2016) o algoritmo foi utilizado na classificação de emoções em músicas.

De modo geral, o algoritmo de Florestas Aleatórias ou *Random Forest* (RF) é uma combinação de árvores de decisão, onde cada árvore gerada depende de um subconjunto de dados amostrados de forma independente e que apresentem a mesma distribuição para todas as árvores que formam a floresta.

Pode-se definir o algoritmo como um classificador formado por árvores de decisão  $h(x, \Theta_k), k = 1$ , onde  $\Theta_k$  são vetores aleatórios independentes e identicamente distribuídos, no qual cada árvore vota na classe que a entrada  $x$  tem maior probabilidade de pertencer (BREIMAN, 2001).

Dado um conjunto de classificadores  $h_1(x), h_2(x), \dots, h_k(x)$ , e um conjunto de treinamento  $X$  selecionado aleatoriamente da distribuição de dados  $Y$ , segundo (NUGRAHAENI; MUTIJARSA, 2017) pode-se definir a função de margem como:

$$margem(X, Y) = media_k I(h_k(X) = Y) - max_{j \neq Y} media_k I(h_k(X) = j), \quad (2.11)$$

no qual a função  $I(\cdot)$  corresponde a função de indicação. A função margem calcula o quanto a média de votos da classe correta (para a relação  $X, Y$ ) é maior que a média dos votos para as outras classes. Assim, quanto maior for o valor de margem, maior é a certeza da classificação correta.

Pode-se definir a função para o cálculo do erro de classificação como:

$$PE^* = P_{X,Y}(\text{margem}(X,Y) < 0), \quad (2.12)$$

onde é calculada a probabilidade de  $X, Y$  em relação ao valor calculado pela função de margem.

O algoritmo de Florestas Aleatórias executa de forma eficiente para bases de dados grandes e com alto número de dimensões, sem que haja um pré-processamento para seleção de características (KULKARNI; SINHA, 2012). Entretanto, aumentar o número de árvores de decisões geradas pelo algoritmo não, necessariamente, irá implicar em uma melhora na acurácia do modelo. Portanto, testes empíricos devem ser realizados para definir a quantidade de árvores que serão geradas.

## 2.6 Máquinas de Vetor Suporte

O algoritmo de Máquinas de Vetor Suporte (SVM) ou *Support Vector Machines* (SVM) é aplicado em diversas áreas da ciência. Trabalhos como (ZHANG et al., 2015) e (YANG et al., 2012) utilizaram esta técnica para classificar emoções em dados extraídos de amostras de voz. Além da área de processamento de linguagem natural, SVM pode ser empregado na área médica. Em (TIBDEWAL; TALE, 2016) a técnica é utilizada para corretamente identificar a ocorrência de epilepsia. Já em (ORREGO; ORREGO, 2016), SVM é usado para classificar ruídos gerados por artefatos, como o ato de piscar, o movimento dos olhos, sinais gerados por contrações musculares (EMG) e sinais cardíacos (ECG), entre outros, dos sinais de um EEG.

Devida a propriedade de generalização proporcionada pelas SVM em conjunto a alta capacidade de modelar problemas não-lineares complexos, são empregadas na solução de problemas de classificação e regressão, além de problemas de previsão em séries temporais.

Nesse algoritmo, os dados de entrada são linearmente ou não-linearmente mapeados em um hiperplano de alta dimensionalidade (CHANG; LIN, 2011).

Dado um conjunto de entrada  $x_i \in R^n$ , onde  $i$  representa o número de elementos de entrada que varia de  $1, \dots, l$ , e um vetor de saída  $y \in R^l$  tal que  $y_i$  possa assumir os valores do conjunto  $\{-1, 1\}$ ; o algoritmo resolve o seguinte problema de otimização:

$$\min(1/2w^T w + C \sum_{i=1}^l \xi_i), \quad (2.13)$$

sujeita à restrição:

$$y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i, \xi_i \geq 0, \quad (2.14)$$

onde  $\phi(x_i)$  mapeia  $x_i$  em um hiperplano.  $C$  é uma constante positiva, que corresponde ao parâmetro de otimização; esse é responsável pelo balanço entre a capacidade de classificação e a simplicidade do modelo. A definição de  $C$  influencia diretamente a capacidade de generalização do modelo gerado (CORTES; VAPNIK, 1995). Se o valor de  $C$  definido for muito pequeno, fará com que as fronteiras de decisão formadas pelo hiperplano sejam mais suaves. Para um valor alto, o modelo utilizará uma maior quantidade de dados como vetores de suporte, aumentando a sua complexidade.

Uma vez que a variável  $w$  pode apresentar mais de uma dimensão, pode-se formular o seguinte problema de otimização:

$$\min(1/2\alpha^T Q\alpha - e^T \alpha) \quad (2.15)$$

sujeito a restrição:

$$y^T \alpha = 0, 0 \leq \alpha_i \leq C \quad (2.16)$$

onde  $e$  é a transposta de um vetor de uns ( $e = [1, 1, 1, \dots, 1]^T$ ), e  $Q$  é uma matriz semi-definida positiva de tamanho  $l \times l$ , calculada por:

$$Q_{ij} \equiv y_i y_j K(x_i, x_j) \quad (2.17)$$

A função  $K(\cdot)$  é conhecida como função *kernel*. Escolher o melhor *kernel* para solucionar o problema desejado corresponde a um dos parâmetros a serem definidos pelo projetista do modelo. (Chih-Wei Hsu, Chih-Chung Chang; LIN, 2008) recomenda que, primeiro, se utilize o *kernel Radial Basis Function* (RBF), pois o mesmo consegue mapear dados que não são linearmente separáveis e apresentam uma menor complexidade computacional quando comparado, por exemplo, ao *kernel* polinomial. O *kernel* RBF pode ser definido como:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0 \quad (2.18)$$

onde  $\gamma$  corresponde ao grau de influência que uma única amostra de treinamento pode ter, podendo ser considerado como a inversa do raio de influência de uma RBF. Logo, quanto maior for o valor de  $\gamma$  menor será sua influência; e quanto menor for o valor de  $\gamma$  maior ela será.

Já o *kernel* polinomial pode ser definido como:

$$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0. \quad (2.19)$$

Existem situações nas quais a utilização do *kernel* RBF não é justificada, como por exemplo, quando os dados apresentam uma alta dimensionalidade. Para este caso, (Chih-Wei Hsu, Chih-Chung Chang; LIN, 2008) recomenda a utilização do *kernel* linear. O *kernel* Linear é definido como:

$$K(x_i, x_j) = x_i^T x_j \quad (2.20)$$

A Figura 2.10 apresenta um exemplo com a implementação de 3 modelos SVM, cada um utilizando um tipo de *kernel* diferente, sendo possível verificar a diferente fronteira de classificação formada por cada um desses.

Uma vez otimizado o problema de minimização definido na equação 2.15, obtém-se a equação geral de classificação:

$$y_i(w^T \phi(x_i) + b) = \text{sgn}\left(\sum_{i=1}^l y_i \alpha_i K(x_i, x) + b\right) \quad (2.21)$$

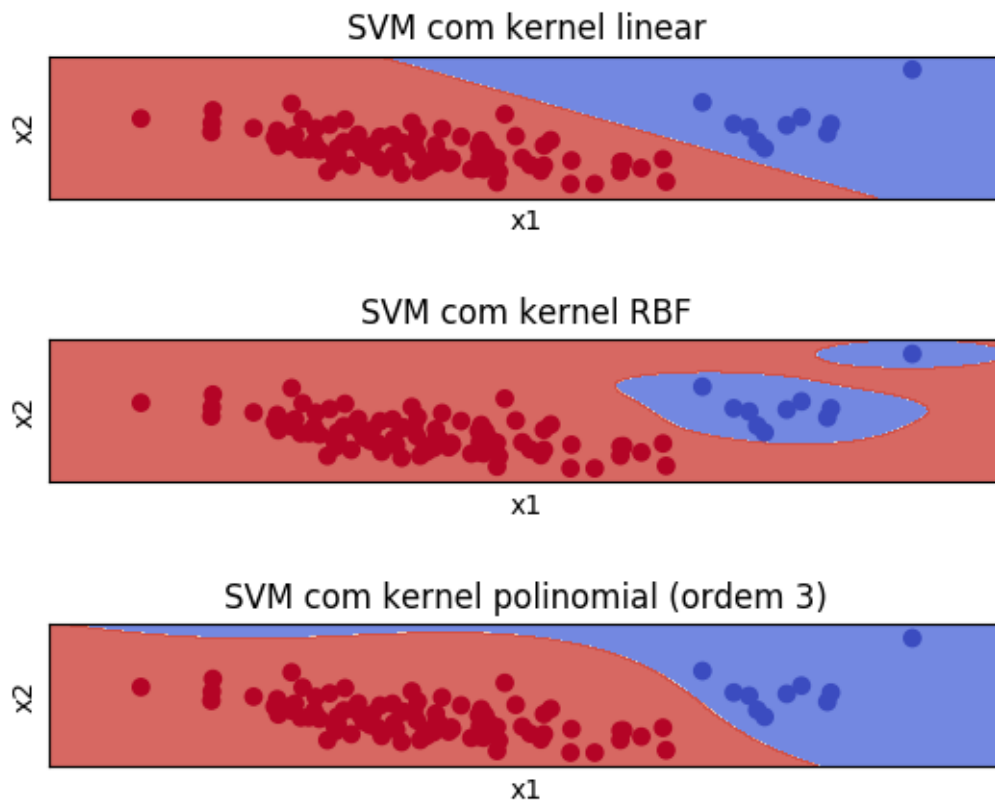
## 2.7 Aprendizado de Máquina em Dados Não-balanceados

Segundo (HE; GARCIA, 2009), pode-se definir como dados não-balanceados qualquer conjunto de dados em que a distribuição de amostras para cada classe presente não é igual. Dados não balanceados podem influenciar negativamente a acurácia de um modelo de classificação.

Em classificação de emoções utilizando dados coletados através de um EEG, a grande maioria dos datasets disponíveis, como o DEAP (KOELSTRA et al., 2012), apresentam os dados de forma não balanceada. Entre as técnicas mais utilizadas para resolver esse problema estão os algoritmos SMOTE e ADASYN.

Em (PEREIRA; GOMES, 2016) é verificada a importância do processo de balanceamento de dados em modelos para classificação de emoções em sinais de EEG, no qual são

Figura 2.10 – Exemplo de SVM com diferentes kernels.



Fonte: Arquivo pessoal.

apresentados resultados que reforçam a utilização deste tipo de abordagem para melhorar os modelos de classificação. Não somente restrito à área de classificação de emoções, em (SHO-ORANGIZ; WEDDELL; JONES, 2016) os algoritmos SMOTE e ADASYN são utilizados em conjunto com técnicas de aprendizado de máquina para a identificação de *microsleeps*.

O não balanceamento dos dados pode vir a acarretar problemas para técnicas de aprendizado de máquina lineares e não-lineares, como SVM e Redes Neurais Artificiais. Essas técnicas podem vir a se tornar enviesadas ao classificar amostras para a classe majoritária, uma vez que levam em conta o valor do erro para o ajuste do modelo de classificação.

Para solucionar o problema, várias técnicas foram desenvolvidas baseadas na manipulação de amostragens criadas a partir da base de dados originais. Entre essas pode-se destacar as técnicas de *Random Oversampling* e *Undersampling*, SMOTE (CHAWLA et al., 2002) e ADASYN (HE et al., 2008).

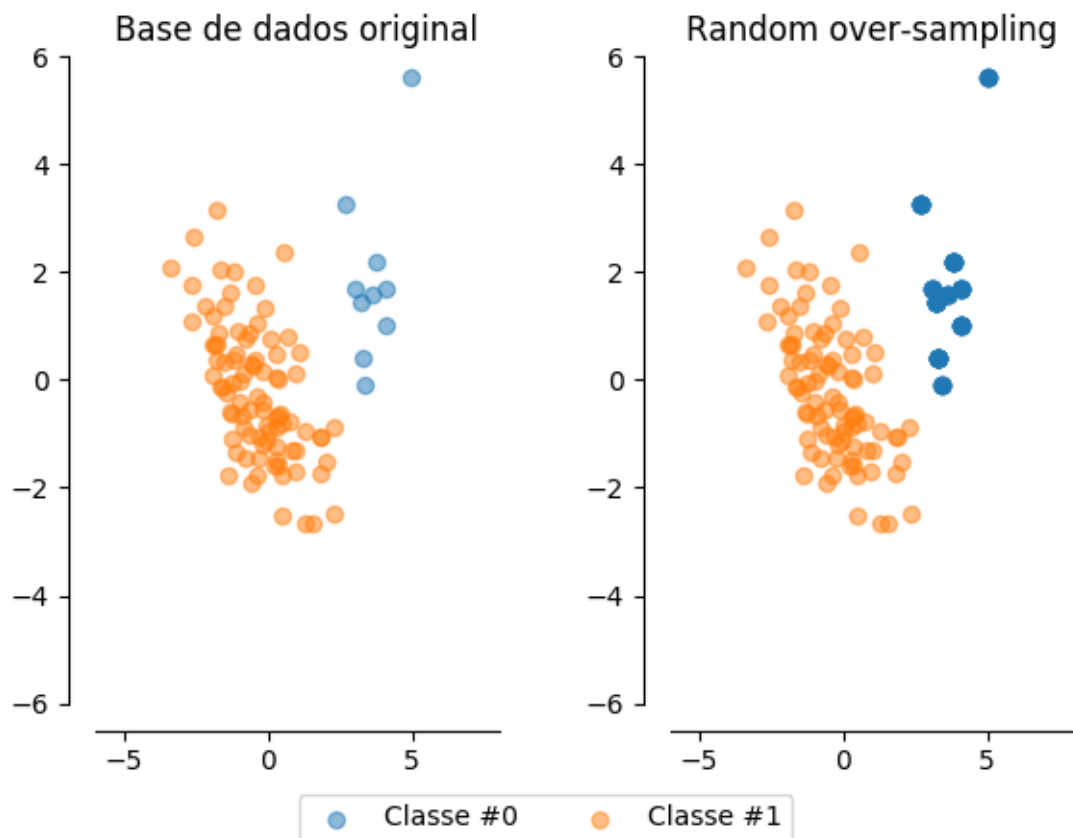
### 2.7.1 *Random Oversampling e Undersampling*

O funcionamento do algoritmo de *Oversampling* aleatório pode ser descrito como o ato de se adicionar, aleatoriamente, amostras à classe minoritária até se atingir o balanço desejado.

Segundo (HE; GARCIA, 2009), dado um conjunto de amostras  $E$  formado pelas amostras da classe minoritária que compõe uma base de dados  $B$ , são selecionados  $S_{min}$  amostras de  $E$  para ser replicadas e adicionadas a um conjunto  $S$ . Por fim, esse conjunto  $S$  é adicionado a base de dados original  $B$ .

A Figura 2.11 apresenta um exemplo da técnica de *Oversampling*, onde no primeiro gráfico pode ser visualizado as amostras originais da base de dados. No segundo gráfico, em azul escuro, são exibidas as amostras da classe minoritária (Classe #0) que foram duplicadas.

Figura 2.11 – Exemplo da técnica Oversampling.



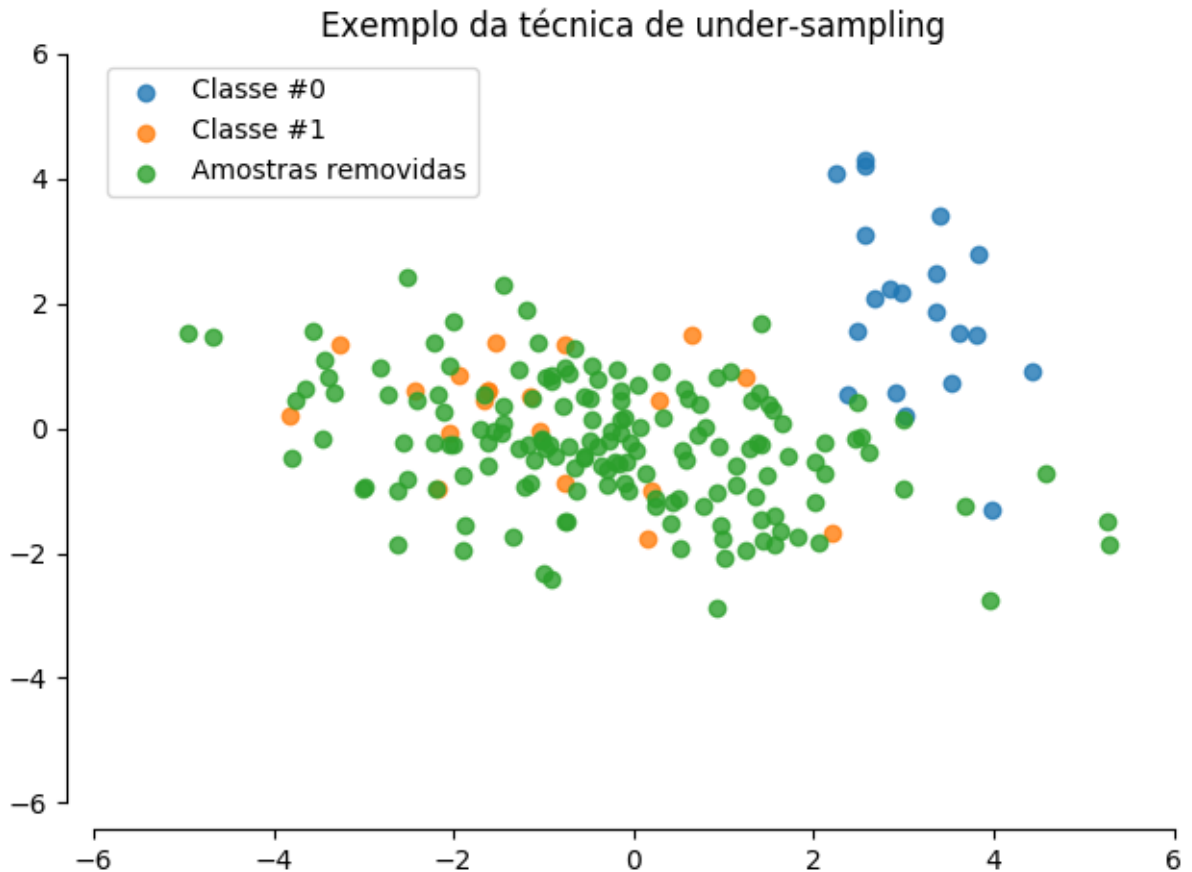
Fonte: Arquivo pessoal.

A técnica de *Undersampling* remove amostras do conjunto original, ao contrário da de *Oversampling*. Assim, são removidas amostras da classe majoritária até que se obtenha equilíbrio entre classes.



A Figura 2.12 apresenta uma exemplificação da técnica de *Undersampling*. Em verde, verifica-se as amostras que foram removidas da base de dados original pertencentes a classe majoritária (Classe #1).

Figura 2.12 – Exemplo da técnica Undersampling.



A utilização dessa abordagem para a solução do problema de base de dados não-balanceados pode apresentar problemas. No caso da técnica de *Oversampling*, ao aumentar o número de amostras replicando dados existentes pode ocorrer *overfitting*. Dessa maneira o modelo passará a decorar as amostras apresentando uma péssima capacidade de generalização, observada pela falta de acurácia na classificação de amostras que não participaram do treinamento.

No caso da técnica de *Undersampling* ocorrem problemas de origem diferente. O modelo de aprendizado de máquina pode vir a não aprender um certo comportamento dos dados, em vista que são retiradas amostras do conjunto de dados original.

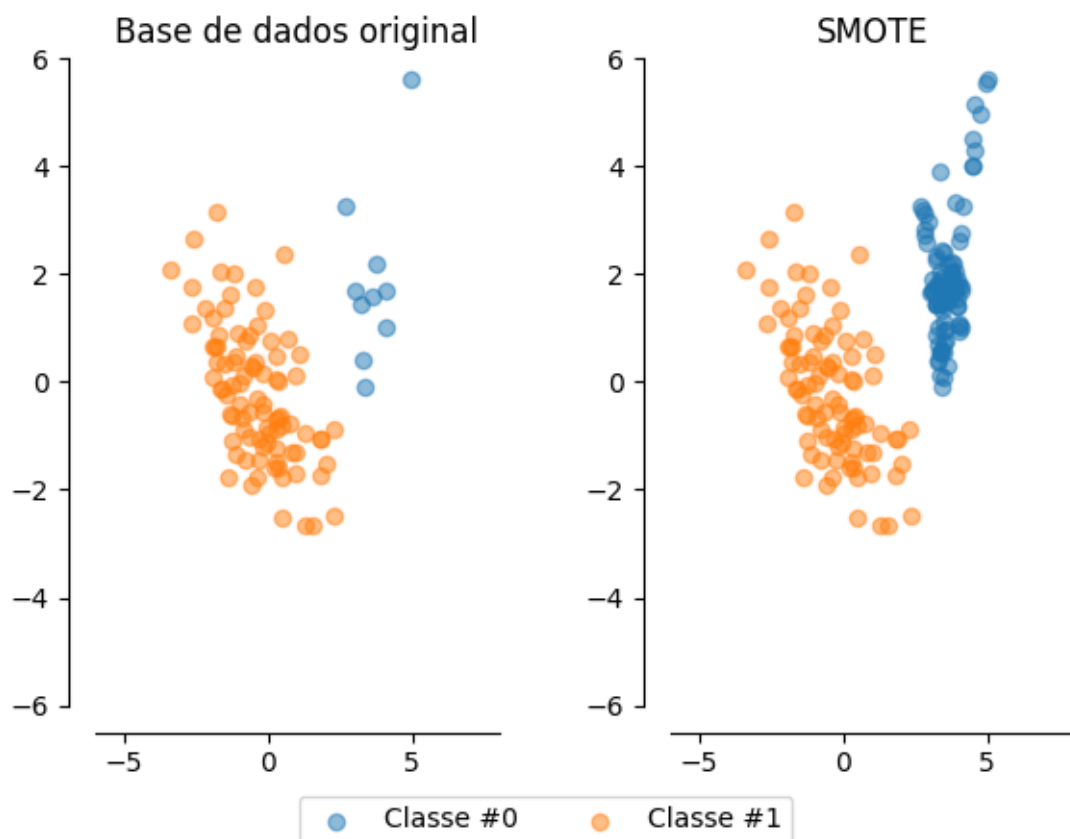
## 2.7.2 SMOTE

Uma outra forma de se balancear uma base de dados é a criação de amostras sintéticas. Entre as técnicas mais utilizadas está o algoritmo *Synthetic Minority Oversampling Technique* (SMOTE) (CHAWLA et al., 2002). Nesse algoritmo são criadas amostras sintéticas ou artificiais baseadas nas amostras reais que formam a base de dados original.

Para um subconjunto  $S_{min}$  formado pelos dados da classe minoritária, são consideradas as  $K$  amostras que estão mais próximas a uma amostra  $x_i \in S_{min}$ . Essas  $K$  amostras são calculadas pelo algoritmo *K-nearest Neighbors* (KNN) (ALTMAN, 1992). Desse modo, para se criar uma nova amostra sintética, basta multiplicar os valores dessas amostras calculadas pelo KNN por um peso aleatório entre 0 e 1.

A Figura 2.13 mostra um exemplo do algoritmo SMOTE. Verifica-se o aumento das amostras da classe minoritária (Classe #0), em azul.

Figura 2.13 – Exemplo da técnica SMOTE.



Fonte: Arquivo pessoal.

Por utilizar o algoritmo KNN para a criação de novas amostras, deve-se definir o parâmetro  $K$  anteriormente a execução do algoritmo.

A não consideração de amostras de outras classes que estão próximas à fronteira de decisão pode acarretar na sobreposição entre classes, tornando o problema mais complexo.

### 2.7.3 ADASYN

O algoritmo *Adaptive Synthetic Sampling* (ADASYN) (HE et al., 2008) surgiu tentando solucionar os problemas apresentados pelo algoritmo SMOTE, ou seja, minimizar a possibilidade de sobreposição de classes.

Desse modo, o algoritmo utiliza um método sistemático para criar de forma adaptável amostras sintéticas pela análise de suas distribuições. Primeiramente é calculado o número de amostras que deverão ser criadas por:

$$G = (|S_{maj} - S_{min}|)\beta, \quad (2.22)$$

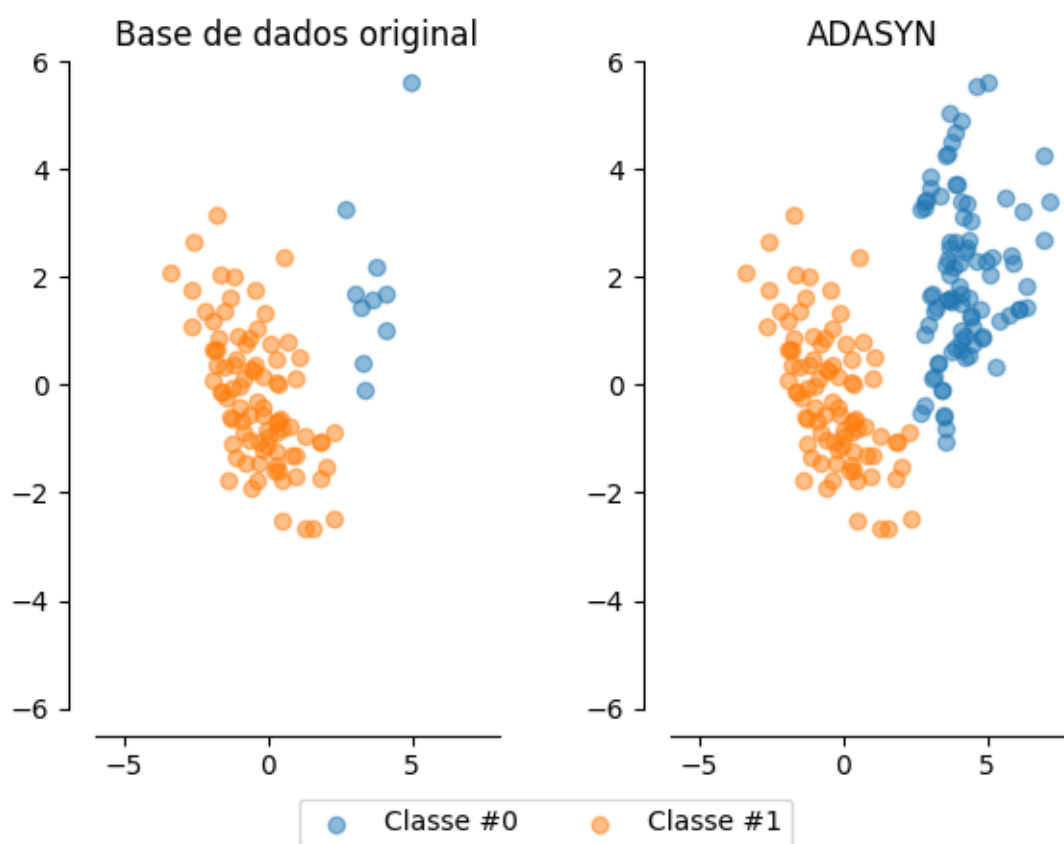
onde  $\beta$  corresponde ao parâmetro de balanço que se deseja obter. Em seguida é calculado o valor de  $\Pi_i$  para cada  $x_i \in S_{min}$ , utilizando o algoritmo KNN:

$$\Pi_i = \frac{\Delta_i/K}{Z}, i = 1, \dots, |S_{min}|, \quad (2.23)$$

onde  $\Delta_i$  representa o número de amostras selecionadas pelo KNN. Por fim, é calculado o número de amostras sintéticas que serão geradas:

$$g_i = \Pi_i G \quad (2.24)$$

Figura 2.14 – Exemplo da técnica ADASYN.



Por limitar a criação dos dados, o algoritmo ADASYN não aumenta o número de amostras na fronteira de decisão. Essa limitação, contribui para diminuir a possibilidade de haver sobreposição de classes. A Figura 2.14 apresenta um exemplo do algoritmo ADASYN. Pode-se notar uma melhor distribuição dos dados gerados, quando comparados ao algoritmo SMOTE apresentado na Figura 2.13.

## 2.8 Análise de Componentes Principais

O algoritmo de Análise de Componentes Principais (PCA) foi criado em 1901 por Karl Pearson em (PEARSON, 1901) e é um dos algoritmos de extração de características e redução do número de dimensões de uma base de dados mais utilizados (SUBASI; GURSOY, 2010).

O principal objetivo desse algoritmo é encontrar uma representação dos dados de dimensão  $d$  em um espaço de menor número de dimensões. O algoritmo se baseia na rotação das coordenadas do sistema, até que se obtenha a identificação de novas direções ou componentes principais no qual a variância dos dados seja máxima (RINGNÉR, 2008).

Conforme (MAHAJAN, 2011), dado um vetor  $x_t (t = 1, \dots, l)$  e  $\sum_{t=0}$ , em que cada um deles possui  $m$  dimensões,  $x_t = (x_t(1), x_t(2), \dots, x_t(m))^T$ , onde  $m < l$ , o PCA aplica uma transformação linear para cada vetor  $x_t$ , em um novo vetor  $s_t$  conforme a equação 2.25 onde  $U$  é uma matriz ortogonal  $m \times m$  cuja coluna  $u_i$  é um autovetor da matriz de covariância  $C$ , definida em 2.26.

$$s_t = U^T x_t. \quad (2.25)$$

$$C = 1/l \sum_{t=1}^l x_t x_t^T. \quad (2.26)$$

O PCA primeiramente obtém os autovalores por:

$$\lambda_i u_i = C u_i = 1, \dots, m \quad (2.27)$$

onde  $\lambda_i$  são os autovalores de  $C$  e  $u_i$  é o autovetor correspondente. Logo, pelos autovetores  $u_i$ , os componentes de  $s_t$  são calculados como transformações ortogonais de  $x_t$  conforme:

$$s_t = u_i^T x_t, i = 1, \dots, m \quad (2.28)$$

Os novos componentes são chamados de componentes principais e são ordenados conforme a sua variância. Portanto, o primeiro componente principal  $s_1$  tem a maior representação dos dados, e assim por diante.

## 2.9 Matriz de Confusão

Uma das métricas mais utilizadas na área de aprendizado de máquina é a matriz de confusão. Nessa matriz, cada coluna representa instâncias das classes reais, enquanto as linhas representam instâncias que foram classificadas pelos algoritmos de aprendizado de máquina.

Em problemas binários, essa matriz é composta por 2 linha e 2 colunas. A Tabela 2.3 apresenta uma matriz de confusão para problemas binários, onde pode-se observar os componentes que a formam.

O valor de *Recall* ou *Sensitivity* é relativo a proporção dos dados positivos reais que são previstos ou classificados como positivos. Pode-se calcular o valor de *Recall* como:

$$Recall = \frac{TP}{P} \quad (2.29)$$

O valor de *Precision* é calculado pela proporção dos casos que são classificados como positivos que são realmente positivos, conforme a equação abaixo.

$$Precision = \frac{TP}{PP} \quad (2.30)$$

Outra métrica que pode ser calculada é o valor de *Fallout*. Que é obtida pela proporção das amostras que negativas que foram corretamente classificadas como negativas. Na equação abaixo é apresentado a equação para o cálculo, onde *FP* são os valores de falsos positivos e *RP*.

$$Fallout = \frac{FP}{PP} \quad (2.31)$$

Tabela 2.3 – Matriz de confusão

	Positivo	Negativo	
Positivo	TP (Verdadeiros Positivos)	FN (Falsos Negativos)	P (Total de Positivos)
Negativo	FP (Falsos Positivos)	TN (Verdadeiros Negativos)	N (Total de Negativos)
	PP (Positivos Previstos)	PN (Negativos Previstos)	1

### 2.9.1 Receiver Operating Characteristic

Uma análise que pode ser realizada a partir da matriz de confusão é o cálculo da curva ROC (*Receiver Operating Characteristic*). A gráfico da curva ROC é formado pelo valor Recall como eixo Y e por *Fallout* como eixo X.

Desse modo, um classificador perfeito teria uma curva posicionada no canto superior esquerdo do primeiro quadrante, ou seja, apresentaria um alto valor de *Recall* e baixo valor de *Fallout*. Já um classificador aleatório, com acurácia em torno dos 50%, apresentaria uma curva ROC próxima a diagonal positiva.

### 3 MATERIAIS E MÉTODOS

Este capítulo descreve os materiais e métodos utilizados para o desenvolvimento dessa dissertação. A priori é apresentada a base de dados, com informações a respeito dos dados e de como foram coletados. Em sequência, é apresentada a metodologia utilizada para o desenvolvimento dos modelos inteligentes, descrevendo como foram utilizadas as técnicas de aprendizado de máquina em conjunto com algoritmos de balanceamento de dados e seleção de características, para a classificação de emoções discretas. Por fim, são apresentados os indicadores de performance utilizados para qualificar a eficiência dos modelos propostos em solucionar o problema em questão.

A Figura 3.1 apresenta o esquemático da metodologia proposta neste trabalho.

#### 3.1 Base de Dados DEAP

Para o desenvolvimento dessa dissertação foi utilizada a base de dados DEAP (KOELSTRA et al., 2012). Os dados foram coletados através de experimentos onde 32 participantes tiveram seu EEG e sinais fisiológicos gravados enquanto assistiam a 40 vídeos, com duração de 1 minuto cada.

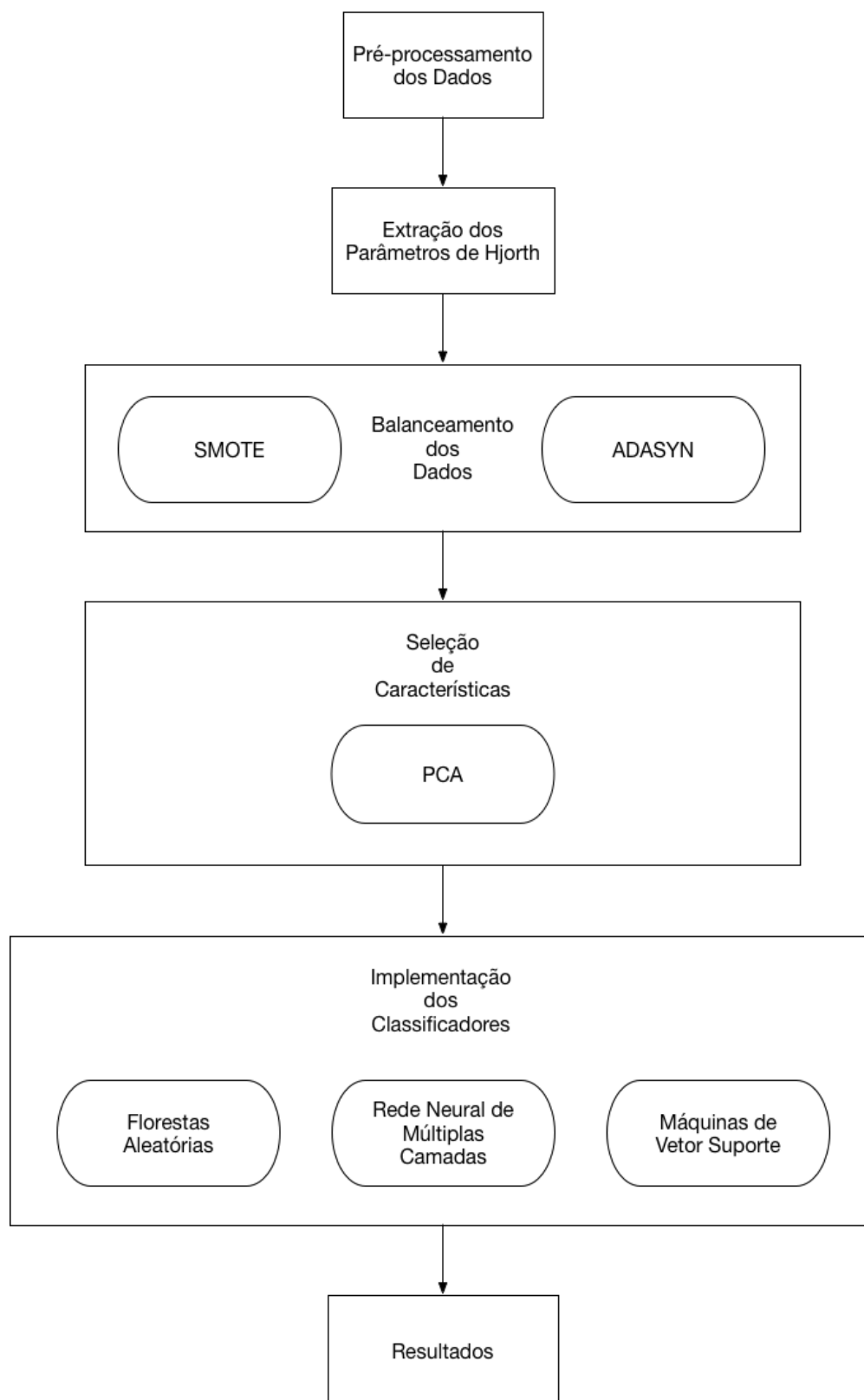
Após esse processo os participantes do experimento avaliaram cada um dos vídeos de acordo com valores de *Arousal*, *Valence*, *Dominance*, familiaridade e o quanto gostaram de cada vídeo. Essa avaliação foi realizada através de um questionário do tipo SAM (*Self Assessment Manikin*). Entretanto, nessa dissertação foram levados em consideração somente os valores de *Arousal*, *Valence* e *Dominance*, em vista que essas variáveis são suficientes para caracterizar uma emoção no modelo emocional PAD.

A Tabela 3.1 descreve o experimento realizado por (KOELSTRA et al., 2012), contendo informações sobre as escalas para os valores anotados pelos participantes em cada um dos vídeos. Da mesma forma, são apresentadas informações sobre os sinais fisiológicos e EEG que foram gravados.

##### 3.1.1 Modelagem das classes conforme o modelo emocional PAD

Uma vez que o intuito dessa dissertação é classificar emoções discretas, inicialmente foi necessária a transformação dos valores de *Arousal*, *Valence* e *dominance* em emoções discretas de acordo com o modelo PAD.

Figura 3.1 – Esquemático da metodologia proposta.



Fonte: Arquivo pessoal.

Nesse modelo emocional, os valores de *Arousal*, *Valence* e *Dominance* são dimensões que compõem um espaço emocional. Portanto, é possível mapear esses valores como coor-



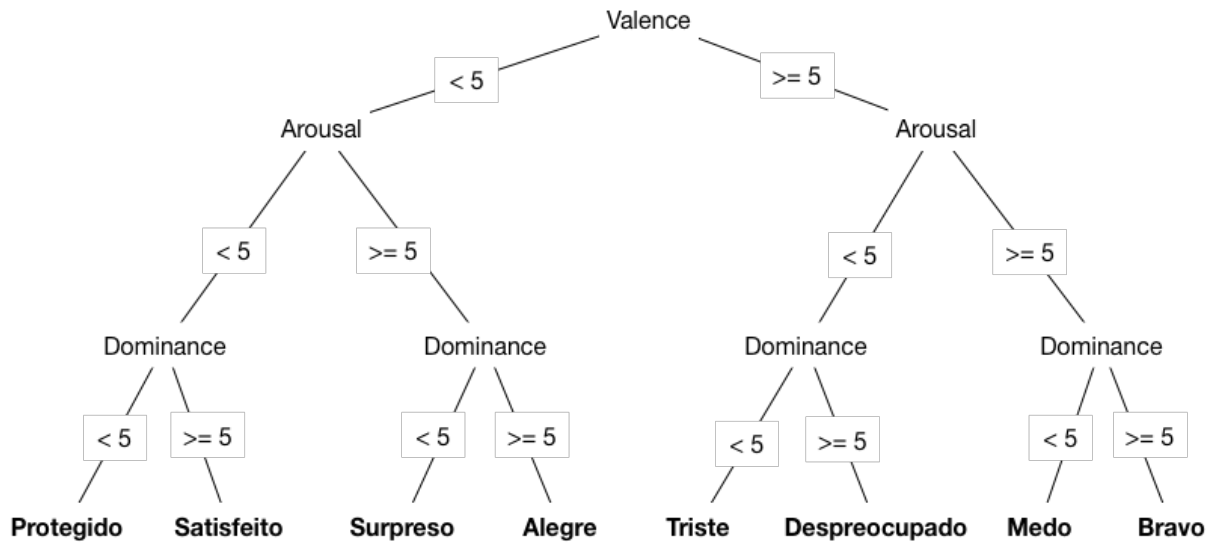
Tabela 3.1 – Descrição do experimento psicológico realizado em (KOELSTRA et al., 2012).

Experimento Psicológico - DEAP	
Número de Participantes	32
Número de Vídeos	40
Método de seleção dos vídeos	Subconjunto de vídeos classificados de forma online com classificações bem definidas
Métricas anotadas	<i>Arousal, Valence, Dominance</i>
Escala	Escala contínua: 1-9
Sinais gravados	EEG com 32 canais e 512Hz

denadas desse espaço possibilitando delimitar regiões no espaço como correspondentes a uma determinada emoção discreta. A Figura 3.2 apresenta uma árvore de decisão que representa a tabela de decisão para as emoções discretas. Essa árvore de decisão, foi criada utilizando as definições das emoções caracterizadas pelo modelo PAD em conjunto com a escala utilizada na base de dados DEAP. Logo, intervalos de *Arousal*, *Valence* e *Dominance* foram definidos para cada uma das emoções:

- Protegido:  $Valence = [1, \dots, 5[$ ;  $Arousal = [1, \dots, 5[$  e  $Dominance = [1, \dots, 5[$ ;
- Satisfeito:  $Valence = [1, \dots, 5[$ ;  $Arousal = [1, \dots, 5[$  e  $Dominance = [5, \dots, 9]$ ;
- Surpreso:  $Valence = [1, \dots, 5[$ ;  $Arousal = [5, \dots, 9]$  e  $Dominance = [1, \dots, 5[$ ;
- Alegre:  $Valence = [1, \dots, 5[$ ;  $Arousal = [5, \dots, 9]$  e  $Dominance = [5, \dots, 9]$ ;
- Triste:  $Valence = [5, \dots, 9]$ ;  $Arousal = [1, \dots, 5[$  e  $Dominance = [1, \dots, 5[$ ;
- Despreocupado:  $Valence = [5, \dots, 9]$ ;  $Arousal = [1, \dots, 5[$  e  $Dominance = [5, \dots, 9]$ ;
- Medo:  $Valence = [5, \dots, 9]$ ;  $Arousal = [5, \dots, 9]$  e  $Dominance = [1, \dots, 5[$ ;
- Bravo:  $Valence = [5, \dots, 9]$ ;  $Arousal = [5, \dots, 9]$  e  $Dominance = [5, \dots, 9]$ .

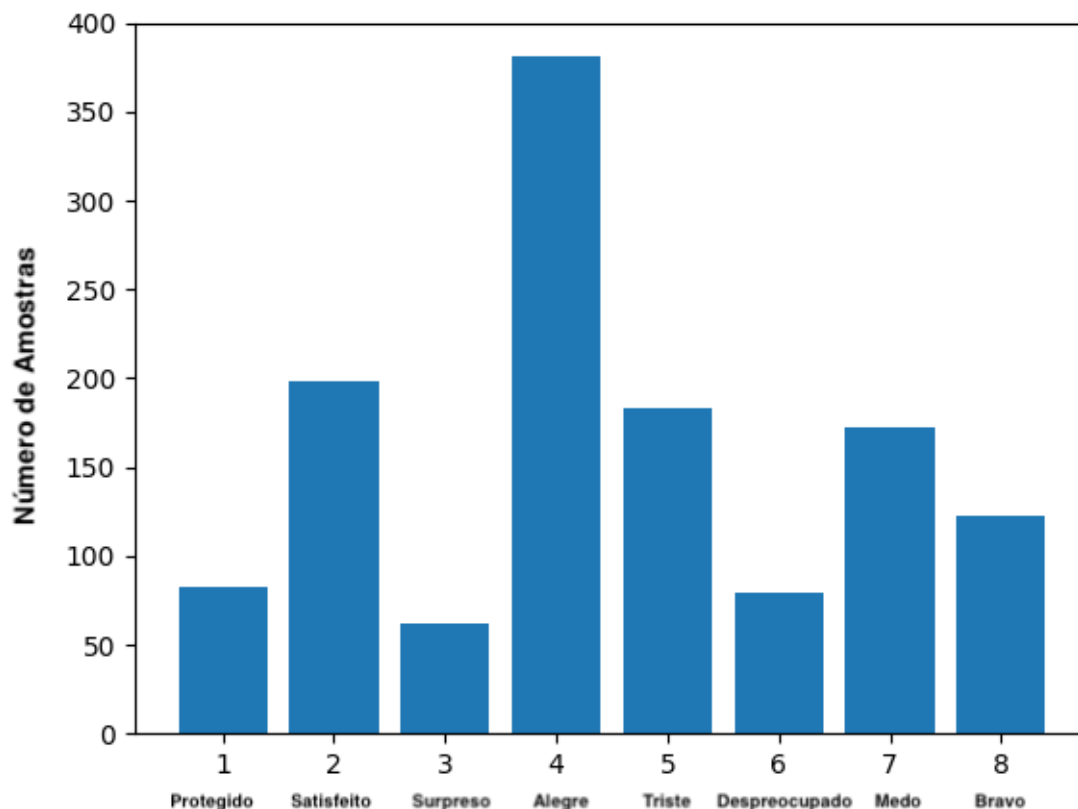
Figura 3.2 – Árvore de decisão para a classificação das amostras em emoções discretas.



Fonte: Arquivo pessoal.

Após transformar os valores de *Arousal*, *Valence* e *Dominance* em classes que representam emoções discretas, foi realizada uma análise da distribuição de amostras para cada uma das classes. A Figura 3.3 mostra a distribuição das amostras da base de dados para cada uma das emoções. Verifica-se que as classes estão desbalanceadas, o que justifica a utilização de um algoritmo de balanceamento de dados.

Figura 3.3 – Distribuição dos dados em função das classes.



Fonte: Arquivo pessoal.

### 3.2 Etapa de Balanceamento dos Dados

O balanceamento dados é um processo muito utilizado em base de dados compostos por sinais de EEG. Dentre as principais técnicas utilizadas estão os algoritmos SMOTE e ADASYN, onde ambos utilizam o processo de criação de amostras sintéticas a partir dos dados originais para aumentar o número de amostras para cada classe.

Por se tratar de um problema complexo, o problema de classificação multi classes foi transformado em 8 problemas de classificação binária, cada um com o objetivo de classificar uma única emoção, no formato *One vs All*, visando melhorar o desempenho dos modelos de classificação. Entretanto, ao transformar os problemas em problemas de classificação binária, o desbalanceamento dos dados não só permanece como aumenta a diferença da distribuição dos dados. Logo, os algoritmos SMOTE e ADASYN foram implementados em Python, utilizando a biblioteca Imblearn.

Para verificar qual algoritmo é o mais apropriado para o problema de classificação de emoções discretas, durante a implementação das técnicas de aprendizado de máquina ambos algoritmos de balanceamento serão testados. Desse modo, durante a análise comparativa dos resultados poderá ser definido o melhor algoritmo.

### 3.3 Etapa de Extração e Seleção de Características

Cada uma das amostras presentes na base de dados DEAP são formadas por 32 canais, referentes aos eletrodos do EEG. Para cada um desses canais serão extraídos os parâmetros de Hjorth (Atividade, Mobilidade e Complexidade) totalizando 96 características para cada amostra.

Após extraídas, o algoritmo PCA será implementado para selecionar as melhores características. Será utilizada a biblioteca Scikit-Learn, em Python. Para selecionar o melhor número de componentes principais serão realizados testes variando-se o número de componentes principais, onde serão coletados os valores de variância explicada.

### 3.4 Implementação dos Classificadores

Nesta etapa ocorrerá a implementação dos modelos utilizando as técnicas de aprendizado de máquina RNA, Florestas Aleatórias e SVM. Todas as técnicas serão implementadas em Python, utilizando a biblioteca Scikit-Learn. O problema de classificação de emoções discretas foi dividido em 8 problemas binários, ou seja, cada emoção será classificada em um modelo diferente. Assim, para cada técnica de aprendizado de máquina, serão implementados 8 modelos. Entretanto, os algoritmos de balanceamento de dados ADAYN e SMOTE também serão verificados, totalizando 16 modelos para cada técnica de aprendizado de máquina.

Para o processo de treinamento dos modelos serão utilizados 80% da base de dados DEAP. Os 20% restantes serão utilizados para os testes.

Conforme apresentado no referencial teórico, todas as 3 técnicas apresentam parâmetros que devem ser definidos antes da execução. Para a implementação dos modelos utilizando RNA serão realizados testes empíricos a fim de definir o número de camadas escondidas, o número de neurônios por camada, o valor da taxa de aprendizado e de *momentum*, e por fim, o número de épocas.

Para a implementação dos modelos utilizando Florestas Aleatórias, o único parâmetro que deverá ser definido a priori é o número de árvores geradas. Portanto para definir esse parâmetro, serão realizados testes empíricos uma vez que aumentar a quantidade de árvores não implica na melhora da acurácia do modelo.

Por fim, serão implementados os modelos utilizando SVM. Assim como as demais técnicas de aprendizado de máquina que serão implementadas nessa dissertação de mestrado, existem parâmetros a serem definidos. Logo, serão realizados testes empíricos para definir os valores de  $\gamma$  e  $C$ .

### **3.5 Métricas de comparação**

Após a implementação de todos os classificadores será realizada uma análise de eficiência dos modelos propostos. Desse modo, cada uma dos modelos gerados serão executados por 10 vezes. Após cada teste serão coletadas métricas para a avaliação de cada modelo. As métricas coletadas serão: Acurácia, curva ROC, matriz de confusão e coeficiente Kappa.

## 4 RESULTADOS E DISCUSSÕES

Neste capítulo serão apresentados os resultados obtidos pelos modelos de classificação de emoções discretas utilizando técnicas de aprendizado de máquina para o conjunto de dados de teste, propostos nessa dissertação de mestrado. Os códigos-fonte utilizados na implementação de cada uma das etapas se encontram no Apêndice A.

### 4.1 Análise de Componentes Principais

Após o processamento e balanceamento dos dados, o próximo passo foi avaliar quais seriam as melhores características a serem utilizadas para o desenvolvimento dos modelos. Assim, foi realizada uma análise de componentes principais pelo algoritmo de Análise de Componentes Principais (PCA).

A Figura 4.1 apresenta os valores de variância explicada obtida pela variação do número de componentes principais utilizados no PCA. Pode-se, constatar que com 18 componentes principais 99% dos dados são explicados, reduzindo a quantidade de características que foram utilizadas no modelo principal.

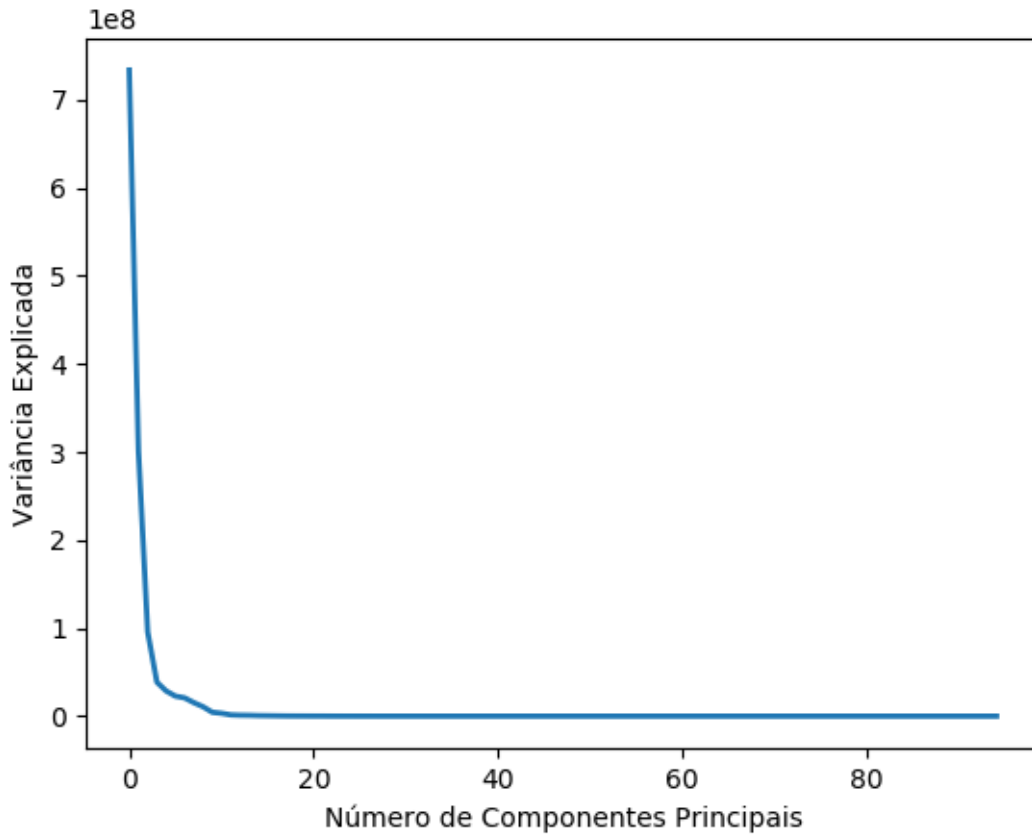
### 4.2 Implementação do modelo utilizando Rede Neural Artificiais

Após definidas as características que seriam utilizadas pelo algoritmo PCA, o próximo passo foi implementar as técnicas de aprendizado de máquina e colher seus resultados para análise de performance e acurácia. O primeiro modelo implementado utilizou uma RNA.

Para a definição dos parâmetros da rede foram realizados testes empíricos para definir os parâmetros de taxa de aprendizado e número de neurônios por camada. A partir da análise dos resultados desses teste, foram definidos os seguintes parâmetros para a rede:

- Taxa de aprendizado: 0.001
- *Momentum*: 0.9
- Número de neurônios em cada camada escondida: 100
- Número de camadas escondidas: 2
- Épocas: 1000

Figura 4.1 – Análise de variância explicada.



Fonte: Arquivo pessoal do autor.

Após a definição dos parâmetros da rede foram executados 10 testes, onde foram coletadas métricas e indicadores de acurácia para os modelos. As Tabelas 4.1 e 4.2 apresentam a média de acurácia obtida pelos modelos durante os testes.

Tabela 4.1 – Resultados para o modelo utilizando RNA + SMOTE.

#	Protegido	Satisfeito	Surpreso	Alegre	Triste	Despreocupado	Medo	Bravo
Teste 1	0.755741127	0.618937644	0.799180328	0.552777778	0.644646925	0.696465696	0.72972973	0.657327586
Teste 2	0.751565762	0.662817552	0.723360656	0.566666667	0.637813212	0.688149688	0.725225225	0.665948276
Teste 3	0.776617954	0.598152425	0.778688525	0.563888889	0.685649203	0.704781705	0.682432432	0.653017241
Teste 4	0.730688935	0.625866051	0.827868852	0.575	0.671981777	0.671517672	0.72972973	0.724137931
Teste 5	0.753653445	0.632794457	0.795081967	0.558333333	0.671981777	0.711018711	0.698198198	0.711206897
Teste 6	0.772442589	0.630484988	0.790983607	0.6	0.656036446	0.669438669	0.722972973	0.635775862
Teste 7	0.803757829	0.669745958	0.805327869	0.563888889	0.633257403	0.715176715	0.698198198	0.689655172
Teste 8	0.784968685	0.623556582	0.75	0.561111111	0.662870159	0.696465696	0.70045045	0.689655172
Teste 9	0.711899791	0.665127021	0.807377049	0.516666667	0.665148064	0.669438669	0.677927928	0.653017241
Teste 10	0.791231733	0.62817552	0.803278689	0.555555556	0.665148064	0.700623701	0.704954955	0.700431034
Média	<b>0.763256785</b>	<b>0.63556582</b>	<b>0.788114754</b>	<b>0.561388889</b>	<b>0.659453303</b>	<b>0.692307692</b>	<b>0.706981982</b>	<b>0.678017241</b>
$\sigma$	$\pm 0.028$	$\pm 0.023$	$\pm 0.030$	$\pm 0.02$	$\pm 0.0165$	$\pm 0.017$	$\pm 0.0190$	$\pm 0.019$

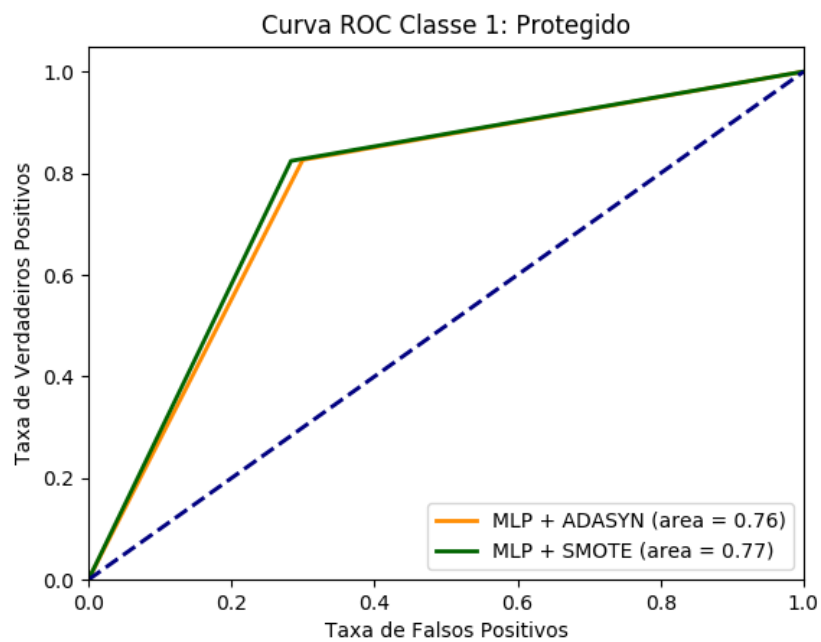
Além da acurácia dos modelos utilizando Redes Neurais Artificiais, para cada um dos modelos de classificação foram geradas as curvas ROC e matrizes de confusão. As Figuras 4.2,

Tabela 4.2 – Resultados para o modelo utilizando RNA + ADASYN.

#	Protegido	Satisfeito	Surpreso	Alegre	Triste	Despreocupado	Medo	Bravo
Teste 1	0.756813417	0.605504587	0.769547325	0.555865922	0.626146789	0.648648649	0.645089286	0.612554113
Teste 2	0.784067086	0.676605505	0.788065844	0.48603352	0.676605505	0.692307692	0.645089286	0.670995671
Teste 3	0.714884696	0.623853211	0.740740741	0.530726257	0.619266055	0.659043659	0.611607143	0.675324675
Teste 4	0.721174004	0.605504587	0.740740741	0.533519553	0.630733945	0.642411642	0.647321429	0.673160173
Teste 5	0.742138365	0.610091743	0.781893004	0.536312849	0.651376147	0.683991684	0.65625	0.634199134
Teste 6	0.735849057	0.674311927	0.773662551	0.547486034	0.610091743	0.694386694	0.660714286	0.658008658
Teste 7	0.737945493	0.630733945	0.761316872	0.541899441	0.605504587	0.667359667	0.651785714	0.597402597
Teste 8	0.756813417	0.649082569	0.726337449	0.561452514	0.587155963	0.642411642	0.65625	0.666666667
Teste 9	0.746331237	0.655963303	0.777777778	0.527932961	0.672018349	0.713097713	0.602678571	0.694805195
Teste 10	0.716981132	0.662844037	0.742798354	0.544692737	0.598623853	0.702702703	0.647321429	0.642857143
Média	<b>0.74129979</b>	<b>0.639449541</b>	<b>0.760288066</b>	<b>0.536592179</b>	<b>0.627752294</b>	<b>0.674636175</b>	<b>0.642410714</b>	<b>0.652597403</b>
$\sigma$	$\pm 0.0212$	$\pm 0.027$	$\pm 0.021$	$\pm 0.020$	$\pm 0.030$	$\pm 0.0260$	$\pm 0.0194$	$\pm 0.0304$

4.3, 4.4, 4.5, 4.6, 4.7, 4.8 e 4.9 apresentam as curvas ROC obtidas pelos modelos de classificação.

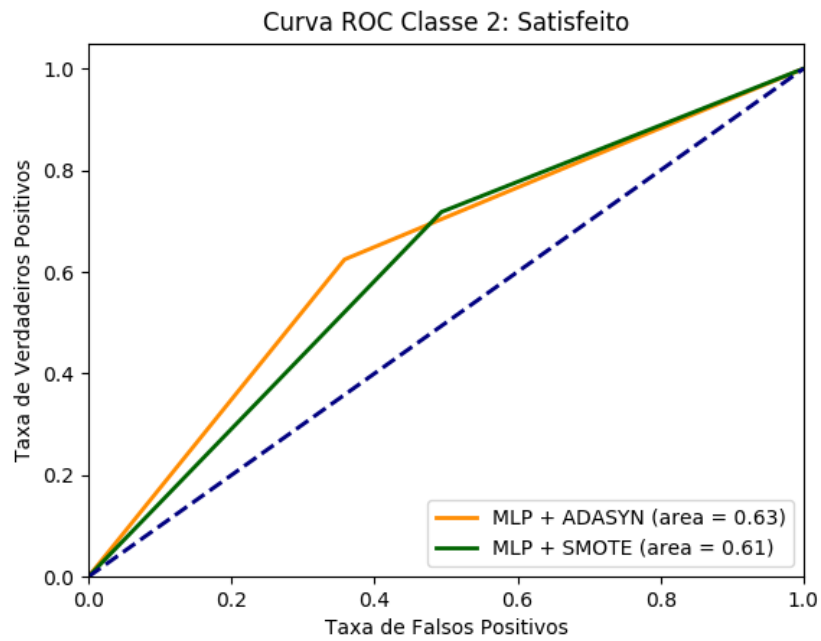
Figura 4.2 – MLP - Curva ROC para a Classe 1: Protegido



Fonte: Arquivo pessoal do autor.

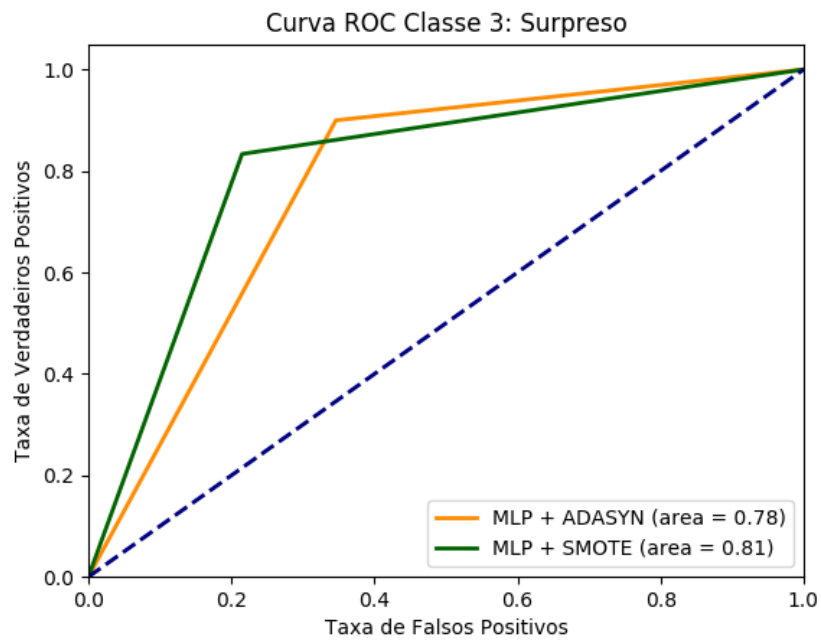


Figura 4.3 – MLP - Curva ROC para a Classe 2: Satisfeito



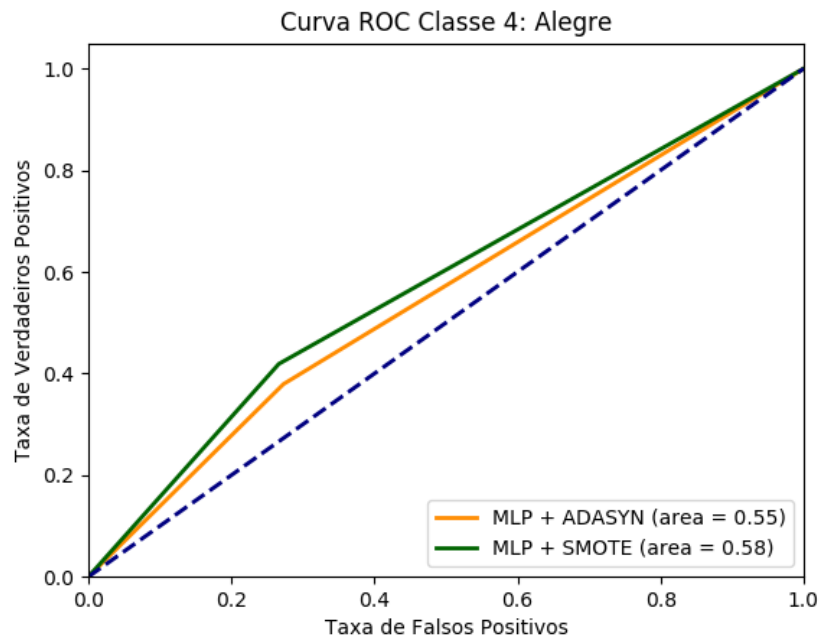
Fonte: Arquivo pessoal do autor.

Figura 4.4 – MLP - Curva ROC para a Classe 3: Surpreso



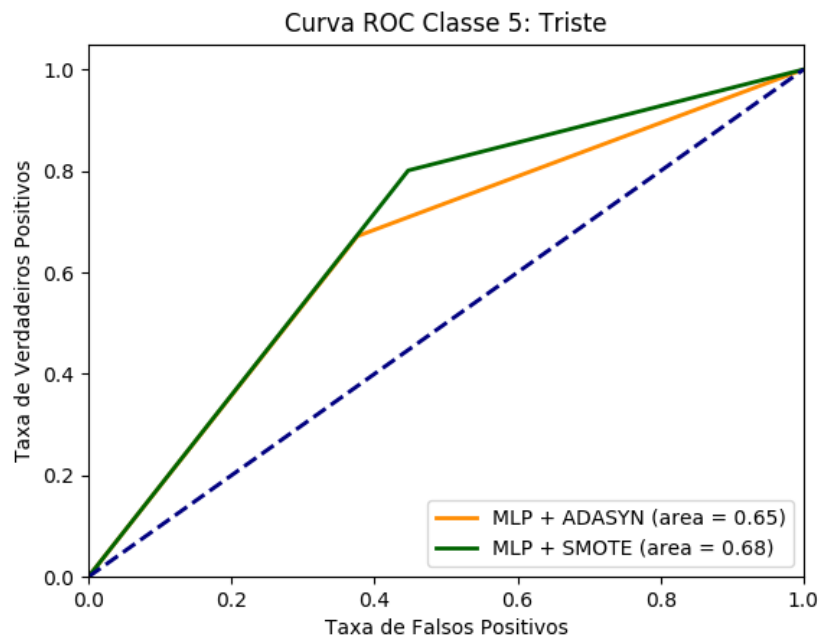
Fonte: Arquivo pessoal do autor.

Figura 4.5 – MLP - Curva ROC para a Classe 4: Alegre



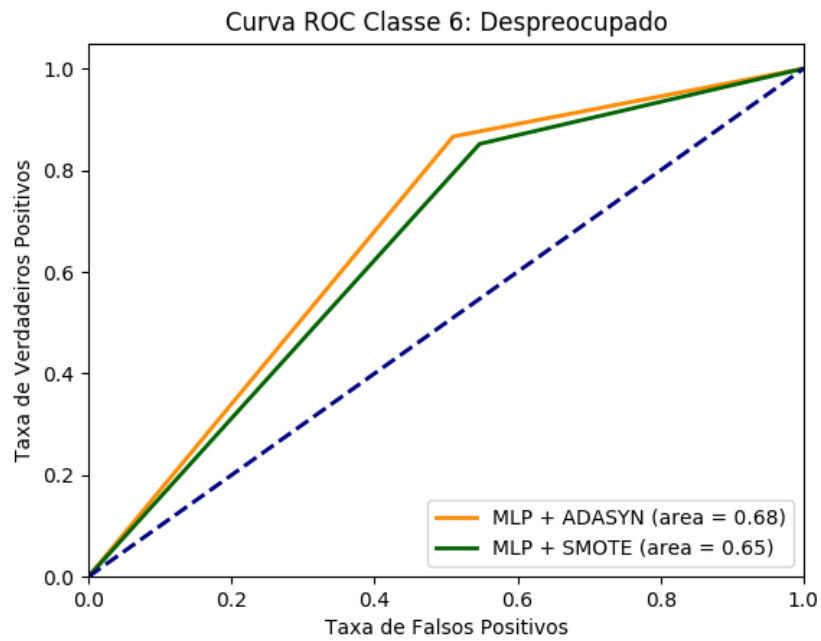
Fonte: Arquivo pessoal do autor.

Figura 4.6 – MLP - Curva ROC para a Classe 5: Triste



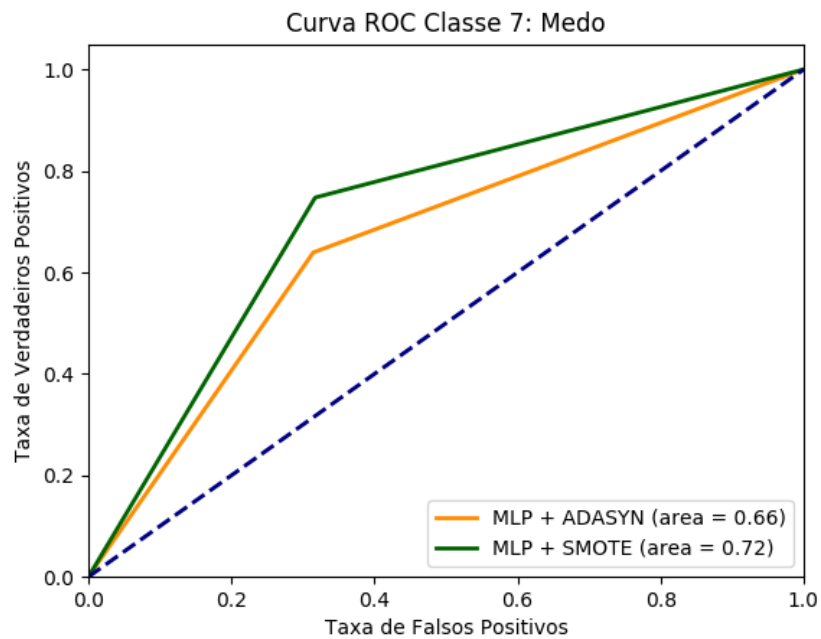
Fonte: Arquivo pessoal do autor.

Figura 4.7 – MLP - Curva ROC para a Classe 6: Despreocupado



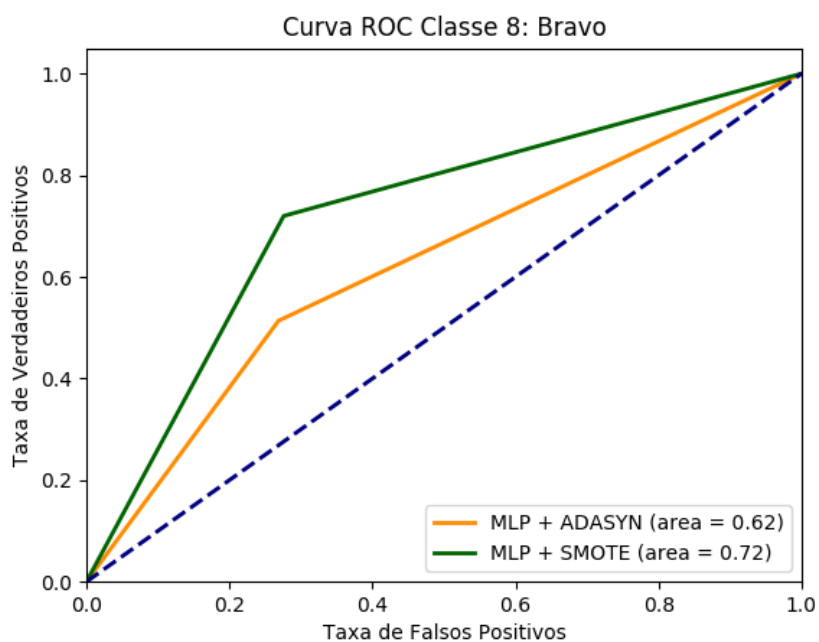
Fonte: Arquivo pessoal do autor.

Figura 4.8 – MLP - Curva ROC para a Classe 7: Medo



Fonte: Arquivo pessoal do autor.

Figura 4.9 – MLP - Curva ROC para a Classe 8: Bravo



Fonte: Arquivo pessoal do autor.

As Tabelas 4.3, 4.5, 4.7, 4.9, 4.11, 4.13, 4.15 e 4.17, apresentam as matrizes de confusão obtidas pelo resultado da classificação dos modelos utilizando o algoritmo ADASYN.

Já as Tabelas 4.4, 4.6, 4.8, 4.10, 4.12, 4.14, 4.16 e 4.18, apresentam as matrizes de confusão obtidas pelo resultado da classificação dos modelos utilizando o algoritmo SMOTE.

Tabela 4.3 – MLP + ADASYN - Matriz de confusão para a Classe 1: Protegido

	Positivo - Real	Negativo - Real	Total
Positivo - Previsto	173 (36.26%)	74 (15.51%)	247
Negativo - Previsto	40 (8.38%)	190 (39.83%)	230
Total	213	264	477

Tabela 4.4 – MLP + SMOTE - Matriz de confusão para a Classe 1: Protegido

	Positivo - Real	Negativo - Real	Total
Positivo - Previsto	172 (35.9%)	68 (14.19%)	240
Negativo - Previsto	42 (8.76%)	197 (41.12%)	239
Total	214	265	479

Tabela 4.5 – MLP + ADASYN - Matriz de confusão para a Classe 2: Satisfeito

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	138 (31.65%)	77 (17.66%)	215
<b>Negativo - Previsto</b>	83 (19.03%)	138 (31.65%)	221
<b>Total</b>	221	215	436

Tabela 4.6 – MLP + SMOTE - Matriz de confusão para a Classe 2: Satisfeito

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	117 (27.02%)	114 (26.32%)	231
<b>Negativo - Previsto</b>	57 (13.16%)	145 (33.48%)	202
<b>Total</b>	174	259	433

Tabela 4.7 – MLP + ADASYN - Matriz de confusão para a Classe 3: Surpreso

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	155 (31.89%)	82 (16.87%)	237
<b>Negativo - Previsto</b>	25 (5.14%)	224 (46.09%)	249
<b>Total</b>	180	306	486

Tabela 4.8 – MLP + SMOTE - Matriz de confusão para a Classe 3: Surpreso

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	190 (38.93%)	52 (10.65%)	242
<b>Negativo - Previsto</b>	41 (8.4%)	205 (42%)	246
<b>Total</b>	231	257	488

Tabela 4.9 – MLP + ADASYN - Matriz de confusão para a Classe 4: Alegre

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	128 (35.75%)	48 (13.4%)	176
<b>Negativo - Previsto</b>	113 (31.56%)	69 (19.27%)	182
<b>Total</b>	241	117	358

Tabela 4.10 – MLP + SMOTE - Matriz de confusão para a Classe 4: Alegre

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	124 (34.44%)	45 (12.5%)	169
<b>Negativo - Previsto</b>	111 (30.83%)	80 (22.22%)	191
<b>Total</b>	235	125	360

Tabela 4.11 – MLP + ADASYN - Matriz de confusão para a Classe 5: Triste

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	141 (32.33%)	85 (19.49%)	226
<b>Negativo - Previsto</b>	69 (15.82%)	141 (32.33%)	210
<b>Total</b>	210	226	436

Tabela 4.12 – MLP + SMOTE - Matriz de confusão para a Classe 5: Triste

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	115 (26.19%)	93 (21.18%)	208
<b>Negativo - Previsto</b>	46 (10.47%)	185 (42.14%)	231
<b>Total</b>	161	278	439

Tabela 4.13 – MLP + ADASYN - Matriz de confusão para a Classe 6: Despreocupado

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	122 (25.36%)	127 (26.4%)	249
<b>Negativo - Previsto</b>	31 (6.44%)	201 (41.78%)	232
<b>Total</b>	153	328	481

Tabela 4.14 – MLP + SMOTE - Matriz de confusão para a Classe 6: Despreocupado

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	111 (23.07%)	134 (27.85%)	245
<b>Negativo - Previsto</b>	35 (7.27%)	201 (41.78%)	236
<b>Total</b>	146	335	481

Tabela 4.15 – MLP + ADASYN - Matriz de confusão para a Classe 7: Medo

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	157 (35.04%)	72 (16.07%)	229
<b>Negativo - Previsto</b>	79 (17.63%)	140 (31.25%)	219
<b>Total</b>	236	212	448

Tabela 4.16 – MLP + SMOTE - Matriz de confusão para a Classe 7: Medo

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	168 (37.83%)	78 (17.56%)	246
<b>Negativo - Previsto</b>	50 (11.26%)	148 (33.33%)	198
<b>Total</b>	218	226	444

Tabela 4.17 – MLP + ADASYN - Matriz de confusão para a Classe 8: Bravo

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	177 (38.31%)	65 (14.06%)	242
<b>Negativo - Previsto</b>	107 (23.16%)	113 (24.45%)	220
<b>Total</b>	284	178	462

Tabela 4.18 – MLP + SMOTE - Matriz de confusão para a Classe 8: Bravo

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	168 (36.2%)	64 (13.79%)	232
<b>Negativo - Previsto</b>	65 (14%)	167 (35.99%)	232
<b>Total</b>	233	231	464

Percebe-se que acurácia obtida pelos modelos utilizando Redes Neurais Artificiais divergem muito para cada emoção. A emoção Surpresa, representada pela Classe 3, foi a que melhor foi classificada obtendo uma taxa de acurácia média de 78% para o modelo utilizando o algoritmo SMOTE e 76% para o modelo utilizando o algoritmo ADASYN. Porém, a divergência pode ser observada ao analisarmos a emoção Alegria (Classe 4). Para esta emoção, foi obtida uma taxa de acurácia de 56% utilizando o algoritmo SMOTE e 53% utilizando o algoritmo ADASYN.

Em média, os modelos utilizando RNA que tiveram os dados balanceados pelo algoritmo SMOTE se sobressaíram quando comparados aos modelos que utilizaram o algoritmo ADASYN para o balanceamento dos dados.

### 4.3 Implementação do modelo utilizando Máquinas de Vetor Suporte

Para o modelo utilizando Máquinas de Vetor Suporte foi utilizado o kernel RBF. Entretanto, existem parâmetros que devem ser ajustados através de testes empíricos. Após esses testes, os seguintes parâmetros foram definidos:

- Kernel: RBF
- $C$ : 1000
- $\gamma$ : 0.000001

Uma vez definidos os parâmetros, foram executados 10 testes para o modelo proposto. As Tabelas 4.19 e 4.20 apresentam os resultados obtidos durante os testes realizados.

Tabela 4.19 – Resultados para o modelo utilizando SVM + SMOTE.

#	Protegido	Satisfeito	Surpreso	Alegre	Triste	Despreocupado	Medo	Bravo
Teste 1	0.868475992	0.799076212	0.891393443	0.722222222	0.806378132	0.860706861	0.828828829	0.855603448
Teste 2	0.88308977	0.743648961	0.879098361	0.730555556	0.829157175	0.848232848	0.817567568	0.844827586
Teste 3	0.899791232	0.780600462	0.895491803	0.711111111	0.826879271	0.85031185	0.817567568	0.851293103
Teste 4	0.88308977	0.792147806	0.901639344	0.730555556	0.835990888	0.862785863	0.826576577	0.842672414
Teste 5	0.891440501	0.780600462	0.889344262	0.702777778	0.781321185	0.85031185	0.815315315	0.834051724
Teste 6	0.88308977	0.778290993	0.875	0.766666667	0.790432802	0.833679834	0.828828829	0.864224138
Teste 7	0.889352818	0.822170901	0.862704918	0.711111111	0.776765376	0.833679834	0.837837838	0.827586207
Teste 8	0.912317328	0.794457275	0.918032787	0.708333333	0.84738041	0.841995842	0.81981982	0.840517241
Teste 9	0.889352818	0.815242494	0.905737705	0.736111111	0.813211845	0.862785863	0.833333333	0.846982759
Teste 10	0.885177453	0.778290993	0.870901639	0.761111111	0.817767654	0.852390852	0.788288288	0.844827586
<b>Média</b>	<b>0.888517745</b>	<b>0.788452656</b>	<b>0.888934426</b>	<b>0.728055556</b>	<b>0.812528474</b>	<b>0.84968815</b>	<b>0.821396396</b>	<b>0.845258621</b>
$\sigma$	$\pm 0.0115$	$\pm 0.0219$	$\pm 0.0171$	$\pm 0.0218$	$\pm 0.0236$	$\pm 0.0107$	$\pm 0.0138$	$\pm 0.0104$

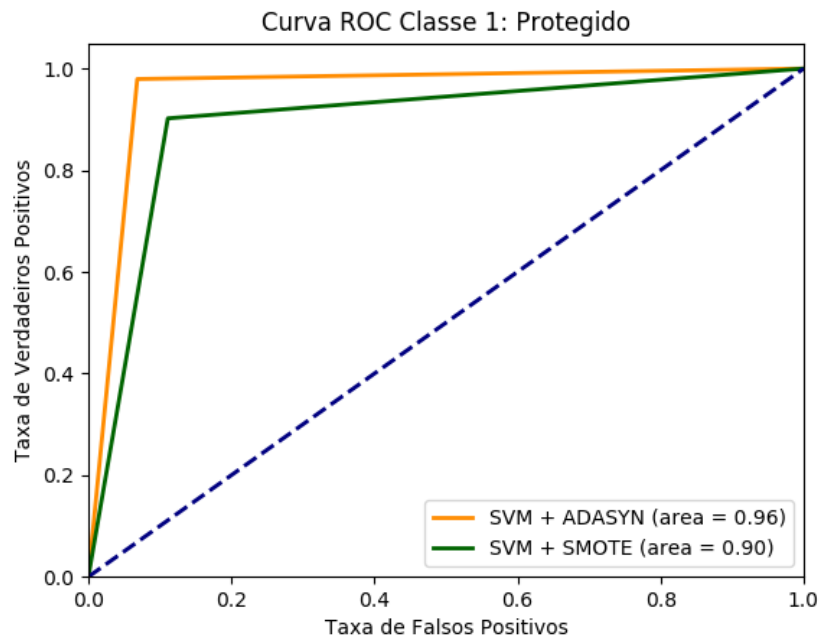
Tabela 4.20 – Resultados para o modelo utilizando SVM + ADASYN.

#	Protegido	Satisfeito	Surpreso	Alegre	Triste	Despreocupado	Medo	Bravo
Teste 1	0.947589099	0.853211009	0.946502058	0.729050279	0.837155963	0.931392931	0.868303571	0.904761905
Teste 2	0.947589099	0.814220183	0.952674897	0.729050279	0.869266055	0.941787942	0.850446429	0.885281385
Teste 3	0.949685535	0.848623853	0.952674897	0.712290503	0.848623853	0.933471933	0.857142857	0.878787879
Teste 4	0.939203354	0.841743119	0.946502058	0.740223464	0.850917431	0.933471933	0.870535714	0.893939394
Teste 5	0.94129979	0.844036697	0.969135802	0.740223464	0.837155963	0.927234927	0.84375	0.872294372
Teste 6	0.953878407	0.839449541	0.940329218	0.712290503	0.832568807	0.93970894	0.879464286	0.902597403
Teste 7	0.918238994	0.830275229	0.932098765	0.729050279	0.850917431	0.937629938	0.879464286	0.874458874
Teste 8	0.937106918	0.839449541	0.946502058	0.681564246	0.871559633	0.916839917	0.881696429	0.913419913
Teste 9	0.945492662	0.827981651	0.927983539	0.720670391	0.864678899	0.945945946	0.859375	0.883116883
Teste 10	0.945492662	0.860091743	0.956790123	0.684357542	0.864678899	0.916839917	0.859375	0.906926407
<b>Média</b>	<b>0.942557652</b>	<b>0.839908257</b>	<b>0.947119342</b>	<b>0.717877095</b>	<b>0.852752294</b>	<b>0.932432432</b>	<b>0.864955357</b>	<b>0.891558442</b>
$\sigma$	$\pm 0.009$	$\pm 0.013$	$\pm 0.011$	$\pm 0.020$	$\pm 0.014$	$\pm 0.009$	$\pm 0.013$	$\pm 0.014$

As Figuras 4.10, 4.11, 4.12, 4.13, 4.14, 4.15, 4.16 e 4.17 apresentam as curvas ROC obtida pela análise dos resultados obtidos pelos modelos utilizando o algoritmo de aprendizado de máquina SVM.

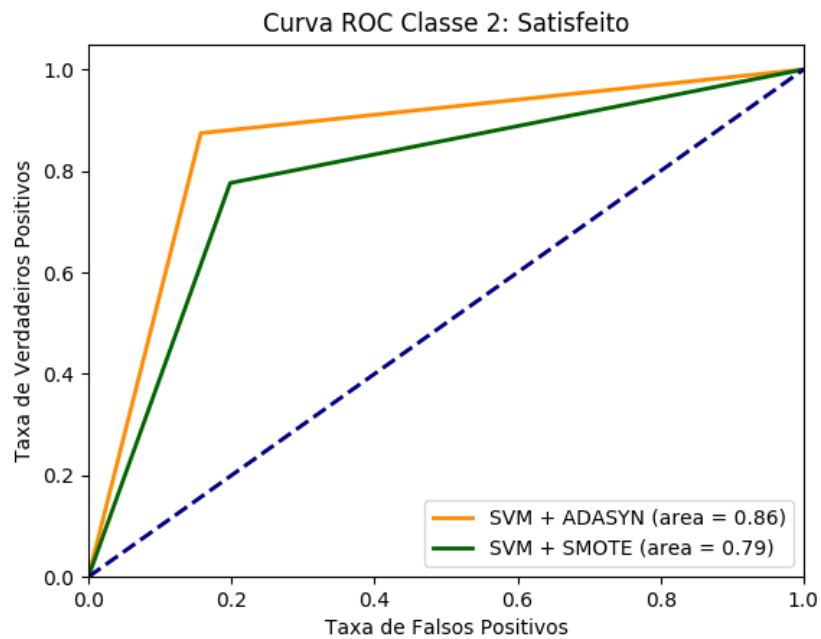


Figura 4.10 – SVM - Curva ROC para a Classe 1: Protegido



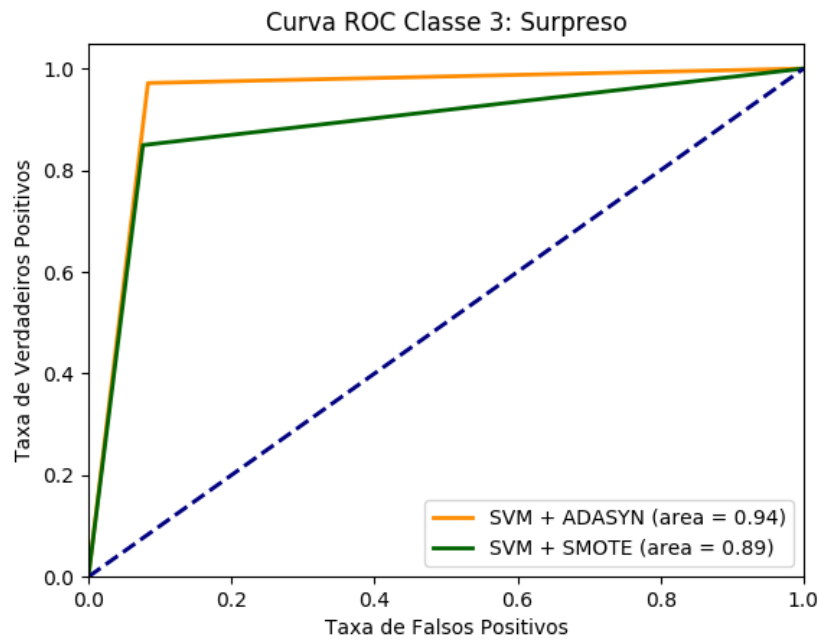
Fonte: Arquivo pessoal do autor.

Figura 4.11 – SVM - Curva ROC para a Classe 2: Satisfeito



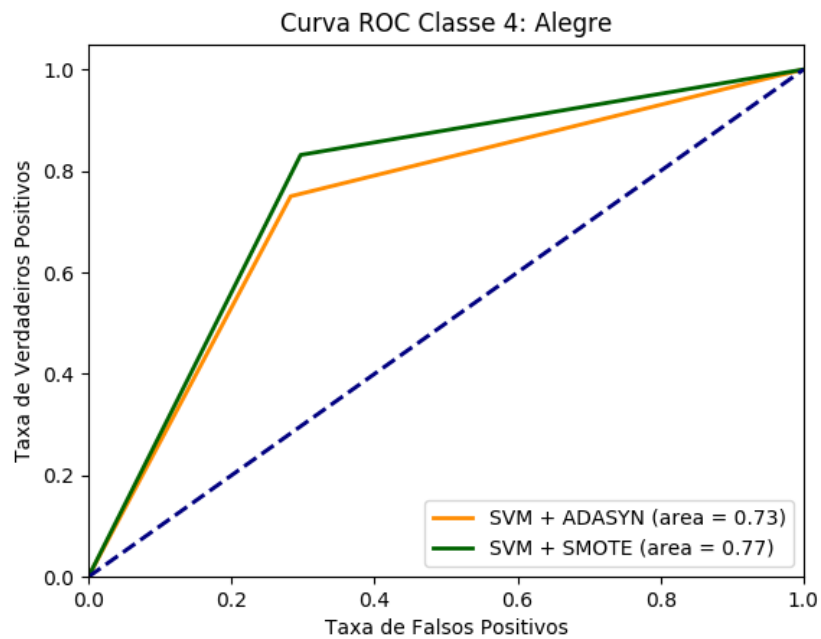
Fonte: Arquivo pessoal do autor.

Figura 4.12 – SVM - Curva ROC para a Classe 3: Surpreso



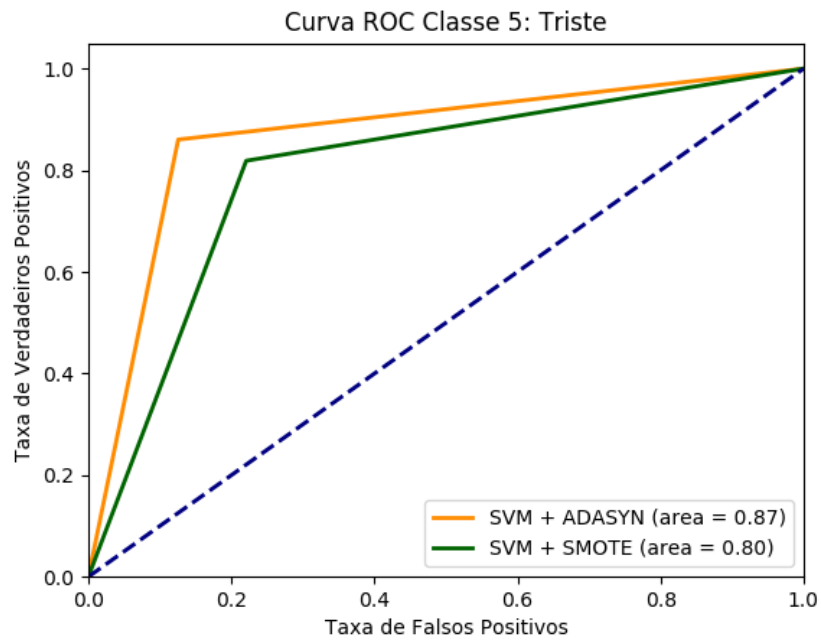
Fonte: Arquivo pessoal do autor.

Figura 4.13 – SVM - Curva ROC para a Classe 4: Alegre



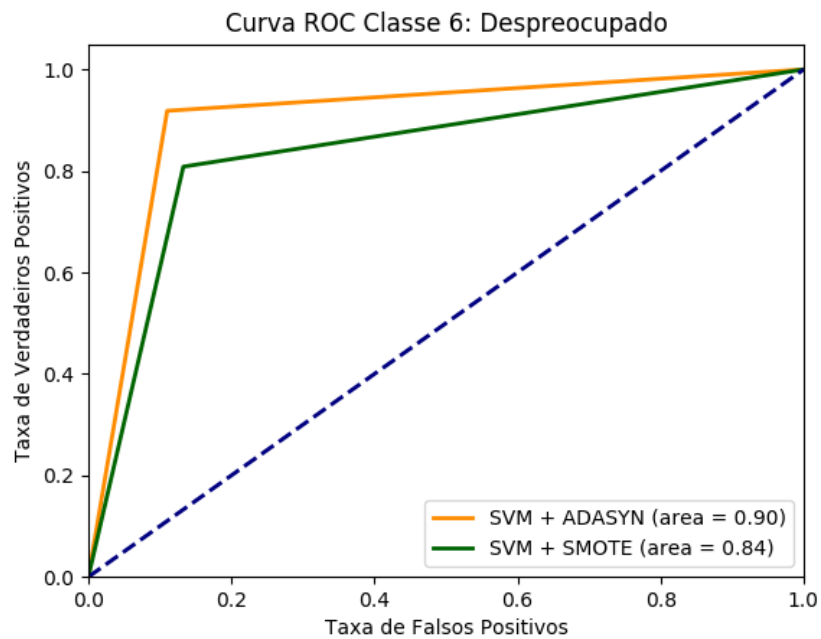
Fonte: Arquivo pessoal do autor.

Figura 4.14 – SVM - Curva ROC para a Classe 5: Triste



Fonte: Arquivo pessoal do autor.

Figura 4.15 – SVM - Curva ROC para a Classe 6: Despreocupado



Fonte: Arquivo pessoal do autor.

Figura 4.16 – SVM - Curva ROC para a Classe 7: Medo

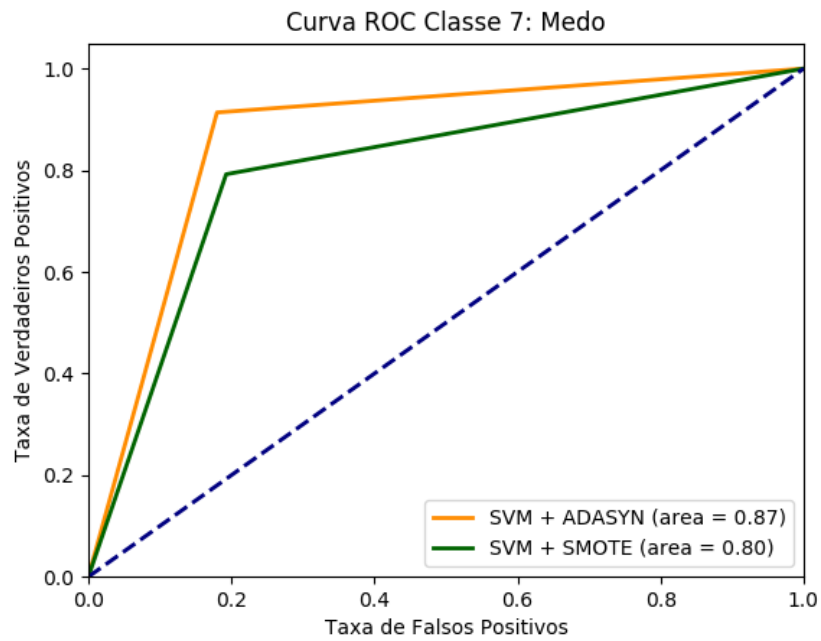
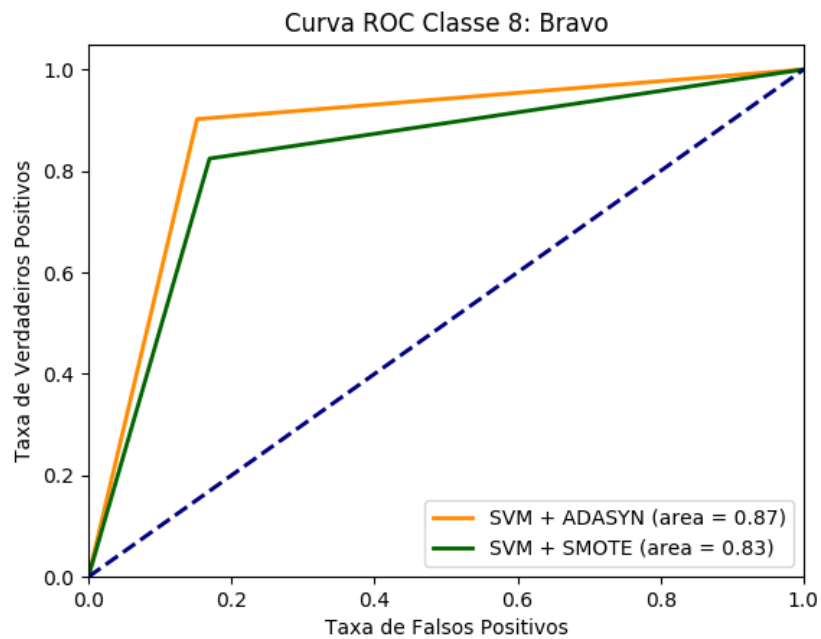


Figura 4.17 – SVM - Curva ROC para a Classe 8: Bravo



As Tabelas 4.21, 4.23, 4.25, 4.27, 4.29, 4.31, 4.33 e 4.35 apresentam as matrizes de confusão para cada modelo utilizando o algoritmo ADASYN para o balanceamento dos dados.

Já as Tabelas 4.22, 4.24, 4.26, 4.28, 4.30, 4.32, 4.34 e 4.36 apresentam as matrizes de confusão para os modelos que utilizaram o algoritmo SMOTE.

Tabela 4.21 – SVM + ADASYN - Matriz de confusão para a Classe 1: Protegido

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	218 (45.7%)	16 (3.35%)	234
<b>Negativo - Previsto</b>	5 (1.04%)	238 (49.89%)	243
<b>Total</b>	223	254	477

Tabela 4.22 – SVM + SMOTE - Matriz de confusão para a Classe 1: Protegido

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	208 (43.42%)	26 (5.42%)	234
<b>Negativo - Previsto</b>	24 (5%)	221 (46.13%)	245
<b>Total</b>	232	247	479

Tabela 4.23 – SVM + ADASYN - Matriz de confusão para a Classe 2: Satisfeito

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	193 (44.26%)	36 (8.25%)	229
<b>Negativo - Previsto</b>	26 (5.86%)	181 (41.51%)	207
<b>Total</b>	219	217	436

Tabela 4.24 – SVM + SMOTE - Matriz de confusão para a Classe 2: Satisfeito

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	186 (42.95%)	46 (10.62%)	232
<b>Negativo - Previsto</b>	45 (10.39%)	156 (36.02%)	201
<b>Total</b>	231	202	433

Tabela 4.25 – SVM + ADASYN - Matriz de confusão para a Classe 3: Surpreso

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	220 (45.26%)	20 (4.11%)	240
<b>Negativo - Previsto</b>	7 (1.44%)	239 (49.17%)	246
<b>Total</b>	227	259	486

#### 4.4 Implementação do modelo utilizando Florestas Aleatórias

Para a implementação do modelo utilizando o algoritmo de Florestas Aleatórias, o único parâmetro definido foi o número de árvores a serem geradas pelo algoritmo. Esse valor foi

Tabela 4.26 – SVM + SMOTE - Matriz de confusão para a Classe 3: Surpreso

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	230 (47.13%)	19 (3.89%)	249
<b>Negativo - Previsto</b>	36 (7.37%)	203 (41.59%)	239
<b>Total</b>	266	222	488

Tabela 4.27 – SVM + ADASYN - Matriz de confusão para a Classe 4: Alegre

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	142 (39.66%)	56 (15.64%)	198
<b>Negativo - Previsto</b>	40 (11.17%)	120 (33.51%)	160
<b>Total</b>	182	176	358

Tabela 4.28 – SVM + SMOTE - Matriz de confusão para a Classe 4: Alegre

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	128 (35.55%)	54 (15%)	182
<b>Negativo - Previsto</b>	30 (8.33%)	148 (41.11%)	178
<b>Total</b>	158	202	360

Tabela 4.29 – SVM + ADASYN - Matriz de confusão para a Classe 5: Triste

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	181 (41.51%)	26 (5.96%)	207
<b>Negativo - Previsto</b>	32 (7.33%)	197 (45.18%)	229
<b>Total</b>	213	223	436

Tabela 4.30 – SVM + SMOTE - Matriz de confusão para a Classe 5: Triste

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	166 (37.81%)	47 (10.7%)	213
<b>Negativo - Previsto</b>	41 (9.33%)	185 (42.36%)	226
<b>Total</b>	207	232	439

Tabela 4.31 – SVM + ADASYN - Matriz de confusão para a Classe 6: Despreocupado

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	210 (43.65%)	26 (5.4%)	236
<b>Negativo - Previsto</b>	20 (4.15%)	225 (46.77%)	245
<b>Total</b>	230	251	481

variado de 10 a 1500, onde foram analisadas a acurácia para cada um dos testes. Após os testes foi definida a utilização de 1000 árvores para o modelo final.

Tabela 4.32 – SVM + SMOTE - Matriz de confusão para a Classe 6: Despreocupado

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	209 (43.45%)	32 (6.65%)	241
<b>Negativo - Previsto</b>	46 (9.56%)	194 (40.33%)	240
<b>Total</b>	255	226	481

Tabela 4.33 – SVM + ADASYN - Matriz de confusão para a Classe 7: Medo

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	187 (41.74%)	41 (9.15%)	228
<b>Negativo - Previsto</b>	19 (4.24%)	201 (44.86%)	220
<b>Total</b>	206	242	448

Tabela 4.34 – SVM + SMOTE - Matriz de confusão para a Classe 7: Medo

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	176 (39.63%)	42 (9.45%)	218
<b>Negativo - Previsto</b>	47 (10.58%)	179 (40,31%)	226
<b>Total</b>	223	221	444

Tabela 4.35 – SVM + ADASYN - Matriz de confusão para a Classe 8: Bravo

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	184 (39.82%)	33 (7.14%)	217
<b>Negativo - Previsto</b>	24 (5.19%)	221 (47.83%)	245
<b>Total</b>	208	254	462

Tabela 4.36 – SVM + SMOTE - Matriz de confusão para a Classe 8: Bravo

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	201 (43.31%)	41 (8.83%)	242
<b>Negativo - Previsto</b>	39 (8.4%)	183 (39.43%)	222
<b>Total</b>	240	224	464

O modelo foi treinado 10 vezes onde métricas de acurácia foram coletadas. As Tabelas 4.37 e 4.38 apresentam os resultados obtidos pelos modelos.

As Figuras 4.18, 4.19, 4.20, 4.21, 4.22, 4.23, 4.24 e 4.25 apresentam as curvas ROC obtidas para cada um dos modelos.

Por fim, as Tabelas 4.39, 4.41, 4.43, 4.45, 4.47, 4.49, 4.51 e 4.53 apresentam as matrizes de confusão geradas para cada um dos modelos utilizando o algoritmo de balanceamento ADASYN.

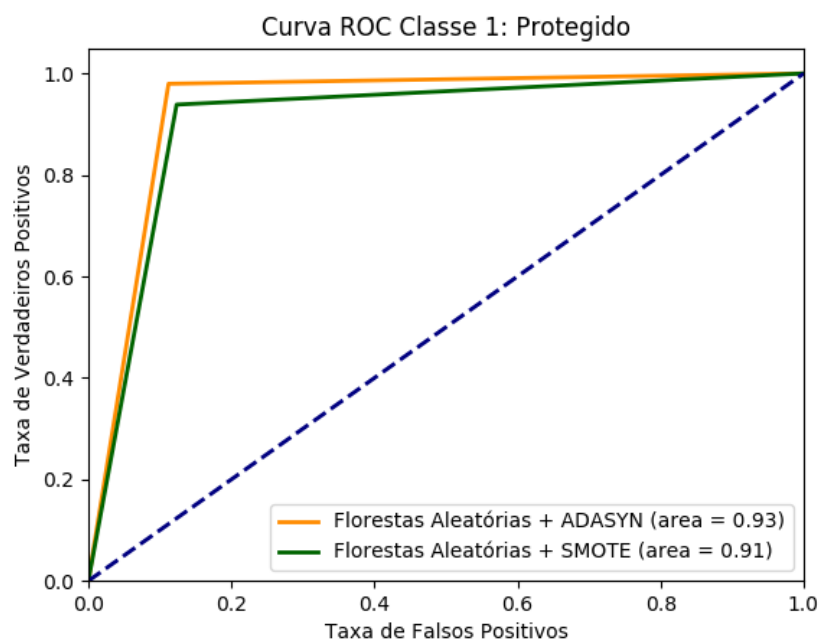
Tabela 4.37 – Resultados para o modelo utilizando Florestas Aleatórias + SMOTE.

#	Protegido	Satisfeito	Surpreso	Alegre	Triste	Despreocupado	Medo	Bravo
Teste 1	0.912317328	0.866050808	0.895491803	0.741666667	0.799544419	0.912681913	0.887387387	0.894396552
Teste 2	0.918580376	0.833718245	0.93647541	0.769444444	0.858769932	0.910602911	0.86036036	0.870689655
Teste 3	0.899791232	0.840646651	0.93852459	0.747222222	0.856492027	0.925155925	0.855855856	0.875
Teste 4	0.916492693	0.810623557	0.920081967	0.775	0.813211845	0.910602911	0.851351351	0.881465517
Teste 5	0.908141962	0.84295612	0.909836066	0.780555556	0.810933941	0.893970894	0.880630631	0.872844828
Teste 6	0.91440501	0.822170901	0.911885246	0.766666667	0.838268793	0.908523909	0.869369369	0.890086207
Teste 7	0.93736952	0.831408776	0.93442623	0.755555556	0.856492027	0.881496881	0.86036036	0.885775862
Teste 8	0.929018789	0.769053118	0.918032787	0.719444444	0.822323462	0.918918919	0.853603604	0.86637931
Teste 9	0.920668058	0.831408776	0.928278689	0.738888889	0.813211845	0.918918919	0.864864865	0.896551724
Teste 10	0.918580376	0.84295612	0.93442623	0.713888889	0.858769932	0.893970894	0.86036036	0.894396552
Média	<b>0.917536534</b>	<b>0.829099307</b>	<b>0.922745902</b>	<b>0.750833333</b>	<b>0.832801822</b>	<b>0.907484407</b>	<b>0.864414414</b>	<b>0.882758621</b>
$\sigma$	$\pm 0.010$	$\pm 0.025$	$\pm 0.014$	$\pm 0.022$	$\pm 0.023$	$\pm 0.013$	$\pm 0.011$	$\pm 0.011$

Tabela 4.38 – Resultados para o modelo utilizando Florestas Aleatórias + ADASYN.

#	Classe 1	Satisfeito	Surpreso	Alegre	Triste	Despreocupado	Medo	Bravo
Teste 1	0.916142558	0.873853211	0.956790123	0.74301676	0.894495413	0.945945946	0.90625	0.913419913
Teste 2	0.947589099	0.853211009	0.948559671	0.751396648	0.873853211	0.93970894	0.881696429	0.928571429
Teste 3	0.947589099	0.864678899	0.942386831	0.726256983	0.880733945	0.95010395	0.90625	0.91991342
Teste 4	0.958071279	0.876146789	0.948559671	0.762569832	0.848623853	0.931392931	0.933035714	0.913419913
Teste 5	0.949685535	0.883027523	0.958847737	0.748603352	0.887614679	0.937629938	0.910714286	0.906926407
Teste 6	0.949685535	0.880733945	0.940329218	0.765363128	0.880733945	0.945945946	0.888392857	0.922077922
Teste 7	0.943396226	0.885321101	0.942386831	0.729050279	0.905963303	0.941787942	0.881696429	0.885281385
Teste 8	0.951781971	0.871559633	0.919753086	0.729050279	0.869266055	0.93970894	0.868303571	0.909090909
Teste 9	0.953878407	0.866972477	0.95473251	0.751396648	0.871559633	0.948024948	0.877232143	0.911255411
Teste 10	0.962264151	0.869266055	0.950617284	0.726256983	0.866972477	0.935550936	0.883928571	0.932900433
Média	<b>0.948008386</b>	<b>0.872477064</b>	<b>0.946296296</b>	<b>0.743296089</b>	<b>0.877</b>	<b>0.941</b>	<b>0.893</b>	<b>0.914</b>
$\sigma$	$\pm 0.012$	$\pm 0.009$	$\pm 0.011$	$\pm 0.014$	$\pm 0.015$	$\pm 0.005$	$\pm 0.019$	$\pm 0.013$

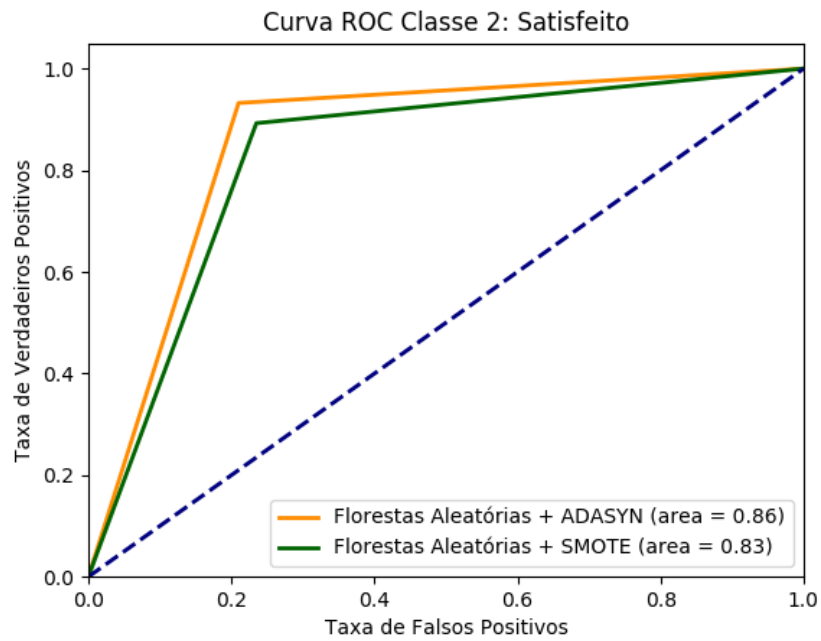
Figura 4.18 – Florestas Aleatórias - Curva ROC para a Classe 1: Protegido



Fonte: Arquivo pessoal do autor.

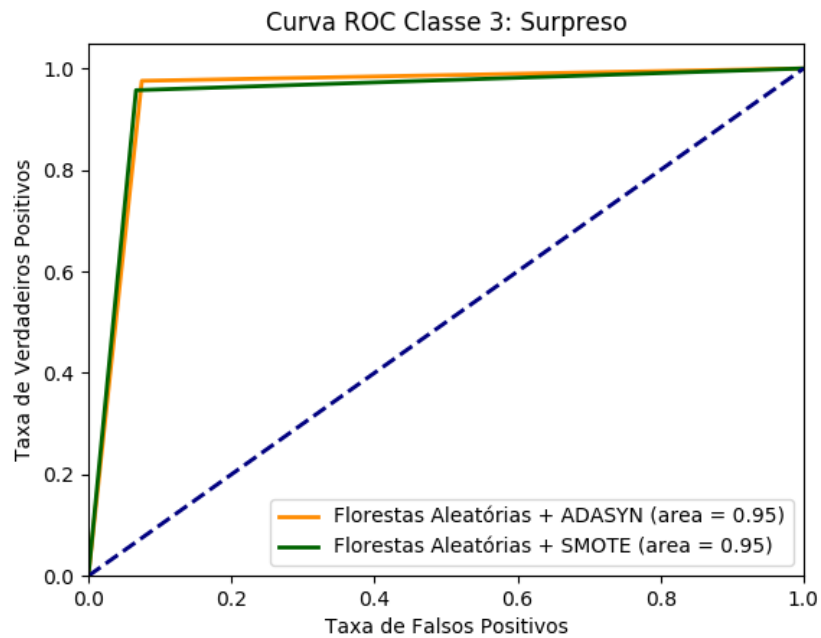


Figura 4.19 – Florestas Aleatórias - Curva ROC para a Classe 2: Satisfeito



Fonte: Arquivo pessoal do autor.

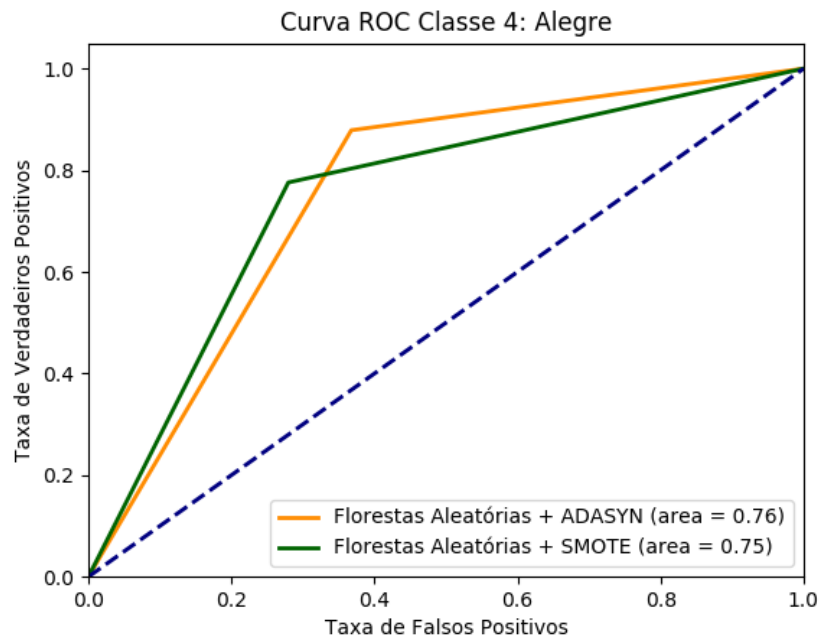
Figura 4.20 – Florestas Aleatórias - Curva ROC para a Classe 3: Surpreso



Fonte: Arquivo pessoal do autor.

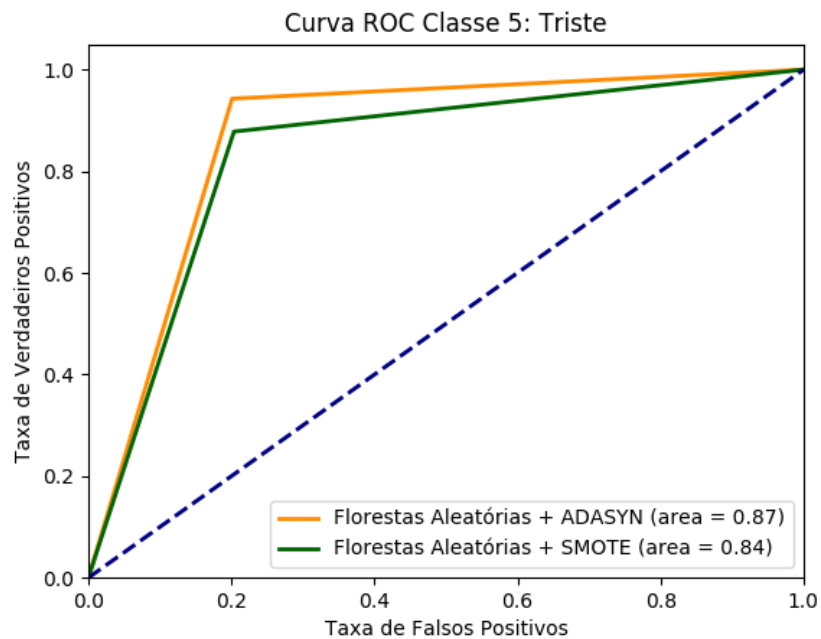
Já as Tabelas 4.40, 4.42, 4.44, 4.46, 4.48, 4.50, 4.52 e 4.54 apresentam as matrizes de confusão geradas para os modelos utilizando o algoritmo SMOTE.

Figura 4.21 – Florestas Aleatórias - Curva ROC para a Classe 4: Alegre



Fonte: Arquivo pessoal do autor.

Figura 4.22 – Florestas Aleatórias - Curva ROC para a Classe 5: Triste

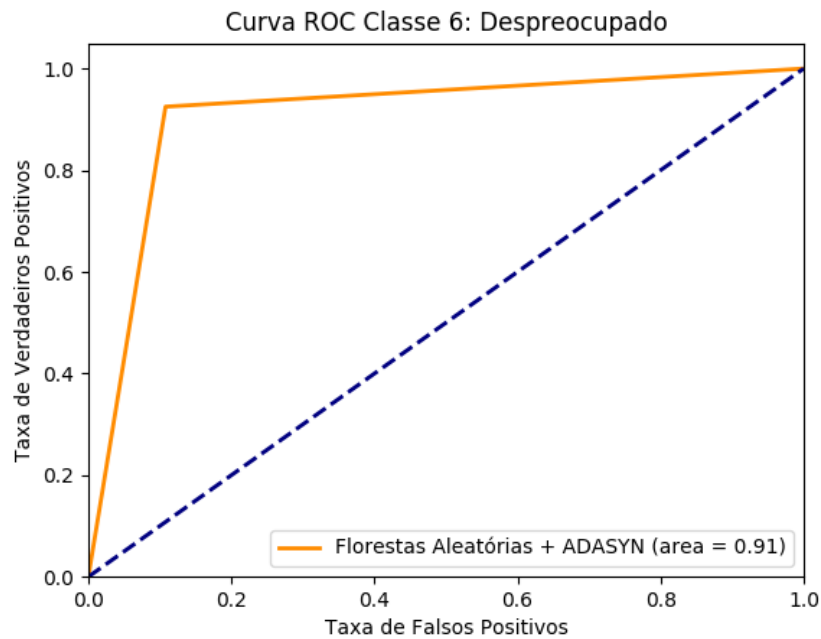


Fonte: Arquivo pessoal do autor.

Tabela 4.39 – Florestas Aleatórias + ADASYN - Matriz de confusão para a Classe 1: Protegido

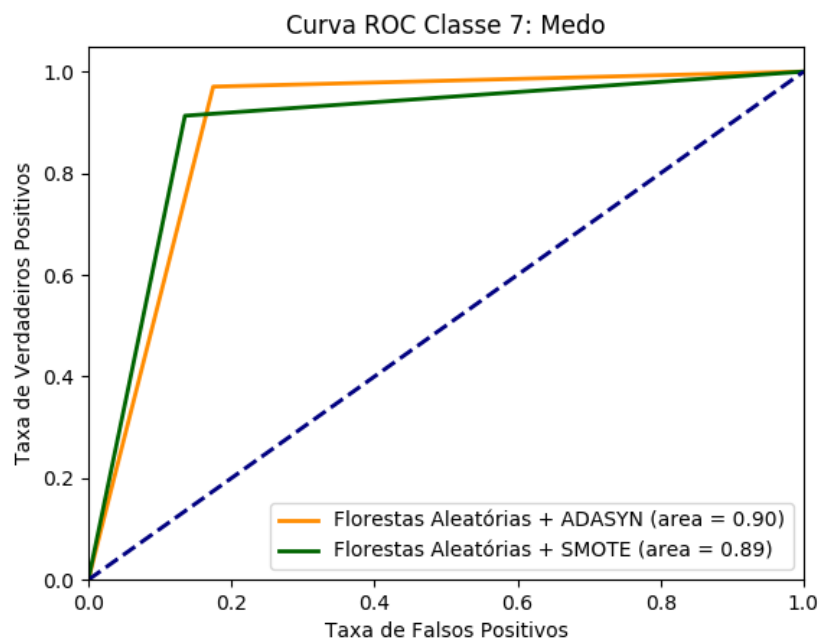
	Positivo - Real	Negativo - Real	Total
Positivo - Previsto	206 (43.18%)	26 (5.45%)	232
Negativo - Previsto	5 (1.04%)	240 (50.31%)	245
Total	211	266	477

Figura 4.23 – Florestas Aleatórias - Curva ROC para a Classe 6: Despreocupado



Fonte: Arquivo pessoal do autor.

Figura 4.24 – Florestas Aleatórias - Curva ROC para a Classe 7: Medo

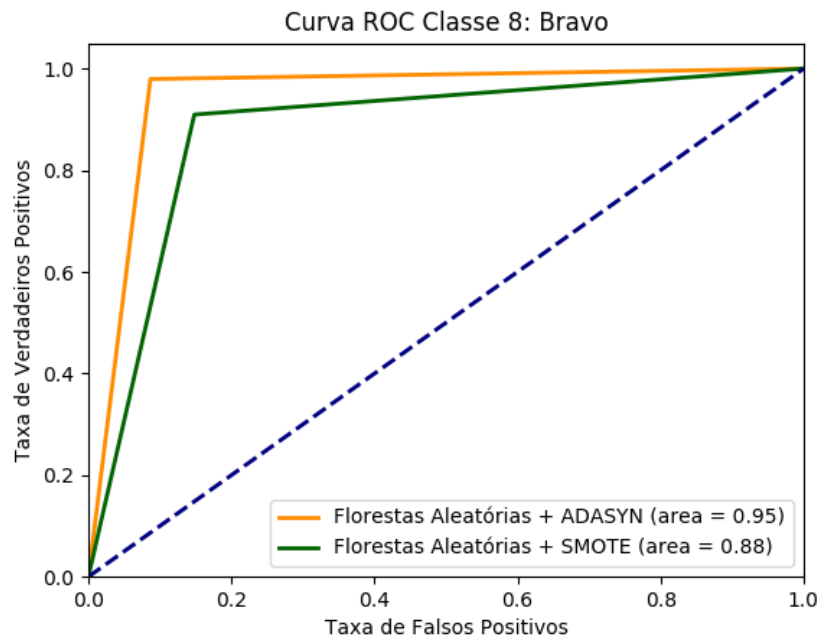


Fonte: Arquivo pessoal do autor.

Tabela 4.40 – Florestas Aleatórias + SMOTE - Matriz de confusão para a Classe 1: Protegido

	Positivo - Real	Negativo - Real	Total
Positivo - Previsto	206 (43%)	29 (6.05%)	235
Negativo - Previsto	15 (3.13%)	229 (47.8%)	244
Total	221	258	479

Figura 4.25 – Florestas Aleatórias - Curva ROC para a Classe 8: Bravo



Fonte: Arquivo pessoal do autor.

Tabela 4.41 – Florestas Aleatórias + ADASYN - Matriz de confusão para a Classe 2: Satisfeito

	Positivo - Real	Negativo - Real	Total
Positivo - Previsto	158 (36.23%)	42 (9.63%)	200
Negativo - Previsto	16 (3.66%)	220 (50.45%)	236
Total	174	262	436

Tabela 4.42 – Florestas Aleatórias + SMOTE - Matriz de confusão para a Classe 2: Satisfeito

	Positivo - Real	Negativo - Real	Total
Positivo - Previsto	189 (43.46%)	58 (13.39%)	247
Negativo - Previsto	20 (4.61%)	166 (38.33%)	186
Total	209	224	433

Tabela 4.43 – Florestas Aleatórias + ADASYN - Matriz de confusão para a Classe 3: Surpreso

	Positivo - Real	Negativo - Real	Total
Positivo - Previsto	223 (45.6%)	18 (3.7%)	241
Negativo - Previsto	6 (1.23%)	239 (49.17%)	245
Total	229	257	486

Tabela 4.44 – Florestas Aleatórias + SMOTE - Matriz de confusão para a Classe 3: Surpreso

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	238 (48.77%)	17 (3.48%)	255
<b>Negativo - Previsto</b>	10 (2.04%)	223 (45.69%)	233
<b>Total</b>	248	240	488

Tabela 4.45 – Florestas Aleatórias + ADASYN - Matriz de confusão para a Classe 4: Alegre

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	117 (32.68%)	68 (18.99%)	185
<b>Negativo - Previsto</b>	21 (5.86%)	152 (42.45%)	173
<b>Total</b>	138	220	358

Tabela 4.46 – Florestas Aleatórias + SMOTE - Matriz de confusão para a Classe 4: Alegre

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	134 (37.22%)	52 (14.44%)	186
<b>Negativo - Previsto</b>	39 (10.83%)	135 (37.5%)	174
<b>Total</b>	173	187	360

Tabela 4.47 – Florestas Aleatórias + ADASYN - Matriz de confusão para a Classe 5: Triste

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	167 (38.3%)	42 (9.63%)	209
<b>Negativo - Previsto</b>	13 (2.98%)	214 (49.08%)	227
<b>Total</b>	180	256	436

Tabela 4.48 – Florestas Aleatórias + SMOTE - Matriz de confusão para a Classe 5: Triste

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	180 (41%)	46 (10.47%)	226
<b>Negativo - Previsto</b>	26 (5.92%)	187 (42.59%)	213
<b>Total</b>	206	233	439

Tabela 4.49 – Florestas Aleatórias + ADASYN - Matriz de confusão para a Classe 6: Despreocupado

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	208 (43.24%)	22 (4.57%)	230
<b>Negativo - Previsto</b>	4 (0.83%)	247 (51.35%)	251
<b>Total</b>	212	269	481

Tabela 4.50 – Florestas Aleatórias + SMOTE - Matriz de confusão para a Classe 6: Despreocupado

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	209 (43.45%)	26 (5.4%)	235
<b>Negativo - Previsto</b>	14 (2.91%)	232 (48.23%)	246
<b>Total</b>	223	258	481

Tabela 4.51 – Florestas Aleatórias + ADASYN - Matriz de confusão para a Classe 7: Medo

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	175 (39.06%)	37 (8.35%)	212
<b>Negativo - Previsto</b>	7 (1.56%)	229 (51.11%)	236
<b>Total</b>	182	266	448

Tabela 4.52 – Florestas Aleatórias + SMOTE - Matriz de confusão para a Classe 7: Medo

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	205 (46.17%)	32 (7.2%)	237
<b>Negativo - Previsto</b>	18 (4.05%)	189 (42.56%)	207
<b>Total</b>	223	221	444

Tabela 4.53 – Florestas Aleatórias + ADASYN - Matriz de confusão para a Classe 8: Bravo

	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	200 (43.29%)	19 (4.11%)	219
<b>Negativo - Previsto</b>	5 (1.08%)	238 (51.51%)	243
<b>Total</b>	205	257	462

Tabela 4.54 – Florestas Aleatórias + SMOTE - Matriz de confusão para a Classe 1: Bravo

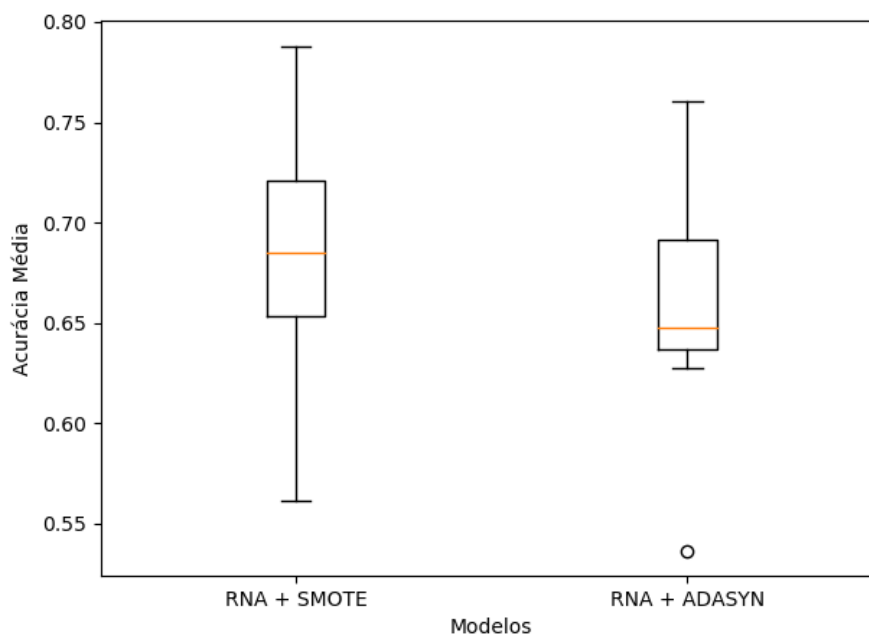
	<b>Positivo - Real</b>	<b>Negativo - Real</b>	<b>Total</b>
<b>Positivo - Previsto</b>	207 (44.61%)	36 (7.75%)	243
<b>Negativo - Previsto</b>	20 (4.31%)	201 (43.31%)	221
<b>Total</b>	227	237	464

#### 4.5 Comparando os resultados

As Figuras 4.26, 4.27 e 4.28 apresentam os resultados obtidos para os modelos propostos. Na Figura 4.26, pode-se perceber que o modelo utilizando o algoritmo SMOTE foi superior

ao modelo utilizando o ADASYN. Já nos modelos utilizando os algoritmos de Florestas Aleatórias e de Máquinas de Vetor Suporte, o inverso ocorreu. Nesses modelos o algoritmo ADASYN foi superior.

Figura 4.26 – Acurácia média para o modelo utilizando RNA.



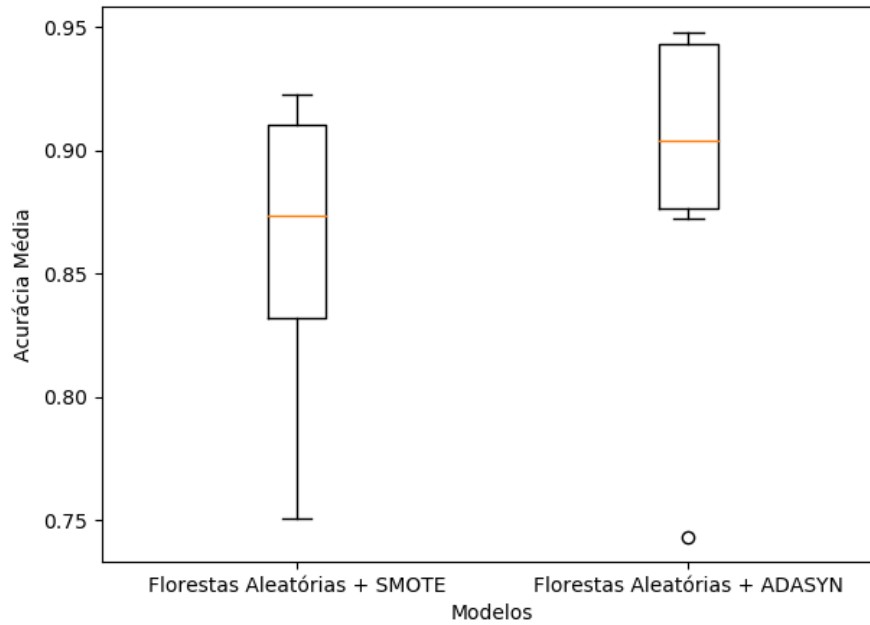
Fonte: Arquivo pessoal do autor.

A Tabela 4.55 apresenta um comparativo com a média de acurácia obtida pelos melhores modelos. Pode-se perceber que o modelo utilizando o algoritmo de aprendizado de máquina Florestas Aleatórias em conjunto com o algoritmo ADASYN foi superior aos demais modelos.

Tabela 4.55 – Tabela comparativa com a acurácia média dos modelos.

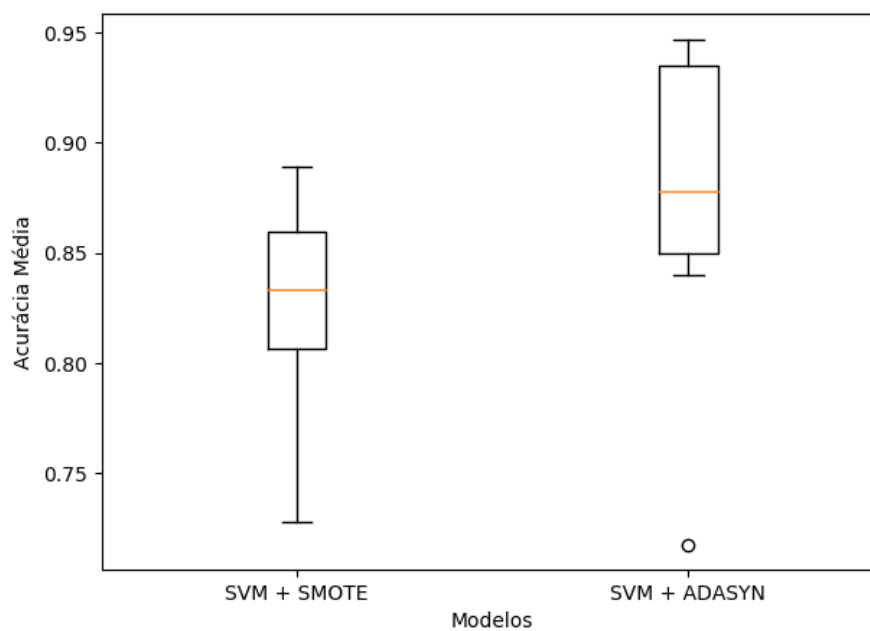
<b>Modelos</b>	<b>Acurácia Média</b>
SVM + ADASYN	87.36%
RF + ADASYN	89.22%
RNA + SMOTE	68.56%

Figura 4.27 – Acurácia média para o modelo utilizando Florestas Aleatórias.



Fonte: Arquivo pessoal do autor.

Figura 4.28 – Acurácia média para o modelo utilizando SVM.



Fonte: Arquivo pessoal do autor.



## 5 CONCLUSÃO

A classificação de emoções é um problema complexo e que durante anos vem sendo estudado pela academia. O avanço nessa área do conhecimento é de extrema importância para o entendimento do comportamento humano, além de fornecer mecanismos de classificação automática de emoções somente pela análise da atividade cerebral de uma pessoa.

Neste trabalho foram desenvolvidos modelos utilizando técnicas de aprendizado de máquina com o intuito de fornecer alternativas e melhorar a classificação de emoções pela análise dos sinais do cérebro obtidos através de um EEG. As técnicas de aprendizado de máquina utilizadas foram: Redes Neurais Artificiais (RNA), Florestas Aleatórias e Máquinas de Vetor Suporte (SVM). Entretanto, foi necessária a utilização de algoritmos de balanceamento de dados para balancear os dados. Assim também foi investigado qual seria o melhor algoritmo para balancear os dados.

Após implementados e realizados testes, foi verificado que os modelos utilizando o algoritmo de aprendizado de máquina Florestas Aleatórias em conjunto com o algoritmo de balanceamento de dados ADASYN apresentaram os melhores resultados, com uma acurácia média de 89,22%. Os modelos utilizando o algoritmo SVM apresentaram uma acurácia média de 87,36%, e também utilizaram o algoritmo ADASYN. Os modelos utilizando o algoritmo de RNA com o algoritmo de balanceamento SMOTE, apresentaram os piores resultados com uma acurácia média de 68,56%.

Desse modo, pode-se inferir que ambos os modelos utilizando tanto Florestas Aleatórias e SVM se mostraram capazes de classificar com uma alta acurácia as emoções discretas. E também, que o algoritmo ADASYN apresentou um melhor desempenho em balancear os dados provenientes de um EEG.

### 5.1 Propostas de Continuidade

Como proposta de continuidade estão:

- A investigação de técnicas de aprendizado de máquina, como modelos de *Deep Learning*;
- A construção de uma nova base de dados formada por uma maior quantidade de participantes;
- Investigar se existe um paralelo entre as emoções e cada região do cérebro.

## REFERÊNCIAS

- ALTMAN, N. S. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. **The American Statistician**, v. 46, n. 3, p. 175–185, 1992. ISSN 0003-1305. Disponível em: <<http://www.tandfonline.com/doi/abs/10.1080/00031305.1992.10475879>>.
- BAGHDADI, A.; ARIBI, Y.; ALIM, A. M. A survey of methods and performances for EEG-based emotion recognition. In: **Advances in Intelligent Systems and Computing**. Springer, Cham, 2017. v. 552, p. 164–174. ISBN 9783319529400. ISSN 21945357. Disponível em: <<http://link.springer.com/10.1007/978-3-319-52941-7>>.
- BANN, E. Y. Discovering Basic Emotion Sets via Semantic Clustering on a Twitter Corpus. n. May, p. 105, 2012. Disponível em: <<http://arxiv.org/abs/1212.6527>>.
- BARRETT, L. F. Are Emotions Natural Kinds? **Perspectives on Psychological Science**, v. 1, n. 1, p. 28–58, 2006. ISSN 17456924.
- BRAZIER, L. Moral conflict. **Psychologist**, v. 16, n. 1, p. 33, 2003. ISSN 09528229. Disponível em: <<http://www.jstor.org/stable/2246769>> <<http://mind.oxfordjournals.org/cgi/doi/10.1093/mind/LI.202>>.
- BREIMAN, L. Random forests. **Machine Learning**, v. 45, n. 1, p. 5–32, 2001. ISSN 08856125.
- CHANG, C.-C.; LIN, C.-J. Libsvm. **ACM Transactions on Intelligent Systems and Technology**, v. 2, n. 3, p. 1–27, 2011. ISSN 21576904. Disponível em: <<http://dl.acm.org/citation.cfm?doid=1961189.1961199>>.
- CHAWLA, N. V. et al. SMOTE: Synthetic minority over-sampling technique. **Journal of Artificial Intelligence Research**, v. 16, p. 321–357, 2002. ISSN 10769757. Disponível em: <<http://www.jair.org/media/953/live-953-2037-jair.pdf>>.
- Chih-Wei Hsu, Chih-Chung Chang; LIN, C.-J. A Practical Guide to Support Vector Classification. **BJU international**, v. 101, n. 1, p. 1396–400, 2008. ISSN 1464-410X. Disponível em: <<http://www.csie.ntu.edu.tw/http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.p>>.
- CHUNG, S. Y.; YOON, H. J. Affective classification using Bayesian classifier and supervised learning. **Control, Automation and Systems (ICCAS)**, p. 1768–1771, 2012.
- COAN, J. A.; ALLEN, J. J.; HARMON-JONES, E. Voluntary facial expression and hemispheric asymmetry over the frontal cortex. **Psychophysiology**, v. 38, n. 6, p. 912–925, 2001. ISSN 0048-5772. Disponível em: <<http://doi.wiley.com/10.1111/1469-8986.3860912>>.
- CORTES, C.; VAPNIK, V. Support-Vector Networks. **Machine Learning**, Kluwer Academic Publishers, v. 20, n. 3, p. 273–297, 1995. ISSN 15730565. Disponível em: <<https://link.springer.com/content/pdf/10.1007%7B%7D2FBF00994018.pdf>> <<http://link.springer.com/10.1007/BF00994>>.
- DREVER, J. The nature of emotion. **The Journal of Abnormal and Social Psychology**, v. 27, n. 3, p. 297–301, 1932. ISSN 0096-851X. Disponível em: <<http://doi.apa.org/getdoi.cfm?doi=10.1037/h0074005>>.

HAYKIN, S. **Neural Networks and Learning Machines**. [S.l.: s.n.], 2008. v. 3. 906 p. ISSN 14337851. ISBN 9780131471399.

HE, H. et al. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: **Proceedings of the International Joint Conference on Neural Networks**. IEEE, 2008. p. 1322–1328. ISBN 9781424418213. ISSN 1098-7576. Disponível em: <<http://ieeexplore.ieee.org/document/4633969/>>.

HE, H.; GARCIA, E. A. Learning from imbalanced data. **IEEE Transactions on Knowledge and Data Engineering**, v. 21, n. 9, p. 1263–1284, sep 2009. ISSN 10414347. Disponível em: <<http://ieeexplore.ieee.org/document/5128907/>>.

HJORTH, B. EEG analysis based on time domain properties. **Electroencephalography and Clinical Neurophysiology**, Elsevier Publishing Company, v. 29, n. 3, p. 306–310, 1970. ISSN 00134694. Disponível em: <[http://ac.els-cdn.com/0013469470901434/1-s2.0-0013469470901434-main.pdf?{\\\_}tid=62d41780-6574-11e7-b150-0000aacb361{\&}acdnat=1499693800{\\\_}9a1b390aed45d302946ab49](http://ac.els-cdn.com/0013469470901434/1-s2.0-0013469470901434-main.pdf?{\_}tid=62d41780-6574-11e7-b150-0000aacb361{\&}acdnat=1499693800{\_}9a1b390aed45d302946ab49)>.

IZARD, C. E. Basic emotions, relations among emotions, and emotion^cognition relations. **Psychological Review**, v. 99, n. 3, p. 561–565, 1992. ISSN 0033-295X. Disponível em: <<http://doi.apa.org/getdoi.cfm?doi=10.1037/0033-295X.99.3.561>>.

IZARD, C. E. Basic Emotions, Natural Kinds, Emotion Schemas, and a New Paradigm. **Perspectives on Psychological Science**, v. 2, n. 3, p. 260–280, 2007. ISSN 17456924.

JENKE, R.; PEER, A.; BUSS, M. Feature extraction and selection for emotion recognition from EEG. **IEEE Transactions on Affective Computing**, v. 5, n. 3, p. 327–339, 2014. ISSN 19493045.

KOELSTRA, S. et al. DEAP: A database for emotion analysis; Using physiological signals. **IEEE Transactions on Affective Computing**, v. 3, n. 1, p. 18–31, 2012. ISSN 19493045.

KOTTI, M.; STYLIANOU, Y. Effective emotion recognition in movie audio tracks. In: **2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. IEEE, 2017. p. 5120–5124. ISBN 978-1-5090-4117-6. Disponível em: <<http://ieeexplore.ieee.org/document/7953132/>>.

KRISNANDHIKA, B. et al. Emotion recognition system based on EEG signals using relative wavelet energy features and a modified radial basis function neural networks. In: **2017 International Conference on Consumer Electronics and Devices (ICCED)**. IEEE, 2017. p. 50–54. ISBN 978-1-5386-0403-8. Disponível em: <<http://ieeexplore.ieee.org/document/8019990/>>.

KULKARNI, V. Y.; SINHA, P. K. Pruning of random forest classifiers: A survey and future directions. **Proceedings - 2012 International Conference on Data Science and Engineering, ICDSE 2012**, p. 64–68, 2012. ISSN 2051-0845.

LEE, Y. Y.; HSIEH, S. Classifying different emotional states by means of eegbased functional connectivity patterns. **PLoS ONE**, v. 9, n. 4, 2014. ISSN 19326203.

LERNER, J. S.; KELTNER, D. Beyond valence: Toward a model of emotion-specific influences on judgement and choice. **Cognition & Emotion**, v. 14, n. 4, p. 473–493, 2000. ISSN 0269-9931. Disponível em: <<http://www.tandfonline.com/doi/abs/10.1080/026999300402763>>.

LEVENSON, R. W. Basic Emotion Questions. **Emotion Review**, v. 3, n. 4, p. 379–386, 2011. ISSN 1754-0739. Disponível em: <<http://journals.sagepub.com/doi/10.1177/1754073911410743>>.

LEVENSON, R. W. et al. Emotion and autonomic nervous system activity in the Minangkabau of West Sumatra. **Journal of Personality and Social Psychology**, v. 62, n. 6, p. 972–988, 1992. ISSN 0022-3514. Disponível em: <<http://doi.apa.org/getdoi.cfm?doi=10.1037/0022-3514.62.6.972>>.

LI, X. et al. Classification of EEG signals using a multiple kernel learning support vector machine. **Sensors (Basel, Switzerland)**, v. 14, n. 7, p. 12784–12802, 2014. ISSN 14248220.

MAHAJAN, K. Classification of EEG using PCA, ICA and Neural Network. **International Journal of Engineering and Advanced Technology (IJEAT)**, v. 1, n. 1, p. 80–83, 2011. Disponível em: <<http://core.kmi.open.ac.uk/download/pdf/2680667.pdf>>.

MCCULLOCH, W. S.; PITTS, W. A Logical Calculus of the Idea Immanent in Nervous Activity. **Bulletin of Mathematical Biophysics**, v. 5, p. 115–133, 1943. ISSN 0007-4985. Disponível em: <<http://www.cse.chalmers.se/{~}coquand/AUTOMATA/mcp.p>>.

MEHRABIAN, A. Framework for a comprehensive description and measurement of emotional states. **Genetic, social, and general psychology monographs**, v. 121, n. 1978, p. 339–361, aug 1995. ISSN 8756-7547. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/7557355>>.

MEHRABIAN, A. Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in Temperament. **Current Psychology**, v. 14, n. 4, p. 261–292, 1996. ISSN 0737-8262. Disponível em: <<https://link.springer.com/content/pdf/10.1007{\%}2FBF02686918.pdfhttp://link.springer.com/10.1007/BF02686>>.

NICOLAS-ALONSO, L. F.; GOMEZ-GIL, J. Brain computer interfaces, a review. **Sensors**, v. 12, n. 2, p. 1211–1279, 2012. ISSN 14248220.

NUGRAHAENI, R. A.; MUTIJARSA, K. Comparative analysis of machine learning KNN, SVM, and random forests algorithm for facial expression classification. **Proceedings - 2016 International Seminar on Application of Technology for Information and Communication, ISEMANTIC 2016**, IEEE, p. 163–168, aug 2017. Disponível em: <<http://ieeexplore.ieee.org/document/7873831/>>.

ORREGO, S. T.; ORREGO, N. T. Classifying Artifacts and Neural EEG Components using SVM. **2016 IEEE Latin American Conference on Computational Intelligence (LA-CCI)**, IEEE, p. 0–4, nov 2016. Disponível em: <<http://ieeexplore.ieee.org/document/7885733/>>.

PEARSON, K. LIII. On lines and planes of closest fit to systems of points in space. **Philosophical Magazine Series 6**, Taylor & Francis Group, v. 2, n. 11, p. 559–572, nov 1901. ISSN 1941-5982. Disponível em: <<http://www.tandfonline.com/doi/abs/10.1080/14786440109462720>>.

PEREIRA, E. T.; GOMES, H. M. The role of data balancing for emotion classification using EEG signals. **Digital Signal Processing (DSP), 2016 IEEE International Conference on**, IEEE, p. 555–559, oct 2016. Disponível em: <<http://ieeexplore.ieee.org/document/7868619/>>.

PLUTCHIK, R. **The nature of emotions: Human emotions have deep evolutionary roots**. 2001. 344–350 p.

POSNER, J.; RUSSELL, J. A.; PETERSON, B. S. The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. **Development and Psychopathology**, NIH Public Access, v. 17, n. 03, p. 715–34, 2005. ISSN 0954-5794. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/16262989><http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2367156>[http://www.journals.cambridge.org/abstract{\\\_}S0954579405050](http://www.journals.cambridge.org/abstract{\_}S0954579405050)>.

RINGNÉR, M. What is principal component analysis? **Nature Biotechnology**, Nature Publishing Group, v. 26, n. 3, p. 303–304, mar 2008. ISSN 1087-0156. Disponível em: <<http://www.nature.com/doi/10.1038/nbt0308-303>>.

ROSENBLATT, F. **The Perceptron: A Perceiving and Recognising Automaton**, 1957. Report 85–460–1, Project PARA p. Disponível em: <<https://blogs.umass.edu/brain-wars/files/2016/03/rosenblatt-1957.pdf>>.

RUSSELL, J. A. A circumplex model of affect. **Journal of Personality and Social Psychology**, v. 39, n. 6, p. 1161–1178, 1980. ISSN 0022-3514. Disponível em: <<https://www2.bc.edu/james-russell/publications/Russell1980.pdf><http://content.apa.org/journals/psp/39/6/1161>>.

SANDER, D. **Basic tastes and basic emotions: Basic problems and perspectives for a nonbasic solution**. 2008. 45–60 p. Disponível em: <[http://www.journals.cambridge.org/abstract{\\\_}S0140525X08003](http://www.journals.cambridge.org/abstract{\_}S0140525X08003)>.

SHOORANGIZ, R.; WEDDELL, S. J.; JONES, R. D. Prediction of microsleeps from EEG: Preliminary results. In: **Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS**. IEEE, 2016. v. 2016-October, p. 4650–4653. ISBN 9781457702204. ISSN 1557170X. Disponível em: <<http://ieeexplore.ieee.org/document/7591764/>>.

SUBASI, A.; GURSOY, M. I. EEG signal classification using PCA, ICA, LDA and support vector machines. **Expert Systems with Applications**, v. 37, n. 12, p. 8659–8666, 2010. ISSN 09574174. Disponível em: <[http://ac.els-cdn.com/S0957417410005695/1-s2.0-S0957417410005695-main.pdf?{\\\_}tid=ef4e2f94-60ff-11e7-93bf-00000aacb35f{&}acdnat=1499203980{\\\_}bf0e965fc131f6dae86ab89](http://ac.els-cdn.com/S0957417410005695/1-s2.0-S0957417410005695-main.pdf?{\_}tid=ef4e2f94-60ff-11e7-93bf-00000aacb35f{&}acdnat=1499203980{\_}bf0e965fc131f6dae86ab89)>.

TAKAGI, S. et al. Multisensory perception of the six basic emotions is modulated by attentional instruction and unattended modality. **Frontiers in Integrative Neuroscience**, v. 9, n. February, p. 1–10, 2015. ISSN 1662-5145. Disponível em: <<http://journal.frontiersin.org/Article/10.3389/fnint.2015.00001/abstract>>.

TALLGREN, P. et al. Evaluation of commercially available electrodes and gels for recording of slow EEG potentials. **Clinical Neurophysiology**, v. 116, n. 4, p. 799–806, apr 2005. ISSN 13882457. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/15792889><http://linkinghub.elsevier.com/retrieve/pii/S1388245704003906>>.

TIBDEWAL, M. N.; TALE, S. A. Multichannel detection of epilepsy using SVM classifier on EEG signal. **2016 International Conference on Computing Communication Control and automation (ICCUBEA)**, IEEE, p. 1–6, aug 2016. Disponível em: <<http://ieeexplore.ieee.org/document/7860106/>>.

USAKLI, A. B. Improvement of EEG signal acquisition: An electrical aspect for state of the Art of front end. **Computational Intelligence and Neuroscience**, Hindawi, v. 2010, p. 630649, feb 2010. ISSN 16875265. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/20148074http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2817545>>.

VELARDE-REYES, E.; MÁRQUEZ-BOCALANDRO, Y.; MARTÍNEZ-MONTES, E. **EEG Wireless Recording and Multidimensional Analysis**. [S.l.]: American Scientific Publishers, 2012. v. 3. 1–16 p.

WIDROW, B.; HOFF, M. **Adaptive switching circuits**. [s.n.], 1960. 96 – 104 p. ISBN 0-262-01097-6. Disponível em: <<https://books.google.com.br/books?id=Vs4EAAAIAAJ>>.

XIN, L. et al. Relevance Vector Machine Based EEG Emotion Recognition. In: **2016 Sixth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC)**. IEEE, 2016. p. 293–297. ISBN 978-1-5090-1195-7. Disponível em: <<http://ieeexplore.ieee.org/document/7774786/>>.

YANG, N. et al. Speech-based emotion classification using multiclass SVM with hybrid kernel and thresholding fusion. In: **2012 IEEE Workshop on Spoken Language Technology, SLT 2012 - Proceedings**. IEEE, 2012. p. 455–460. ISBN 9781467351263. Disponível em: <<http://ieeexplore.ieee.org/document/6424267/>>.

ZHANG, F.; MENG, H.; LI, M. Emotion extraction and recognition from music. In: **2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, ICNC-FSKD 2016**. IEEE, 2016. p. 1728–1733. ISBN 9781509040933. Disponível em: <<http://ieeexplore.ieee.org/document/7603438/>>.

ZHANG, S.; GAO, J.; CHEN, Z. Analysis of emotion EEG classification based on GA-Fisher classifier. In: **Proceedings - 2011 1st International Workshop on Complexity and Data Mining, IWCDM 2011**. IEEE, 2011. p. 24–27. ISBN 9780769545851. Disponível em: <<http://ieeexplore.ieee.org/document/6128409/>>.

ZHANG, W. et al. Emotion Recognition in Speech Using Multi-classification SVM. In: **2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)**. IEEE, 2015. p. 1181–1186. ISBN 978-1-4673-7211-4. Disponível em: <<http://ieeexplore.ieee.org/document/7518394/>>.

ZHANG, Y.; CHASE, S. M. A control-theoretic approach to brain-computer interface design. **Proceedings of the American Control Conference**, v. 2016-July, p. 5765–5771, 2016. ISSN 07431619.

## APÊNDICE A – Código Fonte

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

"""
@author: vancleys

Source Code for emotion classification in EEG signals.

"""

import numpy as np
import matplotlib.pyplot as plt
from sklearn.neural_network import MLPClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.metrics import classification_report
from sklearn.metrics import roc_curve
from sklearn.metrics import auc
from sklearn.metrics import confusion_matrix
from sklearn.metrics import cohen_kappa_score
from sklearn.model_selection import train_test_split
from imblearn.over_sampling import SMOTE
from imblearn.over_sampling import ADASYN
from sklearn.decomposition import PCA
# from sklearn.manifold import TSNE
from keras.wrappers.scikit_learn import KerasClassifier
from keras.layers import Dense
from keras.layers import Dropout
from keras.models import Sequential
import datetime

def MLP(x, y):
```



```

"""
Return the Multilayer Perceptron Neural Network model.

Keyword arguments:
x -- training data
y -- training label

Model parameters:

epochs -- max number of epochs to train
hidden_layer_sizes -- model architecture
activation -- activation function
solver -- solver for weight otimization
learning_rate -- learning rate for weight otimization
momentum -- momentum value for the SGD solver
"""

ts = datetime.datetime.now()
mlp = MLPClassifier(hidden_layer_sizes=(100, 100), activation="logistic",
                    solver='adam', max_iter=1000)

mlp.fit(x, y)
tf = datetime.datetime.now()
tempo = tf - ts

# print("Training time for the MLP model: " + str(tempo))

return mlp

def dlModel():

    model = Sequential()
    model.add(Dense(50, input_dim=18, activation="relu"))

```



```

model.add(Dropout(0.2))
model.add(Dense(50, activation="relu"))
model.add(Dropout(0.2))
model.add(Dense(50, activation="relu"))
model.add(Dropout(0.2))
model.add(Dense(50, activation="relu"))
model.add(Dropout(0.2))
model.add(Dense(1, activation="sigmoid"))
model.compile(loss='binary_crossentropy', optimizer='rmsprop',
              metrics=['accuracy'])

return model

```

```
def DeepLearning(x, y):
```

```

    ts = datetime.datetime.now()
    dl = KerasClassifier(build_fn=dlModel, epochs=500, batch_size=5, verbose=1)
    dl.fit(x, y)
    tf = datetime.datetime.now()
    tempo = tf - ts

```

```
    print("Training time fo the Deep Learning model: " + str(tempo))
```

```
    return dl
```

```
def SVM(x, y):
```

```

    ts = datetime.datetime.now()
    svm = SVC(kernel='rbf', gamma=0.000001, C=1000)
    svm.fit(x, y)
    tf = datetime.datetime.now()
    tempo = tf - ts

```

```
# print("Training time for the SVM model: " + str(tempo))

return svm

def RF(x, y):

    ts = datetime.datetime.now()
    rf = RandomForestClassifier(n_estimators=1000, n_jobs=-1)
    rf.fit(x, y)
    tf = datetime.datetime.now()
    tempo = tf - ts

    # print("Training time for the Random Forest model: " + str(tempo))

    return rf

def generate_roc_plot(y_true, y_pred):
    """
    Plot the ROC Curve.

    Keyword arguments:
    y_true -- True labels for the test data
    y_pred -- Predicted labels for the test data

    """

    fpr, tpr, _ = roc_curve(y_true, y_pred)
    fpr2, tpr2, _ = roc_curve(y_true, y_pred)
    roc_auc = auc(fpr, tpr)
    plt.figure()
    plt.plot(fpr, tpr, color='darkorange', lw=2,
```

```

        label="Curva ROC (area = %.4f)" % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('Taxa de falsos positivos')
plt.ylabel('Taxa de verdadeiros positivos')
plt.legend(loc='lower right')
plt.title('Curva ROC')
# plt.show()
plt.savefig("./Resultados/roc-smote-tsne.png")

```

```

def testModel(model, x, y):

    y_pred = model.predict(x)

    score = model.score(x, y)

    # print(classification_report(y, y_pred))

    roc_curve(y, y_pred)

    # print("\n Confusion Matrix:")
    # print(confusion_matrix(y, y_pred))

    # print("\n Kappa:")
    # print(cohen_kappa_score(y, y_pred))

    # print("\n Score:")
    print(score)

    # generate_roc_plot(y, y_pred)

```

```

def main():

    x = np.loadtxt('./Data/matlab.csv', delimiter=',')
    y = np.loadtxt('./Data/ymatlab.csv', delimiter=',')

    labels = []

    for item in y:
        if (item != 8):
            labels.append(0)
        else:
            labels.append(1)
    y = labels

    smote = SMOTE()
    x, y = smote.fit_sample(x, y)

    # adasyn = ADASYN()
    # x, y = adasyn.fit_sample(x, y)

    # pca = PCA(n_components=10)
    pca = PCA(n_components=18)
    x = pca.fit_transform(x, y)
    # print(pca.explained_variance_ratio_)

    # tsne = TSNE(n_components=3)
    # tsne.fit_transform(x, y)

    for i in range(10):

        xtrain, xtteste, ytrain, ytteste = train_test_split(x, y, test_size=0.2)

        # mlp = MLP(xtrain, ytrain)
        # deepLearning = DeepLearning(xtrain, ytrain)

```

```
svm = SVM(xtrain, ytrain)
# rf = RF(xtrain, ytrain)

# print("\n")

# print("[*]      Test results for the MLP model \n")
# testModel(mlp, xtteste, ytteste)
# print("[*]      Test results for the Deep Learning model \n")
# testModel(deepLearning, xtteste, ytteste)
# print("[*]      Test results for the SVM model \n")
testModel(svm, xtteste, ytteste)
# print("[*]      Test results for the Random Forest model \n")
# testModel(rf, xtteste, ytteste)

if __name__ == '__main__':
    main()
```