

Moisés Rocha dos Santos

Modelos de Predição Aplicados ao
Aprendizado Motor

São Luís

2019

Moisés Rocha dos Santos

Modelos de Predição Aplicados ao Aprendizado Motor

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Mestre em Ciência da Computação.

Universidade Federal do Maranhão

Centro de Ciências Exatas e Tecnologia

Programa de Pós-Graduação em Ciência da Computação

Orientador: Prof. Dr. Areolino de Almeida Neto

São Luís

2019

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Núcleo Integrado de Bibliotecas/UFMA

Santos, Moisés Rocha.

Modelos de Predição Aplicados ao Aprendizado Motor /
Moisés Rocha Santos. - 2019.

65 p.

Orientador(a): Areolino de Almeida Neto.

Dissertação (Mestrado) - Programa de Pós-graduação em
Ciência da Computação/ccet, Universidade Federal do
Maranhão, São Luís, 2019.

1. Aquisição de Habilidade Motora. 2. Modelos de
Classificação. 3. Tarefa de Traçado. I. de Almeida Neto,
Areolino. II. Título.

Moisés Rocha dos Santos

Modelos de Predição Aplicados ao Aprendizado Motor

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Mestre em Ciência da Computação.

Aprovado em 11 de fevereiro de 2019

Prof. Dr. Areolino de Almeida Neto (Orientador)
Universidade Federal do Maranhão - UFMA

Prof. Dr. Paulo Rogério de Almeida Ribeiro
Universidade Federal do Maranhão - UFMA

Prof. Dr. Alex Oliveira Barradas Filho
Universidade Federal do Maranhão - UFMA

Prof. Dr. Fabrício Lima Brasil
Instituto Internacional de Neurociências
Edmond e Lily Safra - IIN-ELS

São Luís
2019

*Este trabalho é dedicado aos meus tios
Carlos Magno e Antônio dos Santos
(in memoriam)*

Agradecimentos

Em primeiro lugar, gostaria de agradecer aos meus pais, Dona Maria das Mercês e Senhor Ivaldo Santos, que me fizeram realizar o sonho deles de que a Educação é o bem mais precioso que devemos buscar em nossas vidas. Obrigado, esta paixão pelo conhecimento hoje é o que me move e faz-me ampliar meus horizontes. Assim como, agradeço aos irmãos Davi, Lenivalda, Ivaldo Neto e Sandra pelo apoio direto e indireto.

À minha esposa Fiana Cutrim, por sempre estar do meu lado nos momentos de absurda pressão.

Um agradecimento especial à Professora Ellen Caroline pela amizade e todos os conselhos acadêmicos que me deu.

Aos orientadores deste trabalho, Prof. Areolino de Almeida Neto e Prof. Paulo Rogério de A. Ribeiro, que me auxiliaram neste estudo desafiador na área de neuroreabilitação. A experiência de vocês e empenho constante foram decisivos para a conclusão deste trabalho. Agradeço também ao Marco Curado pela ideia que serviu de base para este estudo.

Aos professores do PPGCC-UFMA, pela formação proporcionada através das disciplinas. Estas serviram de norte para este trabalho.

Agradeço ao Mateus Barros por ter implementado em sua monografia a plataforma que fez possível a coleta de dados deste trabalho.

Agradeço também à amizade dos colegas da ECP, LACMOR e INOVTEC. Em muitas situações, vocês foram meu ponto de apoio. Especialmente, agradeço aos amigos Camila Santos, seu marido Daniel, Chrystian, Matheus Menezes, Ítalo, Luann, Antonio Mourão, Eduardo Dorneles, Gilvan, André Baluz, Joaquim, Rodrigo Garcês, Ramon, Dayson. Aos amigos mestres Alex Newman, Marcelo Nascimento, Raphael, Thiago Lemos, Moisés Lawrence, André Felipe pela amizade e referência.

Um agradecimento especial à FAPEMA pelo financiamento deste trabalho em forma de bolsa de estudos.

*“Prefiro ter questões que não podem ser respondidas,
que respostas que não podem ser questionadas.”
(Richard Feynman)*

Resumo

O presente trabalho tem como objetivo propor uma abordagem para estimar a quantidade de sessões necessárias para aprender uma tarefa motora. As atividades motoras são a principal forma de interagir com o mundo que nos rodeia. Portanto, a perda da capacidade de realizar algumas dessas atividades, como resultado de uma doença neurológica, é um dano grave ao indivíduo. Na literatura, há muitos trabalhos sobre aprendizado motor, em sua maioria buscando formas de diminuir o tempo de aquisição de habilidade ou reabilitação motora. Entretanto, poucos trabalhos concentram-se em tentar estimar o tempo de treinamento necessário para adquirir determinado desempenho motor. Desta forma, a metodologia empregada nesta pesquisa consistiu na revisão de literatura de aquisição de habilidade motora, bem como na montagem da configuração inicial de uma plataforma de treinamento, aplicação de um experimento piloto com três participantes e um experimento final com oito participantes. No experimento piloto, uma sessão de treinamento de três blocos para cada participante foi realizada e objetivou-se prever em qual bloco o participante encontrava-se. A partir de três participantes reais, 18 participantes simulados foram gerados, visando a aferir o desempenho do experimento com mais participantes, sendo que se estimou o bloco através do desempenho médio dos participantes. No experimento final, foram realizadas três sessões para cada participante, cujo objetivo era prever em qual sessão o participante alcançaria determinado erro com base no seu perfil e no seu desempenho inicial. Os modelos de classificação utilizados no experimento final foram: Algoritmo K-Vizinhos mais Próximos, Rede Neural MLP, Árvore de Decisão, Máquina de Suporte Vetorial e Aprendizagem de Máquina Automática (AutoML) com "Auto Weka". Nos resultados do experimento piloto, percebeu-se um aperfeiçoamento motor dos participantes após o treino. Através dos dados do experimento piloto, obtiveram-se os melhores resultados utilizando o algoritmo Árvore de Decisão. Nos resultados do experimento final, foi possível observar o aperfeiçoamento e a consistência motora. Utilizando os dados do experimento final, obtiveram-se os melhores resultados com o AutoML. Assim sendo, o trabalho mostrou a possibilidade de estimação da quantidade de sessões para atingir determinado desempenho utilizando algoritmos de predição. Adicionalmente, ressalta-se a relevância do trabalho, uma vez que este servirá de base para experimentos futuros com mais participantes saudáveis, assim como pessoas com dano motor.

Palavras-chave: Tarefa de Traçado. Aquisição de Habilidade Motora. Modelos de Classificação.

Abstract

The present work aims to propose an approach to estimate the number of sessions required to learn a motor task. Motor activities are the main way of interacting with the world. Therefore, loss of ability to perform some of these activities as a result of a neurological disease is a serious injury to the individual. In the literature, there are many works on motor learning, mostly looking for ways to decrease the time of skill acquisition or motor rehabilitation. However, few works concentrate on trying to estimate the training time needed to achieve certain motor performance. The methodology consisted of a review of the state of the art of motor skill acquisition, as well as the initial configuration of a training platform, the application of a pilot experiment with three participants and a final experiment with eight participants. In the pilot experiment, a three-block training session for each participant was performed and it aimed to predict in which block the participant was. From three real participants, 18 simulated participants were generated, in order to measure the performance of the experiment with more participants, and the block was estimated through the average performance of the participants. In the final experiment, three sessions were performed for each participant, whose purpose was to predict in which session the participant would reach a certain error based on their profile and initial performance. The classification models used in the final experiment were: Algorithm K-Neighbors Nearer, Neural Network, Decision Tree, Support Vector Machine and Automatic Machine Learning (AutoML) with "Auto Weka". In the results of the pilot experiment, an improvement of motor skills was observed after the training. Through the data from the pilot experiment, the best results were obtained using the Decision Tree algorithm. In the results of the final experiment, it was possible to observe the motor improvement and the consistency. Using the data from the final experiment, the best results were obtained with AutoML. The work showed the possibility of estimating the number of sessions to achieve a certain performance using prediction algorithms. In addition, the relevance of the work is accentuated, since this will serve as a basis for future experiments with more healthy participants, as well as people with motor damage.

Keywords: Tracing Task. Motor Skill Learning. Classification models.

Lista de ilustrações

Figura 1 – Funcionamento do algoritmo KNN	21
Figura 2 – Árvore de decisão	23
Figura 3 – SVM genérico	24
Figura 4 – Rede neural MLP	25
Figura 5 – Tarefa de traçado	30
Figura 6 – Método de captura do erro na tarefa de traçado	31
Figura 7 – Desenho experimental do experimento final	32
Figura 8 – Matriz de confusão	36
Figura 9 – Desempenho dos participantes do experimento piloto	38
Figura 10 – Acurácia dos modelos	39
Figura 11 – Medida F dos modelos	39
Figura 12 – Coeficiente kappa dos modelos	40
Figura 13 – Desempenho médio dos participantes do experimento final	42
Figura 14 – Desempenho médio dos participantes durante os blocos no experimento final	43
Figura 15 – Acurácia dos modelos	44
Figura 16 – Medida F dos modelos	44
Figura 17 – Coeficiente kappa dos modelos	45
Figura 18 – Acurácia das abordagens tradicionais e AutoML	47
Figura 19 – Medida F das abordagens tradicionais e AutoML	48
Figura 20 – Coeficiente Kappa das abordagens tradicionais e AutoML	48
Figura 21 – Buffer do Auto Weka - classificador A	56
Figura 22 – Buffer do Auto Weka - classificador B	57
Figura 23 – Buffer do Auto Weka - classificador C	58

Lista de tabelas

Tabela 1 – Participantes do experimento final	29
Tabela 2 – Resultados da acurácia dos modelos	40
Tabela 3 – Resultados da medida F dos modelos	40
Tabela 4 – Resultados do coeficiente kappa dos modelos	40
Tabela 5 – Teste ANOVA dos modelos de classificação nos dados simulados	41
Tabela 6 – Test post hoc - Tukey de múltiplas comparações entre médias	41
Tabela 7 – Desempenho médio dos participantes do experimento final	42
Tabela 8 – Teste ANOVA do erro percentual dos participantes de experimento final	43
Tabela 9 – Resultados da acurácia do experimento	43
Tabela 10 – Resultados da medida F do experimento	45
Tabela 11 – Resultados do coeficiente kappa do experimento	45
Tabela 12 – Teste ANOVA dos modelos de classificação do experimento final	46
Tabela 13 – Test post hoc - Tukey de múltiplas comparações entre médias	46
Tabela 14 – Resultados da aplicação de AutoML e metodologias tradicionais	47

Lista de abreviaturas e siglas

AUC	Área abaixo da curva ROC
r^2	Coefficiente de determinação
ROC	Características operacionais do receptor
tDCS	Estimulação Transcraniana por Corrente Direta
tRNS	Estimulação Transcraniana por Ruído Aleatório
AutoML	Aprendizagem de Máquina Automática
FIM	Medida de independência funcional
AVC	Acidente Vascular Cerebral
MLP	Perceptrons de Múltiplas Camadas
RL	Regressão Linear
KRR	Regressão Kernel Ridge
ME	Mistura de Especialistas Lineares
AD	Árvore de Decisão
KNN	K - Vizinhos mais Próximos
ANOVA	Análise de Variância
AutoML	Aprendizado de Máquina Automático
SVM	Máquina com Suporte Vetorial
VP	Verdadeiros Positivos
VN	Verdadeiros Negativos
FP	Falsos Positivos
FN	Falsos Negativos

Sumário

1	INTRODUÇÃO	14
1.1	Objetivos	14
1.2	Trabalhos Relacionados	15
1.3	Contribuições	17
1.4	Organização do Trabalho	17
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	Aprendizado Motor e Reabilitação	18
2.2	Aprendizado de Máquina	19
2.2.1	Tarefas do Aprendizado de Máquina	20
2.2.2	Modelos de Classificação	20
3	METODOLOGIA	28
3.1	Materiais	28
3.2	Participantes	28
3.3	Desenho Experimental	29
3.3.1	Amostras Complementares do Experimento Piloto	32
3.3.2	Modelos de Classificação	33
3.3.3	Validação	35
3.3.4	AutoML	37
4	RESULTADOS E DISCUSSÃO	38
4.1	Experimento Piloto	38
4.1.1	Aperfeiçoamento Motor	38
4.1.2	Análise Comparativa dos Modelos com Dados Simulados	39
4.2	Experimento Final	41
4.2.1	Aperfeiçoamento Motor e Consistência Motora	41
4.2.2	Análise Comparativa dos Modelos com Dados Experimentais	43
4.2.3	Aplicação de AutoML no Aprendizado Motor	47
5	CONCLUSÃO	49
	REFERÊNCIAS	51

APÊNDICES	55
APÊNDICE A – CLASSIFICADOR A	56
APÊNDICE B – CLASSIFICADOR B	57
APÊNDICE C – CLASSIFICADOR C	58
ANEXOS	59
ANEXO A – EDINBURGH INVENTORY	60
ANEXO B – BECK DEPRESSION INVENTORY	62

1 Introdução

Em nossas vidas muito tempo é gasto aprendendo novas habilidades motoras como caminhar, escrever, dirigir, praticar esportes, entre outras. As atividades motoras são a principal forma de interagir com o mundo que nos rodeia (PRICHARD *et al.*, 2014). Portanto a perda da capacidade de realizar algumas dessas atividades, como resultado de uma doença neurológica é um dano grave ao indivíduo.

Quando uma doença neurológica causa um dano ao indivíduo, este deve adquirir uma nova habilidade motora que busque novas formas de ativar a musculatura para realizar a tarefa com o melhor desempenho possível (KITAGO; KRAKAUER, 2013). Dependendo da complexidade da tarefa, as seções de treinamento podem estender-se durante dias, meses e até anos para alcançar o desempenho desejável.

Embora a literatura sobre aprendizado motor seja numerosa, principalmente tratando das características aperfeiçoamento e consistência motora (Seção 2), ainda há uma lacuna no tocante a predição da aquisição da habilidade motora. A predição da quantidade de seções ou blocos de treinamento motor que um indivíduo (no caso, saudável ou portador de uma doença que não afeta o aprendizado motor) levará para adquirir um determinado desempenho em uma tarefa é um exemplo de estudo a ser explorado.

Segundo Jordan e Mitchell (2015), algoritmos de predição vêm sendo aplicados com sucesso em diversas áreas como saúde, educação, economia, ciência, tecnologia e logística. Com o passar do tempo uma quantidade massiva de dados vêm sendo gerados e necessita-se de aplicações que processem e gerem informações que auxiliem na tomada de decisões. Estes algoritmos de predição também tornam possível a análise e obtenção de novos padrões e associações para problemas que outrora eram tratados apenas por pessoas especialistas em um determinado campo.

Lin *et al.* (2018) propuseram a aplicação de algoritmos de aprendizado de máquina para predição de atividades do cotidiano para pacientes pós Acidente Vascular Cerebral - AVC, obtendo uma correlação moderada entre as características do paciente e o Índice de Barthel. O Índice de Barthel, referenciado por Hsueh *et al.* (2002), reflete o nível de independência de um paciente em tarefas do cotidiano.

1.1 Objetivos

O objetivo principal deste trabalho é propor uma abordagem para estimar a quantidade de sessões necessárias para aprender uma tarefa motora.

Como objetivos específicos têm-se:

1. Reproduzir a metodologia da Tarefa de traçado do trabalho de [Prichard et al. \(2014\)](#);
2. Observar as propriedades do Aprendizado Motor no Experimento aplicado;
3. Aplicar modelos de predição presentes na literatura para estimar a aquisição de habilidade motora;
4. Comparar o desempenho de diferentes modelos de predição para o problema aqui introduzido.

1.2 Trabalhos Relacionados

Atualmente, estudos de diversas áreas como ciência, tecnologia, comércio tem gerado uma quantidade massiva de dados. A análise destes dados tem sido mais complexa e com custo computacional mais árduo ([JORDAN; MITCHELL, 2015](#)).

Há uma grande diversidade de técnicas de reabilitação motora clássicas e recém descobertas, porém em todas estas o período de prática é determinado de forma arbitrária, ou seja, o paciente realiza o treinamento motor sem uma projeção com uma boa aproximação de quando alcançará um determinado limiar de aptidão na tarefa.

[Tsuji et al. \(2000\)](#) propuseram um modelo de predição para avaliação da progressão do AVC tendo como métrica a “medida de independência funcional” (FIM) motora. Os dados foram provenientes de um centro de reabilitação motora no Japão. Através do uso de regressão múltipla *Stepwise*, os autores obtiveram um coeficiente de determinação igual a 0,68.

[Sonoda et al. \(2005\)](#) utilizaram características do paciente para predizer a “medida de independência funcional” (FIM) motora. Esta métrica é um dos principais referenciais utilizados para avaliar reabilitação motora. O estudo contou com dados de 87 pacientes de reabilitação após o AVC. Através de regressão linear múltipla (RL), foi possível alcançar um coeficiente de correlação igual 0,88 nos dados de validação.

No trabalho de [Hahne et al. \(2014\)](#), os autores utilizaram técnicas de regressão linear e não linear para o controle mioelétrico independente, simultâneo e proporcional de próteses com dois graus de liberdade. As próteses são geralmente aplicadas em pacientes com membros amputados, então estes têm que adquirir com o tempo a habilidade motora de controlá-las. Foram usadas as técnicas: Regressão Linear - RL, Perceptrons de Múltiplas Camadas - MLP, Regressão Kernel Ridge - KRR e Mistura de Especialistas Lineares - ME (técnica inspirada na regressão linear). A originalidade do trabalho deve-se à KRR e ME não terem sido utilizados em trabalhos anteriores para a mesma tarefa, segundo o conhecimento dos autores. Como resultado foi mostrado que ME é promissor para o controle de próteses com 2 graus de liberdade devido o bom ajuste aos comandos e custo computacional baixo, que requer menor complexidade de hardware.

Forkert et al. (2015) propuseram o uso do método de máquina com suporte vetorial para predição do resultado de dano funcional em pacientes de AVC. Este resultado é essencial para traçar alternativas de abordagens de reabilitação. Os melhores resultados apresentaram 0.85 de acurácia.

Sprint et al. (2015) avaliaram a aplicação de dados de sensores inerciais vestíveis no suporte a decisões clínicas em reabilitação. Estes dados são utilizados para prever a “medida de independência funcional” (FIM) de pacientes de reabilitação. Com os dados de 20 pacientes foram treinados algoritmos de aprendizado de máquina a fim de prever a FIM. Os métodos utilizados foram: Regressão com Suporte Vetorial, Regressão Linear e Regressão por Floresta Aleatória. A técnica que obteve melhores resultados foi a Regressão com Suporte Vetorial com coeficiente de correlação de Pearson igual a 0,97.

Sato et al. (2016) criaram um modelo de predição para estimar o nível de independência nas atividades do cotidiano em pacientes após o AVC. Utilizou um banco de dados com características de 65 pacientes onde tinha como parâmetros a idade, tempo após o AVC, pontuação na sub escala de independência funcional motora, pontuação na sub escala de independência funcional cognitiva, conjunto de avaliação de redução do AVC, Escala de equilíbrio de Berg e Índice de vitalidade para prever a pontuação na sub escala de independência funcional motora futura. Para a obtenção do modelo, os autores utilizaram a Regressão Múltipla *Stepwise* e obtiveram como resultado um coeficiente de determinação igual a 0,60, apresentando correlação moderada.

O trabalho de Julianjatsono, Ferdiana e Hartanto (2017) tem o objetivo de prover um método de avaliação de função motora com maior resolução para facilitar a detecção de pequenas melhoras em pacientes. Na literatura é largamente utilizada a Avaliação Fugl-Meyer com resolução de 3 níveis em cada item. A proposta é escalar estes 3 níveis em 14 níveis utilizando métodos de regressão. Os métodos de regressão utilizados foram: Regressão Linear, Regressão Linear Bayesiana, Rede Neural para Regressão, Regressão por Floresta de Decisão e Regressão por Árvore de Decisão. Os melhores resultados foram do método de Rede Neural para Regressão com coeficiente de correlação de Pearson igual a 0,58.

No trabalho de Sale et al. (2018), foi feita a predição através de aprendizado de máquina da melhoria do paciente que recebeu tratamento de reabilitação na fase inicial após o AVC. A coleta de dados foi feita com 55 pacientes que receberam tratamento na fase inicial do AVC. A melhoria dos paciente é medida usando a “pontuação de independência funcional” (FIM) motora. O algoritmo que obteve o melhor desempenho foi a Regressão com Suporte Vetorial com *kernel* Linear com coeficiente de correlação entre 0,71 e 0,81 para diferentes configurações.

Lin et al. (2018) fizeram a predição de atividades do cotidiano de pacientes iniciando a reabilitação pós Acidente Vascular Cerebral – AVC. Utilizando uma base de dados de um

hospital em Taiwan entre 2014 e 2016, eles buscaram prever o índice Barthel utilizando algoritmos de aprendizado de máquina. O índice Barthel é um índice de atividades do dia a dia que um paciente consegue fazer, ou seja, seu nível de independência em tarefas cotidianas (HSUEH et al., 2002). Lin et al. (2018) conseguiu uma boa predição com o algoritmo Máquina com Suporte Vetorial - SVM (Área sob a curva (AUC) = 0,77), com erro absoluto médio aceitável. O Coeficiente de Determinação apresentou uma correlação moderada ($r^2 = 0,57$).

1.3 Contribuições

As contribuições deste trabalho são:

- Expandir a literatura acerca de algoritmos de predição aplicados à estimação de aprendizado motor;
- Mostrar a possibilidade de prever em qual sessão de treinamento um indivíduo saudável alcançará determinado desempenho na tarefa de traçado;
- Mostrar algumas propriedades do aprendizado motor (aperfeiçoamento e consistência motora) através do erro percentual obtido através dos experimentos da tarefa de traçado.

1.4 Organização do Trabalho

Este trabalho é composto por cinco capítulos. Neste capítulo, fez-se uma introdução ao tema, bem como situou o contexto deste trabalho em relação à literatura atual do tema. No Capítulo 2, foi feita uma fundamentação teórica necessária para o entendimento dos conceitos que serão aplicados na metodologia e nas análises dos resultados deste trabalho. No Capítulo 3, são descritos os materiais e métodos que tornaram possível a construção e validação dos resultados deste trabalho. O Capítulo 4 é destinado aos resultados obtidos nos experimentos aplicados e discussões a fim de interpretá-los com base na literatura do tema. No Capítulo 5, conclui-se o trabalho destacando os principais resultados e propondo perspectivas futuras para este trabalho.

2 Fundamentação Teórica

2.1 Aprendizado Motor e Reabilitação

Segundo [Kitago e Krakauer \(2013\)](#), "o sistema motor humano tem a capacidade de aprender com a prática e a experiência". Esta oração é central para introduzir o conceito de aprendizagem motora, bem como o que se pretende realizar nesta pesquisa. Como o sistema motor é capaz de aprender com a prática e a experiência com o passar do tempo, o ser humano aperfeiçoa-se na tarefa que está sendo executada.

Um aspecto que deve ser levado em consideração é que o indivíduo pode apresentar um aperfeiçoamento motor relevante após minutos de prática ou, dependendo da complexidade da tarefa, até meses.

Outra característica importante é a consistência que explica como a experiência influencia no aprendizado motor. Por essa característica, o indivíduo tende a errar menos com o passar do tempo, diminuindo a variabilidade das execuções da tarefa ([KITAGO; KRAKAUER, 2013](#); [DAYAN; COHEN, 2011](#)).

Outro aspecto importante da aprendizagem motora é o estado inicial em que se encontra o sistema motor do indivíduo. Este estado inicial influencia no resultado final após uma seção de treinamento, ou seja, o mesmo protocolo de treinamento pode levar a resultados diferentes dependendo do estado inicial do indivíduo ([SHMUELOF; KRAKAUER, 2014](#)).

Ainda segundo [Kitago e Krakauer \(2013\)](#), a aprendizagem motora divide-se em duas áreas: adaptação motora e aquisição de habilidade. Por adaptação, entende-se que esta está relacionada à capacidade do sistema motor responder para alterar as condições ambientais para recuperar o nível de desempenho nas novas configurações do ambiente, ou seja, adaptar-se às novidades impostas por algum novo obstáculo.

Entende-se por aquisição de habilidade motora a capacidade de adquirir novos padrões de ativação muscular e um melhor desempenho pela redução dos erros sem a redução da velocidade de execução do movimento, baseado no trabalho de [Hallett e Grafman \(1997\)](#). Esta definição contrasta com a de adaptação motora. Outra peculiaridade da aquisição em relação à adaptação é que, enquanto a adaptação ocorre durante apenas uma seção, a aquisição de habilidade motora requer um estendido período de prática, podendo durar vários dias, meses ou até anos dependendo da complexidade da tarefa ([KITAGO; KRAKAUER, 2013](#)). O ganho de desempenho pode ser aferido durante as seções (efeitos *online*) e entre as seções (efeitos *offline*) ([KARNI et al., 1998](#)).

Segundo [Korman et al. \(2003\)](#), quanto maior o número de repetições de uma tarefa, melhor o desempenho nesta, ou seja, a prática é o método mais efetivo de obtenção de desempenho. Outros fatores no planejamento da prática podem também contribuir para uma melhor retenção da habilidade motora.

[Krakauer \(2006\)](#) diz que reabilitação motora é um processo para reaprender a mover-se para realizar uma tarefa com sucesso. Ainda nesse trabalho, o autor cita diversas abordagens de reabilitação motora para recuperar aptidão em diversas tarefas como: reabilitação baseada em realidade virtual; terapia com interação robótica; terapia de movimento induzido por restrições; treinamento de capacidade de braço para pacientes com hemiparesia leve.

No trabalho de [Reis et al. \(2009\)](#), investigou-se o efeito da estimulação não invasiva do cérebro via *Transcranial Direct Current Stimulation* - tDCS (Estimulação Transcraniana por Corrente Direta) na consolidação do aprendizado motor. Nesse trabalho, foi analisada a tarefa "*Pinch Task*", com a execução de cinco sessões ou dias de treinamento, cada sessão possuía nove blocos, cada bloco possuía 10 "*trials*" ou treinos. Como resultados principais obteve-se a função "*Speed-Accuracy Tradeoff*" para esta tarefa e encontrou-se a evidência que a aplicação da estimulação tDCS contribui apenas para os efeitos *offline*, mas não para os efeitos *online* ou retenção a longo prazo.

A função "*Speed-Accuracy Tradeoff*" trata da velocidade do movimento. Quando se aumenta a velocidade do movimento, perde-se acurácia, ou seja, aumenta-se o erro. Então a acurácia por si só não é uma boa medida de desempenho, segundo [Kitago e Krakauer \(2013\)](#).

[Prichard et al. \(2014\)](#) mostraram os efeitos de diferentes protocolos de estimulação não invasiva no cérebro, tDCS e *Transcranial Random Noise Stimulation* - tRNS (cuja tradução livre é Estimulação Transcraniana por Ruído Aleatório), nos subcomponentes de aprendizagem de habilidade motora. A tarefa utilizada nesse trabalho foi a tarefa de traçado. Foram executadas três sessões diárias, cada sessão possui 12 blocos, cada bloco possui 15 "*trials*". Como resultados principais, obteve-se que os efeitos do tDCS e tRNS são diferentes. Enquanto os efeitos do tDCS contribuíram mais para os efeitos *online*, os efeitos tRNS são mais graduais e puderam ser observados em efeitos *offline*. É importante frisar que esse trabalho também fornece uma metodologia para a captura do erro percentual da tarefa de traçado.

2.2 Aprendizado de Máquina

Por muito tempo, a solução de problemas com o uso da computação baseou-se na codificação explícita dos passos necessários para conclusão de uma tarefa específica. Entretanto, com o avanço das pesquisas na área e evolução da capacidade de processamento

dos computadores, a Aprendizagem de Máquina tornou-se algo cotidiano, presente em aplicações de Negócios, Educação, Transporte, Segurança, Saúde e qualquer área que demande um processamento de uma grande massa de dados e extração de conhecimentos que ajude na solução de problemas complexos.

Uma definição clássica para aprendizado de máquina foi enunciada por [Mitchell \(1997\)](#) como: a capacidade de, por meio de experiências, melhorar o desempenho na realização de uma tarefa.

Por experiência, entende-se como um conjunto de exemplos da tarefa que se objetiva solucionar. Assim é possível encontrar uma função através de um algoritmo de aprendizado de máquina que resolva um problema tendo como entrada um conjunto de exemplos, sem a necessidade da codificação explícita dos passos para resolver determinado problema.

2.2.1 Tarefas do Aprendizado de Máquina

Os algoritmos de aprendizado podem ser classificados seguindo diversos critérios. Um destes critérios divide o aprendizado de máquina em Aprendizado Supervisionado e Aprendizado Não Supervisionado.

O aprendizado supervisionado consiste em encontrar um modelo ou uma função que a partir de um conjunto de exemplos possa prever um determinado parâmetro de uma nova amostra. Dependendo da natureza numérica do parâmetro que deseja-se prever, o aprendizado supervisionado pode dividir-se em modelos de classificação e modelos de regressão. Modelos de classificação são aplicados quando a classe que se deseja prever é discreta, enquanto que, modelos de regressão são aplicados quando a saída é contínua.

O aprendizado não supervisionado tem o objetivo de explorar ou descrever um conjunto de dados. Neste tipo de tarefa agrupam-se os dados por semelhança ou criam-se regras para a divisão dos dados.

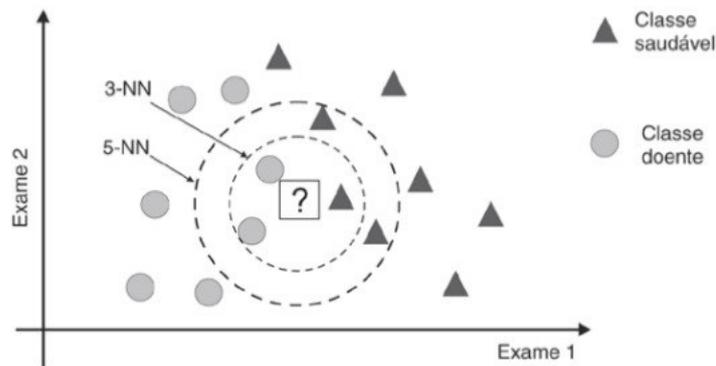
2.2.2 Modelos de Classificação

K-Vizinhos Mais Próximos

O algoritmo K-vizinhos mais próximos (KNN) é um dos algoritmos mais simples de aprendizagem de máquina. Este algoritmo baseia-se em distâncias. Para escolher a qual classe um novo elemento pertence, o KNN busca os k elementos do grupo de treinamento mais próximos do novo elemento. A classe mais frequente entre os k elementos de treinamento mais próximos será atribuída ao novo elemento, conforme ilustrado na [Figura 1](#).

Na [Figura 1](#), tem-se duas classes, saudável e doente, e um elemento desconhecido. Se considerarmos o $k = 3$ o elemento desconhecido será classificado como doente, pois dois

Figura 1 – Funcionamento do algoritmo KNN



Os triângulos cinza escuro representam a classe saudável. Os círculos cinza claro representam a classe doente. O quadrado com símbolo de interrogação representa um elemento desconhecido. Os círculos tracejados representam a delimitação dos métodos 5-NN e 3-NN, respectivamente. Na abscissa, tem-se o Exame 1 para determinar se o sujeito está saudável ou doente, e na ordenada o Exame 2.

Fonte: [Carvalho et al. \(2011\)](#).

dos três elementos mais próximos do desconhecido pertencem à classe doente. Entretanto, ao considerarmos o $k = 5$, o elemento desconhecido será classificado como saudável, pois três dos cinco elementos mais próximos do desconhecido pertencem à classe saudável. Através deste exemplo, é possível perceber a importância da escolha do valor de k para uma eficiente classificação.

O Algoritmo 1 mostra um pseudocódigo do KNN.

Algoritmo 1: Algoritmo do KNN adaptado de ([MARSLAND, 2015](#))

Entrada: X_{tr} : um conjunto de treinamento com exemplos rotulados

Resultado: Função de Estimação

```

1 para  $i \leftarrow 1$  a  $X_{tr}$  faça
2   | Calcular a distancia de  $i$  para todos os exemplos de treinamento;
3   | Selecionar os  $k$  exemplos mais próximos de  $i$ ;
4   |  $i =$  classe mais frequente dos  $k$  exemplos mais próximos;
5 fim

```

Para o cálculo da distância na linha 2 do Algoritmo 1, podem ser utilizadas diferentes métricas. O objetivo da utilização de diferentes métricas de distância é a possibilidade de evidenciar características que podem ser importantes no processo de predição. Nas Equações 2.1 e 2.2, referenciadas por [Gonzalez e Woods \(2010\)](#), podem-se ver diferentes cálculos de distância, a saber, Euclidiana e Manhattan (ou City Block), respectivamente.

$$d_{euclidiana} = \sqrt{\sum_{i=1}^n (x_{2i} - x_{1i})^2} \quad (2.1)$$

$$d_{manhattan} = \sum_{i=1}^n |x_{2i} - x_{1i}| \quad (2.2)$$

Nas Equações 2.1 e 2.2, x_2 representa um elemento ainda não classificado e x_1 , um elemento cuja classe é conhecida.

Árvore de Decisão

Uma árvore de decisão é um algoritmo de aprendizado de máquina que usa o princípio da divisão e conquista para resolver problemas de decisão. Um problema difícil é dividido em subproblemas. Por sua vez, os subproblemas resolvidos são agrupados em uma árvore, produzindo uma solução para o problema complexo. Alguns algoritmos implementados com base em árvores de decisão possuem a mesma ideia básica.

Uma árvore de decisão pode ser formalmente definida como um grafo acíclico direcionado em que os nós podem ser de dois tipos: de divisão ou folha (MARS LAND, 2015).

Um nó folha pode ser definido como uma função. Por simplicidade, a função é a constante que minimiza a função custo. Assumindo o custo no intervalo $[0, 1]$, essa constante é a moda estatística, que se refere ao valor mais frequente do conjunto.

Um nó de divisão possui um teste de condição baseado nos valores do domínio do parâmetro. Este teste geralmente pode ter como resposta verdadeiro ou falso, dependendo do valor assumido pelo parâmetro que está sendo avaliado. Geralmente, os testes envolvem apenas um parâmetro por nó de divisão.

Na Figura 2, tem-se uma representação gráfica de uma árvore de decisão. Cada nó folha corresponde a uma classe. Neste exemplo, têm-se dois atributos de entradas, x_1 e x_2 e cinco classes de saída. O teste presente nos nós de divisão consiste em decidir se a condição é verdadeira (T) ou falsa (F).

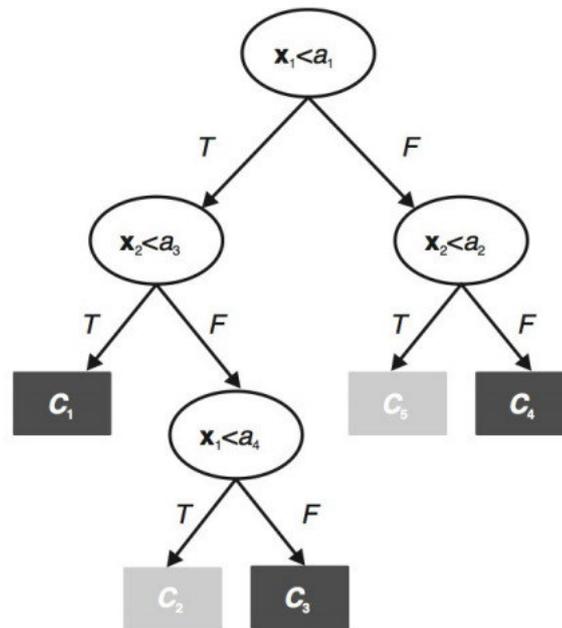
Um algoritmo genérico de uma árvore de decisão pode ser implementado recursivamente. Seja D um conjunto de dados. A função recursiva **Gera Árvore 2** implementa uma árvore de decisão.

Máquina com Suporte Vetorial

O algoritmo máquina com suporte vetorial (SVM) tem como princípio fundamental a construção de um hiperplano que maximiza a margem de separação entre as classes. Segundo Vapnik (1998), o hiperplano é construído com base em um conjunto de pontos chamados de vetores de suporte. Os vetores de suporte são os pontos que maximizam a margem que separa as classes. A Figura 3 ilustra o funcionamento de um algoritmo SVM genérico.

Na Figura 3, M representa a margem que maximiza a separação entre a classe de bolas e a classe de cruces.

Figura 2 – Árvore de decisão



As elipses são os nós de divisão. As setas correspondem aos valores que podem ser assumidos pelo teste condicional: T (Verdadeiro) e F (Falso). Os retângulos correspondem aos nós folha. Dentro das elipses encontram-se testes condicionais, em que x_1 e x_2 são parâmetros de entrada e a_1 , a_2 , a_3 e a_4 são valores que pertencem ao domínio dos parâmetros.

Fonte: [Carvalho et al. \(2011\)](#).

Algoritmo 2: Algoritmo da Árvore de Decisão adaptado de ([CARVALHO et al., 2011](#))

Entrada: D : um conjunto de treinamento com exemplos rotulados

Resultado: Árvore de Decisão

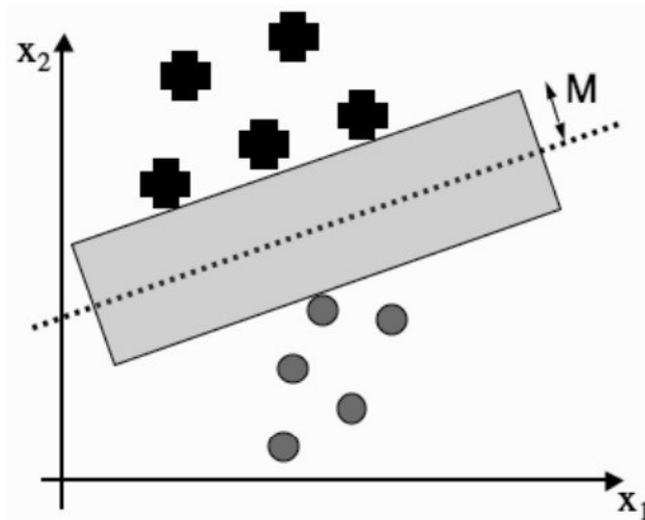
```

1 /* Função Gera Árvore */;
2 Se Critério de Parada( $D$ ) = Verdadeiro então
3   | Retorna um nó folha rotulado com a constante que minimiza a função de perda;
4 Escolha o atributo que maximiza o critério de divisão em  $D$ 
5 para Partição de exemplos  $D_i$  baseado nos valores do atributo escolhido faça
6   | Induz uma subárvore  $Arvore_i = \text{Gera Árvore}(D_i)$ ;
7 fim
8 Retorna uma Árvore contendo um nó de decisão baseado no atributo escolhido, e
   descendentes  $Arvore_i$ 

```

Existem vários algoritmos SVM. Entretanto, uma característica que torna estes algoritmos aplicáveis a uma vasta gama de problemas é a possibilidade de transformar o espaço de entradas em um novo espaço de características, através de funções de kernel. Segundo ([VAPNIK, 1998](#)), a função de kernel pode realizar o mapeamento das entradas para um espaço multidimensional sem aumentar a complexidade dos cálculos. A grande vantagem da aplicação destas funções de kernel é poder tornar os problemas linearmente separáveis através de diferentes funções. Alguns exemplos de função de kernel são linear

Figura 3 – SVM genérico



As cruzeiros pretos representam a classe das cruzeiros. As bolas cinzas representam a classe das bolas. O retângulo representa a margem de separação entre as classes, enquanto que a linha pontilhada representa a fronteira. x_1 e x_2 são parâmetros de entrada do classificador.

Fonte: Marsland (2015).

(Equação 2.3), função de base radial (Equação 2.4), polinomial (Equação 2.5) e logística (Equação 2.6).

$$K(x, y) = x^T y \quad (2.3)$$

$$K(x, y) = \exp(-(x - y)^2 / 2\sigma^2) \quad (2.4)$$

$$K(x, y) = (1 + x^T y)^s \quad (2.5)$$

$$K(x, y) = \tanh(\kappa x^T y - \delta) \quad (2.6)$$

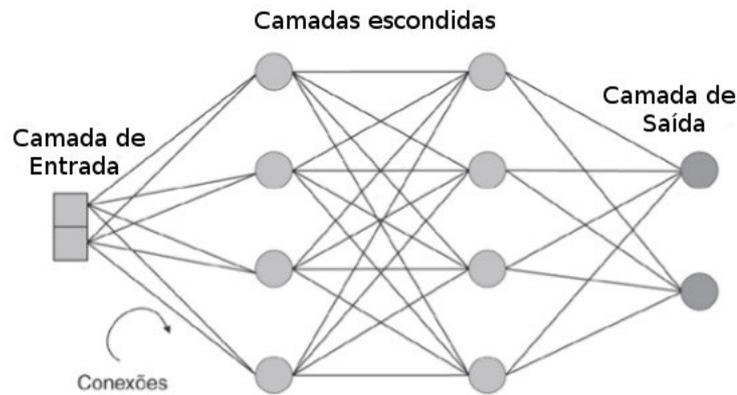
Nas quatro equações acima, x representa o espaço de parâmetros preditores ou independentes e y é o espaço de parâmetros de resposta ou dependentes. Os parâmetros s , δ e κ são escolhidos pelo programador na inicialização do método.

Rede Neural Perceptron com Múltiplas Camadas

As Redes Neurais Artificiais são técnicas que foram inspiradas no sistema nervoso humano, mais especificamente na comunicação e geração do conhecimento a nível dos neurônios (HAYKIN, 2004). Existem várias abordagens e um dos modelos mais populares na literatura é o perceptron com múltiplas camadas (MLP).

As redes neurais MLP têm a estrutura baseada em camadas: uma camada de entrada, uma ou mais camadas escondidas e uma camada de saída. Essa estrutura pode ser vista na Figura 4.

Figura 4 – Rede neural MLP



Os círculos cinza representam os neurônios ou nós de processamento da rede neural. As linhas representam as conexões entre os neurônios. Os quadrados cinza representam os neurônios da camada de entrada.

Fonte: Baseado em [Carvalho et al. \(2011\)](#).

As redes neurais possuem partículas elementares chamadas de neurônios ou nós de processamento e o conjunto de neurônios é chamado de camada. Na Figura 4 os neurônios são representados por círculos ou quadrados. Na rede neural MLP, todos os neurônios são interconectados. A entrada de um neurônio é definida pela Equação 2.7.

$$net_h = \sum_{i=1}^n out_i * w_{hi} \quad (2.7)$$

onde, net_h representa a entrada do neurônio da camada escondida, i define o neurônio da camada anterior, out_i é a saída do neurônio i e w_{hi} é o peso da conexão entre os neurônios h e i . O comportamento das demais camadas será melhor representado no Algoritmo 3 de treinamento.

Existem vários algoritmos utilizados para fazer o treinamento destas redes, dentre eles, um dos mais populares é o *Backpropagation*. O Algoritmo 3 mostra a implementação de uma rede neural MLP. Nesse Algoritmo 3, w_{jl} representa o peso entre o neurônio l e o j -ésimo atributo de entrada ou a saída do j -ésimo neurônio da camada anterior, δ_l é o erro associado ao l -ésimo neurônio, x^j é a entrada recebida pelo neurônio atual e η é a constante de aprendizado, definida pelo programador.

Um parâmetro variável relacionado à redes neurais são as funções de ativação que são utilizadas para calcular o valor da saída dos neurônios, geralmente presentes nas camadas escondidas e de saída. Exemplos de funções: sigmoide (Equação 2.8), tangente hiperbólica (Equação 2.9) e linear. A referencia da notação matemática das funções de ativação é ([MARSLAND, 2015](#)).

Algoritmo 3: Algoritmo da Rede Neural Artificial MLP com algoritmo de treinamento Backpropagation adaptado de (CARVALHO et al., 2011)

Entrada: Um conjunto de n objetos de treinamento
Resultado: Função de Estimção com valores de pesos ajustados

- 1 Iniciam-se os pesos com valores aleatórios pequenos;
- 2 Inicializar $erro_{total} = 0$;
- 3 **repita**
- 4 **para** cada elemento x_i do conjunto de treinamento **faça**
- 5 **para** cada camada da rede, a partir da primeira camada intermediária **faça**
- 6 **para** cada neurônio n_{jl} da camada atual **faça**
- 7 | Calcular o valor da saída produzida pelo neurônio, \hat{f} ;
- 8 **fim**
- 9 **fim**
- 10 Calcular $erro_{parcial} = y - \hat{f}$;
- 11 **fim**
- 12 **para** cada camada da rede, a partir da camada de saída **faça**
- 13 **para** cada neurônio n_{jl} da camada atual **faça**
- 14 | Ajustar pesos do neurônio: $w_{jl}(t+1) = w_{jl}(t) + \eta x^j \delta_l$;
- 15 **fim**
- 16 **fim**
- 17 Calcular $erro_{total} = erro_{total} + erro_{parcial}$;
- 18 **até que** $erro_{total} < \xi$;

$$g_1(x) = \frac{1}{1 + \exp(-x)} \quad (2.8)$$

$$g_2(x) = \tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \quad (2.9)$$

Aprendizado de máquina com seleção de modelos automática

Os algoritmos de aprendizado de máquina possuem a capacidade de aprender padrões ou funções de acordo com o conjunto de dados de treinamento. Apesar disso, os parâmetros que devem ser escolhidos pelo programador na inicialização do treinamento podem influenciar positiva ou negativamente o desempenho do algoritmo. A complexidade da escolha dos melhores parâmetros aumenta de acordo com a quantidade destes.

Segundo Chicco (2017), os hiper-parâmetros dos algoritmos de aprendizado de máquina são propriedades que possuem grande influência na complexidade, velocidade de aprendizado e nos resultados da aplicação. São exemplos de hiper-parâmetros o número k de vizinhos no algoritmo KNN, a quantidade de neurônios da camada escondida de uma rede neural MLP e a função kernel do algoritmo SVM, entre outros.

Para encontrar os melhores modelos e melhores hiper-parâmetros para estes modelos,

podem-se utilizar algoritmos de otimização. Alguns exemplos de algoritmos de otimização utilizados para esta finalidade presentes na literatura são: random search (BERGSTRA; BENGIO, 2012), grid search (BERGSTRA; BENGIO, 2012), algoritmo colônia de vagalumes (ZHANG et al., 2018), enxame de partículas caótico (HONG, 2009), algoritmo de cardume de peixes artificial (LI; SUOHAI, 2013), otimização bayesiana (KOTTHOFF et al., 2017).

Nestes trabalhos, os algoritmos de otimização têm facilitado a seleção dos melhores hiper-parâmetros para os algoritmos de aprendizado de máquina e apresentado melhores resultados que a abordagem tradicional de seleção de parâmetros pela intuição do programador.

3 Metodologia

3.1 Materiais

Na fase de aquisição de dados, foi utilizada uma estação de trabalho "Dell Optiplex 7050", processador i7, 16GB de memória RAM, sistema operacional Ubuntu 16.04 com Python 2.7 configurado e bibliotecas necessárias instaladas. As bibliotecas necessárias para a plataforma de treinamento motor são: PyOpenCV versão 3, Tkinter, Pygame e Pillow. Estas configurações foram adequadas para o *software* que fez a captura dos dados (CARVALHO, 2018). Foi utilizada também uma mesa digitalizadora "One by Wacom CTL471L" de dimensão 21 cm x 14,6 cm, área ativa de 15,2 cm x 9,5 cm e 1024 níveis de pressão.

Na fase de processamento e análise dos dados, foi utilizado um *notebook* Dell 14 com a seguinte configuração: processador i3, 8GB de memória RAM, sistema operacional Ubuntu 16.04 com os *softwares* Python 3.7 e Weka instalados e configurados.

3.2 Participantes

No experimento piloto, os participantes foram estudantes da Universidade Federal do Maranhão. Inicialmente, um grupo de cinco pessoas saudáveis, ou seja, pessoas sem restrição motora foram voluntários no experimento. Todos os participantes forneceram autorização por escrito, a fim de que seus dados pudessem ser usados em uma publicação, enquanto que ao mesmo tempo este trabalho manterá a confidencialidade dos participantes.

Foi aplicado aos participantes o questionário "Edinburgh Handedness Inventory" (ESPÍRITO-SANTO et al., 2017). Este questionário é usado para saber a mão com a qual o indivíduo realiza a maioria das atividades. Foram considerados neste estudo os participantes que utilizam frequentemente apenas a mão direita.

Para a seleção dos participantes, também foi realizado um histórico neurológico, onde se perguntou para os candidatos a participantes do experimento se este tem ou já teve qualquer sintoma neurológico que poderia ocasionar uma alteração de seu desempenho motor. Os candidatos que apresentaram algum sintoma neurológico foram excluídos deste estudo. Ao final do recrutamento e aplicação do experimento piloto, apenas três completaram a tarefa.

Para o experimento final, os participantes foram estudantes e docentes da Universidade Federal do Maranhão. Foram recrutados 15 participantes saudáveis. Todos os participantes forneceram autorização por escrito, a fim de que seus dados pudessem

ser usados em uma publicação, enquanto que ao mesmo tempo este trabalho, de forma semelhante ao experimento inicial, manterá a confidencialidade dos participantes.

Foi aplicado aos participantes o questionário “*Edinburgh Handedness Inventory*” (ESPÍRITO-SANTO et al., 2017). Foram considerados neste estudo os participantes que utilizam frequentemente apenas a mão direita. Outro questionário aplicado foi “*Beck Depression Inventory*”. Este questionário é capaz de detectar sintomas de depressão. Candidatos a participantes com índice maior que 12 foram excluídos deste estudo. Ambos questionários estão disponíveis nos Anexos A e B.

Para a seleção dos participantes, de forma semelhante ao experimento inicial, também foi realizado um histórico neurológico e os candidatos que apresentaram algum destes serão excluídos deste estudo. Ao final do recrutamento e aplicação do experimento final, apenas oito completaram a tarefa.

Na Tabela 1, tem-se a demografia geral dos participantes deste experimento e de outros relacionados a este. Entretanto, ressalta-se que se tentou aproximar a demografia de experimentos relacionados a este, embora os outros possuíssem objetivos diferentes do presente trabalho. As informações dos participantes coletadas para este experimento estão dispostas na Tabela 1.

Tabela 1 – Participantes do experimento final

Experimento	n	Idade $\pm\sigma$	Edinburg $\pm\sigma$	Masculino (%)
Trabalho atual	8	23,88 \pm 4,94	75,19 \pm 26,32	62,50
Reis et al. (2009)	12	32,00 \pm 3,00	-	58,33
Prichard et al. (2014)	18	25,70 \pm 4,60	93,4 \pm 10,7	38,90

3.3 Desenho Experimental

Foi criado um *software* (CARVALHO, 2018) para a tarefa de traçado com o *framework* “Pygame” (SHINNERS, 2011) da linguagem de programação Python (ROSSUM, 1995) com base no experimento proposto por Prichard et al. (2014). Aos participantes, foi requerido que cobrissem uma série de palavras com o auxílio de uma mesa digitalizadora com a sua mão esquerda. A tarefa utilizada para o treinamento motor neste trabalho foi a tarefa de traçado, que pode ser ilustrada na Figura 5.

As palavras utilizadas foram escolhidas de uma base de dados gratuita (Invoke IT Limited ¹). As palavras escolhidas contêm de três a cinco letras do português usado no Brasil. As palavras foram escritas com uma fonte cursiva gratuita (League Script²).

¹ <http://invokeit.wordpress.com/frequency-word-lists/>

² www.theleagueofmoveabletype.com

Figura 5 – Tarefa de traçado



Para controlar um cursor na tela, o participante movimentava a caneta sob a mesa digitalizadora. O participante tenta reproduzir com a caneta na mesa digitalizadora a palavra disposta na tela. O erro percentual é usado para determinar uma medida de habilidade.

Fonte: Próprio Autor.

A métrica extraída da tarefa de traçado é o erro percentual da diferença entre a imagem de fundo e a imagem traçada pelo participante. Ambas imagens passaram por um processo de borragem para que os participantes que não acertarem perfeitamente a forma do desenho não sejam prejudicados. O processo de borragem consiste em realizar a convolução da imagem com um *kernel* Gaussiano de tamanho 50x50 pixels. A Figura 6 ilustra o processamento feito nas imagens.

Ressalta-se que a aquisição das imagens, filtragem e função de erro não fazem parte do objetivo deste trabalho, sendo que esses métodos estão implementados no *software* previamente desenvolvido por outro membro do projeto (CARVALHO, 2018).

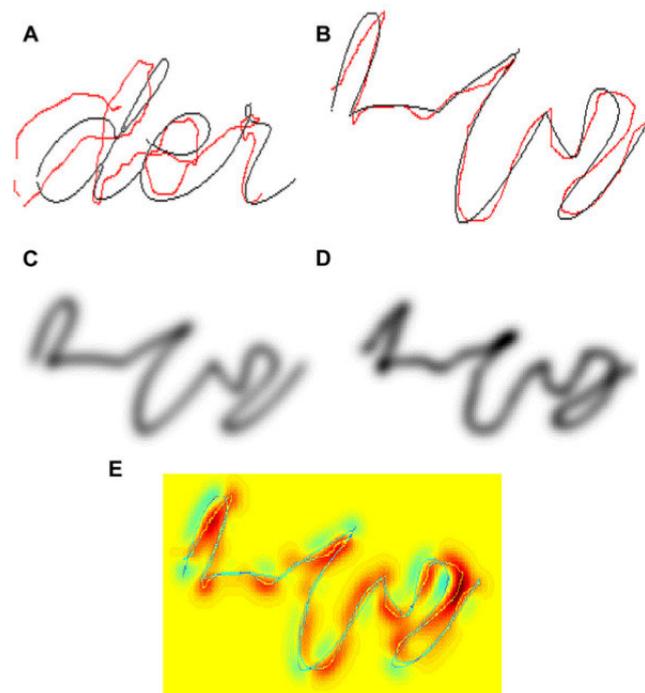
Experimento Piloto

Um dos objetivos deste experimento piloto é extrair um tempo médio por letra para o experimento final, de forma que os treinos fiquem difíceis porém factíveis, conforme Prichard et al. (2014). Então uma das características armazenadas neste experimento foi o tempo de execução do movimento.

O experimento foi composto de uma sessão, onde cada sessão teve três blocos e cada bloco continha 10 treinos de palavras com cinco letras. O tempo foi capturado após a conclusão de cada treino. Em seguida, foi calculado o erro percentual, conforme descrito anteriormente.

O cálculo do tempo médio por letra com base no experimento é descrito pela

Figura 6 – Método de captura do erro na tarefa de traçado



A e **B** são exemplos de treino. Podem ser utilizadas como treinos tanto palavras como em **A** quanto aproximações de palavras como em **B**. Em **C**, tem-se a figura **B** borrada conforme o método descrito anteriormente. Em **D**, tem-se o traçado que foi feito sob a figura **B** borrado. Em **E**, tem-se uma imagem da subtração pixel a pixel da imagem **C** da **D**. Se o traçado for feito na região amarela ou não for feito nenhum traçado o erro é máximo. Se o traçado for feito tanto na região verde quanto na região vermelha será considerado acerto e não haverá acréscimo no erro percentual.

Fonte: Prichard et al. (2014).

Equação 3.1. Neste trabalho, considerou-se o tempo médio por letra como a média do tempo por letra do bloco de todos os treinos realizados.

$$t_{letra} = \frac{\sum_{i=1}^{num_{treinos}} t_{palavra}(i)}{tam_{palavra} * num_{treinos}} \quad (3.1)$$

Na Equação 3.1, t_{letra} corresponde ao tempo por letra a ser utilizado no experimento final. Para calcular este tempo, foram utilizados número de treinos ($num_{treinos}$), o tempo por palavra ($t_{palavra}$) e o tamanho da palavra ($tam_{palavra}$) medido em quantidade de letras.

Após a aquisição dos dados, estes foram formatados em Valor Separados por Vírgulas - CSV (do inglês, “Comma Separated Value”) para que pudessem ser feitos o processamento e a análise. O formato CSV foi escolhido pela simplicidade e por ser largamente utilizado em análises estatísticas. No arquivo CSV, foram armazenados o tempo de execução, o erro percentual do movimento e o número do bloco.

Experimento Final

Cada participante fez três sessões - uma sessão por dia - com nove blocos e cada bloco com 10 treinos. Cada treino é uma palavra que deve ser coberta. Os participantes

foram submetidos aos mesmos exemplos de treinos. Neste experimento o tempo é constante, conforme o cálculo que foi feito no experimento piloto será de 10 segundos por letra. O desenho experimental pode ser ilustrado pela Figura 7.

Figura 7 – Desenho experimental do experimento final

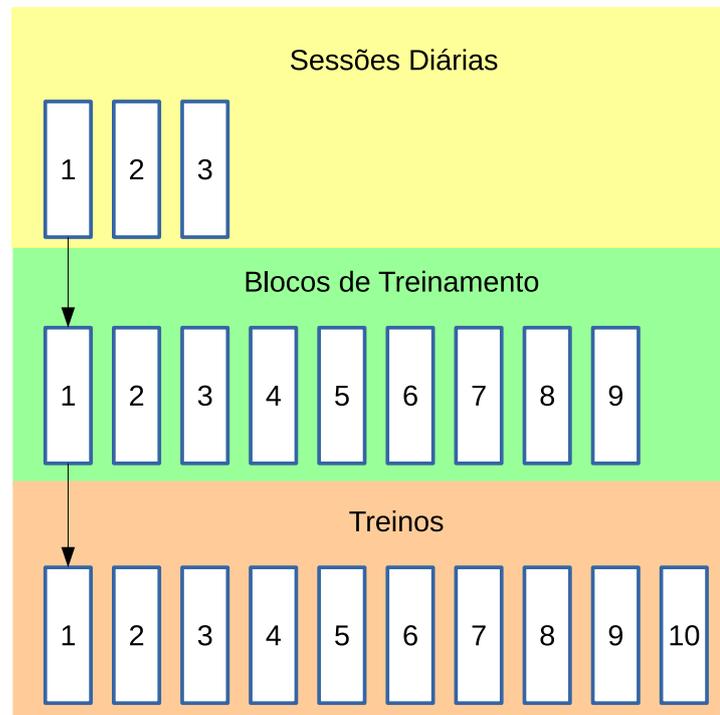


Ilustração de como se dará a distribuição dos treinos nos dias de prática. O Experimento se dará em 3 sessões diárias, cada sessão possui 9 blocos e cada bloco possui 10 treinos.

Fonte: Próprio autor.

Após a aquisição dos dados, estes foram formatados em Valor Separados por Vírgulas - CSV (do inglês, "Comma Separated Value") para que pudessem ser feitos o processamento e a análise. O formato CSV foi escolhido pela simplicidade e por ser largamente utilizado em análises estatísticas. No arquivo CSV, foram armazenados a idade do participante, o gênero, o erro percentual do movimento na sessão 1, o erro percentual do movimento na sessão atual, o número da sessão atual.

3.3.1 Amostras Complementares do Experimento Piloto

Amostras complementares foram criadas a partir das obtidas pelo experimento piloto a fim de utilizar satisfatoriamente os modelos de classificação. Essas amostras foram referentes ao desempenho médio dos três blocos, para as duas medidas de desempenho: tempo e erro percentual.

A partir do dado real, soma-se a este um ruído gaussiano com média zero e desvio padrão 0.01 (valor definido com base na magnitude dos dados). Foram gerados

18 participantes com base nos três participantes reais para os três blocos, a saber, seis participantes com base em cada participante real.

3.3.2 Modelos de Classificação

Todos os métodos de predição utilizados são provenientes da biblioteca “SciKit Learn” (PEDREGOSA et al., 2011) da linguagem de programação Python (ROSSUM, 1995). Esta tem suporte à diversas técnicas de aprendizado de máquina, bem como técnicas de validação relacionadas a estas. Esta biblioteca é largamente utilizada em trabalhos acadêmicos, devido a grande quantidade de técnicas presentes nela e a praticidade da linguagem Python.

No experimento piloto, foram utilizadas os parâmetros dos classificadores inicialmente definidos com pequenas alterações. Os melhores modelos no Experimento Final foram selecionados por um algoritmo chamado “*Grid Search*”, que faz uma busca exaustiva buscando os melhores parâmetros para os classificadores dentro do espaço de busca delimitado.

Experimento Piloto

Os parâmetros (além dos inicialmente definidos pela biblioteca) utilizados para os algoritmos foram:

- K-Vizinhos mais Próximos:
 - Variáveis preditoras: erro percentual médio e tempo médio;
 - Variável de resposta: bloco;
 - Quantidade de vizinhos: 3;
 - Distância: euclidiana.

- Máquina com Suporte Vetorial:
 - Variáveis preditoras: erro percentual médio e tempo médio;
 - Variável de resposta: bloco;
 - Kernel: RBF.

- Rede Neural Artificial MLP:
 - Variáveis preditoras: erro percentual médio e tempo médio;
 - Variável de Resposta: Bloco;
 - Função de Ativação: Sigmoide;

- Camada escondida: 10 neurônios;
- Épocas: 10000
- Constante de Aprendizagem: 0.01;
- Solver: Método Quasi-Newton;
- Árvore de Decisão:
 - Variáveis Predictoras: Erro Percentual Médio e Tempo Médio;
 - Variável de Resposta: Bloco;
 - Configurações Iniciais Inalteradas;

Experimento Final

Os parâmetros selecionados pelo algoritmo *"Grid Search"* e utilizados foram:

- K-Vizinhos mais Próximos:
 - Variáveis Predictoras: Idade, Gênero, Erro Médio da Sessão Inicial, Erro Médio da Sessão Atual;
 - Variável de Resposta: Sessão;
 - Quantidade de Vizinhos: 9;
 - Distância: Euclidiana;
 - Algoritmo: *"Ball Tree"*;
- Máquina com Suporte Vetorial:
 - Variáveis Predictoras: Idade, Gênero, Erro Médio da Sessão Inicial, Erro Médio da Sessão Atual;
 - Variável de Resposta: Sessão;
 - Fator de penalidade por erro 'C': 64;
 - Termo independente no kernel 'Coef0': 0;
 - Kernel: Sigmoid;
- Rede Neural Artificial MLP:
 - Variáveis Predictoras: Idade, Gênero, Erro Médio da Sessão Inicial, Erro Médio da Sessão Atual;
 - Variável de Resposta: Sessão;

- Função de Ativação: Relu;
 - Camada escondida: 100 neurônios;
 - Épocas: 5000
 - Constante de Aprendizagem: 0.01;
 - Solver: Método Quasi-Newton;
- Árvore de Decisão:
 - Variáveis Predictoras: Idade, Gênero, Erro Médio da Sessão Inicial, Erro Médio da Sessão Atual;
 - Variável de Resposta: Sessão;
 - Critério de Qualidade de Divisão: Entropia;
 - Número Mínimo de Amostras em cada Divisão: 64;
 - Método de Divisão: “Best”;

3.3.3 Validação

Para avaliação do desempenho da predição, foi aplicada a validação cruzada “k-fold” com 5 folds no Experimento piloto e 10 “folds” no Experimento final. Isso foi feito para manter o conjunto de treinamento e teste o mais balanceados possível, conforme é feito em [Hahne et al. \(2014\)](#) e [Lemm et al. \(2011\)](#).

A validação “k-fold”, segundo [Norvig e Russell \(2014\)](#), pode ser usada para obter uma estimativa precisa do desempenho de uma técnica. A ideal central é a de um mesmo exemplo servir tanto para treinamento quanto para teste. A base de dados é dividida em k-conjuntos de igual tamanho. Realizam-se k rodadas de treinamento em que $\frac{1}{k}$ dos dados servem para teste e os demais para treinamento. A estimativa final é a média do valor obtido nas k rodadas. Segundo [Norvig e Russell \(2014\)](#), valores comumente utilizados de k são 5 e 10, suficientes para apresentar uma estimativa estatisticamente provável que seja precisa.

Como métricas de avaliação de desempenho dos classificadores para o experimento piloto e final foram utilizadas a Acurácia, Medida F e o coeficiente Kappa.

As métricas de avaliação são calculadas com base na matriz de confusão. A matriz de confusão é uma forma largamente utilizada para facilitar a avaliação de modelos de classificação. Por ilustração, consideremos uma das classes como negativa (-) e a outra como positiva (+). Uma matriz de confusão genérica está ilustrada na Figura 8.

Na Figura 8, Verdadeiros Positivos (VP) corresponde ao número de elementos da classe positiva classificados corretamente; Verdadeiros Negativos (VN) corresponde

Figura 8 – Matriz de confusão

		Classe predita	
		+	-
Classe verdadeira	+	VP	FN
	-	FP	VN

Fonte: [Carvalho et al. \(2011\)](#).

ao número de elementos da classe negativa classificados corretamente; Falsos Positivos (FP) corresponde ao número de elementos cuja classe verdadeira é negativa porém foram classificados como pertencentes à classe positiva; Falsos Negativos (FN) corresponde ao número de elementos cuja classe verdadeira é positiva porém foram classificados como pertencentes à classe negativa. O número total de elementos n é igual à $VP+VN+FP+FN$. Assim as métricas podem ser calculadas com base nos valores da matriz como segue nas Equações 3.2, 3.5 e 3.8.

$$Acuracia = \frac{VP + VN}{n} \quad (3.2)$$

$$Recall = \frac{VP}{VP + FN} \quad (3.3)$$

$$Precisao = \frac{VP}{VP + FP} \quad (3.4)$$

$$MedidaF = \frac{2 * Precisao * Recall}{Precisao + Recall} \quad (3.5)$$

$$P_o = VP + VN \quad (3.6)$$

$$P_e = (VP + FP) * (VP + FN) + (FN + VN) * (FP + VN) \quad (3.7)$$

$$Kappa = \frac{P_o - P_e}{1 - P_e} \quad (3.8)$$

A Medida F é uma métrica que afere o equilíbrio entre a precisão e o *recall*. O coeficiente kappa é a medida de concordância da predição. Em análises estatísticas clínicas, utiliza-se o coeficiente kappa para aceitar ou rejeitar uma hipótese estimada segundo [Altman \(1999\)](#). Quando o coeficiente kappa é igual a 1 tem-se uma perfeita concordância.

3.3.4 AutoML

Para a seleção de modelos de aprendizado de máquina automática, foi utilizado o *software* “Weka”, referenciado por Frank, Hall e Witten (2016). O “Weka” é uma coleção de algoritmos de aprendizado de máquina para tarefas de mineração de dados, neste pode-se realizar pré-processamento, classificação, regressão, agrupamento, regras de associação e visualização. Especificamente para a seleção de modelos, existe um “*plugin*” chamado “Auto Weka” referenciado por Thornton et al. (2013) e Kotthoff et al. (2017).

O “Auto Weka” utiliza otimização bayesiana para procurar o melhor modelo e melhores hiperparâmetros deste dentro de um determinado tempo. Quanto maior o tempo, maior é a possibilidade de encontrar bons resultados se estes forem possíveis.

Neste trabalho foram executados vários testes e o critério de seleção foi que estes deveriam apresentar desempenho melhor que os classificadores KNN, SVM, MLP e árvore de decisão implementados. A duração definida para os testes foi de duas horas. Foram executados 20 testes e destes três atenderam ao critério de seleção.

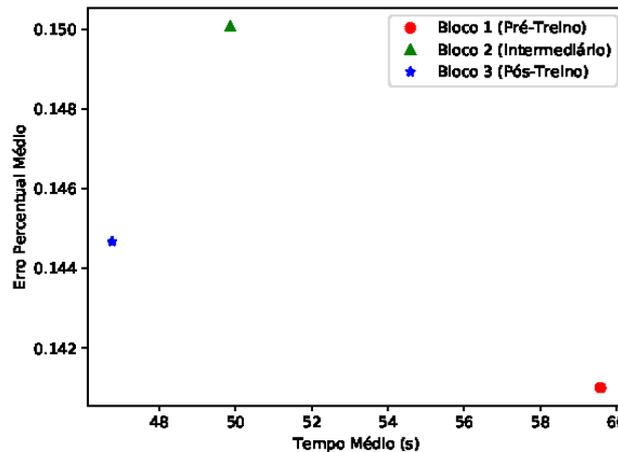
4 Resultados e Discussão

4.1 Experimento Piloto

4.1.1 Aperfeiçoamento Motor

Segundo Kitago e Krakauer (2013) e Reis et al. (2009), o aprendizado motor ocorre quando há um equilíbrio entre velocidade e acurácia na execução de um movimento ou tarefa. O aperfeiçoamento motor é uma característica do aprendizado motor cuja ocorrência pode ser percebida quando tem-se uma diminuição do tempo e do erro de execução de uma tarefa depois de um período de prática. A característica do aperfeiçoamento motor pode ser observada na Figura 9.

Figura 9 – Desempenho dos participantes do experimento piloto



O ponto vermelho representa o desempenho médio dos participantes no Bloco 1 (Pré-Treino). O triângulo verde representa o desempenho médio dos participantes no Bloco 2 (Intermediário). A estrela azul representa o desempenho médio dos participantes no Bloco 3 (Pós-Treino). O desempenho médio é uma função do tempo médio de movimento (abscissa) e o erro percentual médio por bloco (ordenadas).

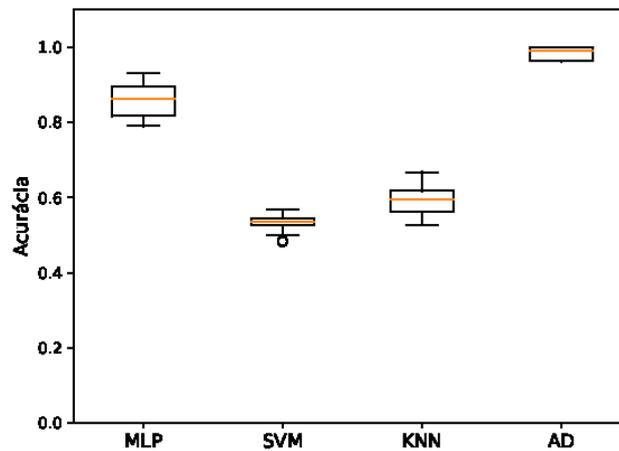
Fonte: Próprio Autor.

Na Figura 9, pode-se perceber que há um aperfeiçoamento motor comparando o Pré-Treino com o Pós-Treino. No Pré-Treino, os participantes tiveram em média um erro percentual pequeno porém o tempo médio de execução foi grande em relação aos demais blocos. No Intermediário, os participantes diminuíram o tempo médio porém aumentaram o erro percentual médio. No Pós-Treino, é possível observar um balanceamento entre tempo e erro percentual médio de execução da tarefa. Por estes aspectos, pode-se deduzir qualitativamente que houve um aperfeiçoamento motor entre o Bloco 1 e Bloco 3, corroborando com a literatura a respeito do tema.

4.1.2 Análise Comparativa dos Modelos com Dados Simulados

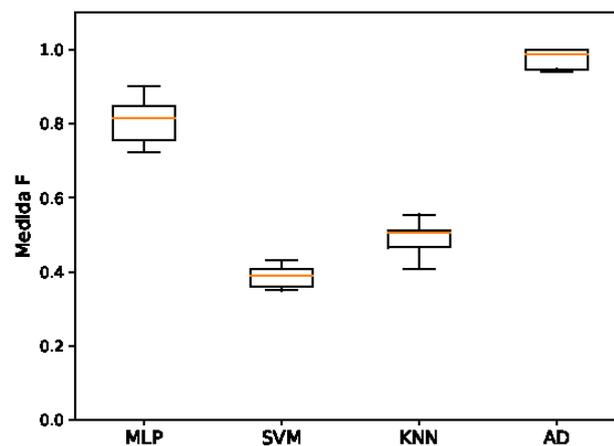
Os modelos de aprendizado de máquina foram treinados a partir dos 18 dados de participantes simulados a partir do experimento piloto. As Figuras 10, 11 e 12 mostram os gráficos de caixa do desempenho dos modelos de classificação na estimação da aquisição de habilidade motora para a tarefa de traçado com dados simulados.

Figura 10 – Acurácia dos modelos



MLP representa o desempenho da rede neural, SVM representa o desempenho da Máquina com Suporte Vetorial, KNN representa o algoritmo dos K vizinhos mais próximos e AD representa o desempenho da Árvore de Decisão.
Fonte: Próprio Autor.

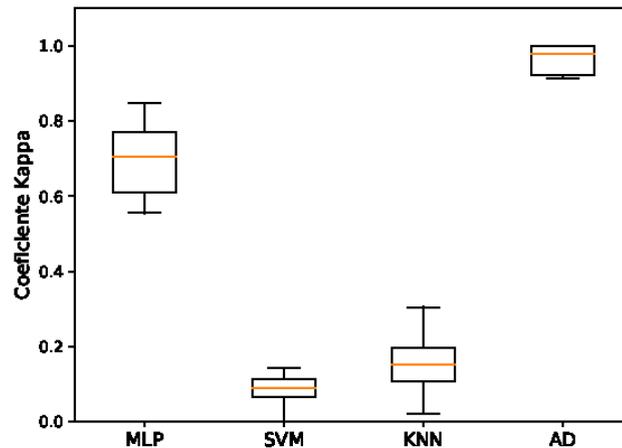
Figura 11 – Medida F dos modelos



MLP representa o desempenho da rede neural MLP, SVM representa o desempenho da Máquina com Suporte Vetorial, KNN representa o algoritmo dos K vizinhos mais próximos e AD representa o desempenho da Árvore de Decisão.
Fonte: Próprio Autor.

Nas Tabelas 2, 3 e 4, aparecem os resultados deste experimento. O algoritmo árvore de decisão apresentou um desempenho superior aos demais algoritmos com acurácia média. Para as outras métricas, medida F e coeficiente kappa, a árvore de decisão também foi superior.

Figura 12 – Coeficiente kappa dos modelos



MLP representa o desempenho da rede neural MLP, SVM representa o desempenho da Máquina com Suporte Vetorial, KNN representa o algoritmo dos K vizinhos mais próximos e AD representa o desempenho da Árvore de Decisão.

Fonte: Próprio Autor.

Tabela 2 – Resultados da acurácia dos modelos

Algoritmo	Máximo	Média	Mínimo	Desvio padrão
MLP	0,9307843137	0,8585375438	0,7904545455	0,0468765757
KNN	0,667826087	0,597611669	0,5280464217	0,0462140203
SVM	0,5676317038	0,5317400289	0,4833333333	0,024330562
AD	1,0000000000	0,9842236467	0,9626068376	0,0175351712

Tabela 3 – Resultados da medida F dos modelos

Algoritmo	Máximo	Média	Mínimo	Desvio padrão
MLP	0,90063593	0,8080552436	0,7243243243	0,0580112876
KNN	0,5543478261	0,4943216965	0,4070121951	0,0446859157
SVM	0,4300847458	0,3877944382	0,35	0,0289899458
AD	1,0000000000	0,9760575899	0,941031941	0,0268039746

Na Tabela 5, tem-se o resultado do teste estatístico de análise de variância (ANOVA) para checar se existe diferença entre os desempenhos dos algoritmos. Para as análises estatísticas, foi usada a linguagem de programação R. Em todos os testes estatísticos, o valor-p < 0,05 foi considerado como estatisticamente relevante.

Na Tabela 5, todos os testes apresentaram valor-p menor que 0,01. Isto sinaliza que

Tabela 4 – Resultados do coeficiente kappa dos modelos

Algoritmo	Máximo	Média	Mínimo	Desvio padrão
MLP	0,8475946285	0,6961175851	0,5553210209	0,0967046745
KNN	0,3047200278	0,1629704512	0,0214635992	0,0883789185
SVM	0,1420183568	0,0830236356	0,001010101	0,0434929453
AD	1,0000000000	0,9643163028	0,9138148668	0,0397978619

Tabela 5 – Teste ANOVA dos modelos de classificação nos dados simulados

Métrica	Df	Df.res	Sum Sq	Sum Sq.res	Mean Sq	valor-p
Acurácia	3	36	1,3731	0,0471	0,4577	< 0.001
Medida F	3	36	2,2319	0,0623	0,7440	< 0.001
Kappa	3	36	5,393	0,186	1,7977	< 0.001

há uma diferença estatística entre o desempenho dos algoritmos no geral. Fez-se também um teste post hoc para verificar a diferença estatística entre os pares de algoritmos. O teste utilizado foi o "Tukey com múltiplas comparações entre médias". O resultado do valor-p para todas as métricas e modelos de classificação está disponível na Tabela 6.

Tabela 6 – Test post hoc - Tukey de múltiplas comparações entre médias

Algoritmos	p (Acurácia)	p (Medida F)	p (Kappa)
KNN-AD	0,0000000	0,0000000	0,0000000
MLP-AD	0,0000000	0,0000000	0,0000000
SVM-AD	0,0000000	0,0000000	0,0000000
MLP-KNN	0,0000000	0,0000000	0,0000000
SVM-KNN	0.0013375	$9,3 * 10^{-6}$	0.0788257
SVM-MLP	0,0000000	0,0000000	0,0000000

Na Tabela 6, nota-se que para quase todas as combinações a diferença estatística foi estatisticamente relevante (valor-p < 0,01). Somente o coeficiente kappa para a dupla SVM e KNN apresentou valor-p > 0,05, então não apresentou diferença estatisticamente relevante.

Como estes testes foram realizados com dados simulados estima-se que ao fazer um experimento com 18 participantes obtenha-se resultados significativos para a área. Na prática, resultados de acurácia entre 0,8 e 0,9 condizem com o estado da arte da pesquisa em predição de aprendizado motor, conforme [Lin et al. \(2018\)](#).

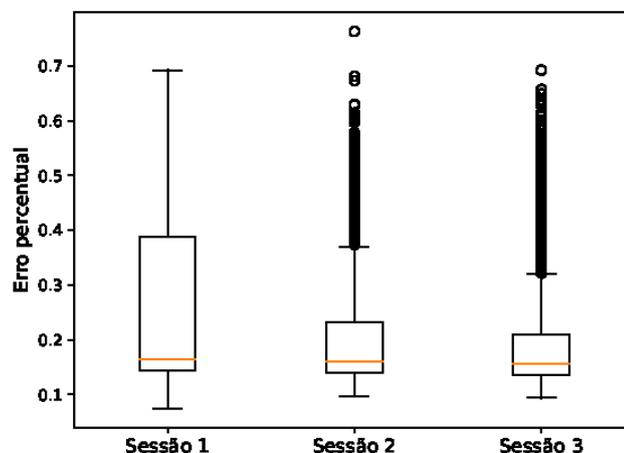
4.2 Experimento Final

4.2.1 Aperfeiçoamento Motor e Consistência Motora

Algumas características do aprendizado (aperfeiçoamento e consistência) motor puderam ser observadas no experimento final, como ilustra a Figura 13.

No experimento final, o tempo de execução das sessões é constante, então o aperfeiçoamento motor pode ser observado com o decaimento do erro no decorrer da prática motora. Na Figura 13, e na Tabela 7 percebe-se a diminuição do erro percentual médio com o passar das sessões de treinamento. Isto evidencia o aperfeiçoamento motor

Figura 13 – Desempenho médio dos participantes do experimento final



O gráfico apresenta a dispersão do desempenho médio dos participantes em termos do erro percentual médio dos participantes do experimento final.

Fonte: Próprio Autor.

na tarefa de traçado, uma das principais características do aprendizado motor, segundo Kitago e Krakauer (2013) e Dayan e Cohen (2011).

Na Figura 13 e Tabela 7, também é perceptível a diminuição do desvio padrão. A diminuição do desvio padrão no desempenho médio dos participantes pode ser atribuído à consistência motora. A consistência motora é o decaimento da variabilidade do desempenho no decorrer da prática. Portanto, neste experimento também observou-se a consistência motora, uma importante característica do aprendizado motor segundo Kitago e Krakauer (2013) e Dayan e Cohen (2011).

Tabela 7 – Desempenho médio dos participantes do experimento final

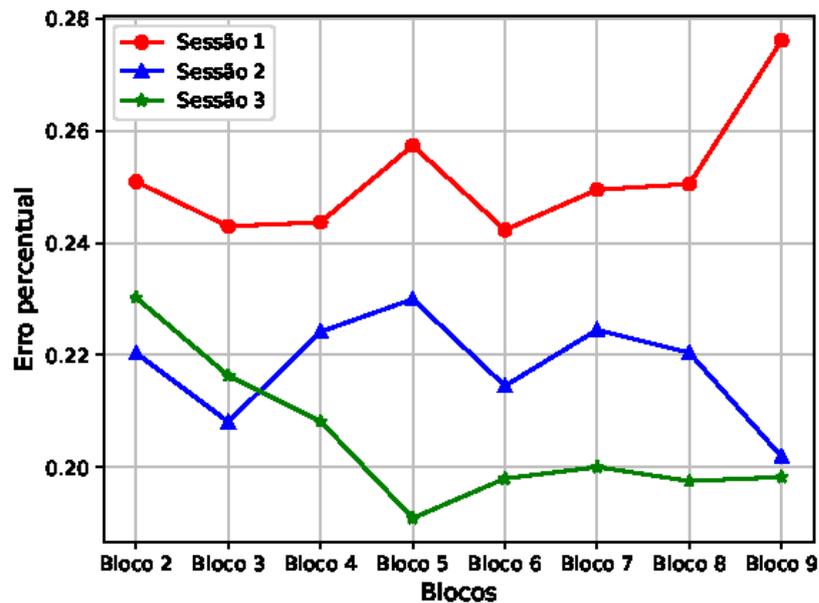
Sessão	Erro percentual médio	Desvio padrão
1	0,2577378993	0,1671990008
2	0,220223997	0,1326769511
3	0,2075684137	0,1236305743

Na Figura 14, observa-se o aperfeiçoamento durante os blocos de treinamento.

Na Figura 14, percebe-se nitidamente um aperfeiçoamento entre a sessão 1 e as demais sessões. Durante o começo da sessão 3 o erro percentual é maior que a sessão 2. A partir do bloco 4 o erro da sessão 3 é menor que o da sessão 2. Porém, na maior parte dos blocos, o erro percentual médio segue a relação: sessão 1 > sessão 2 > sessão 3.

Mesmo com a diminuição da média do erro percentual durante as sessões foi feito o teste estatístico de Análise de Variância com Medidas Repetidas (ANOVA) para checar se a diferença entre os resultados dos participantes é estatisticamente relevante. Na Tabela 8, tem-se o resultado do teste ANOVA.

Figura 14 – Desempenho médio dos participantes durante os blocos no experimento final



As bolas vermelhas representam o desempenho entre blocos durante a sessão 1. Os triângulos azuis representam o desempenho entre blocos durante a sessão 2. As estrelas verdes representam o desempenho entre blocos durante a sessão 3. Na ordenada, tem-se o erro percentual e na abscissa os blocos de treinamento.

Fonte: Próprio Autor.

Tabela 8 – Teste ANOVA do erro percentual dos participantes de experimento final

Métrica	Df	Sum Sq	Mean Sq	Valor-p
Erro Percentual	2	0,010892	0,0054459	0,433

Como pode-se perceber na Tabela 8, o valor-p é maior que 0,05, então a diferença estatística entre as médias não é estatisticamente relevante.

4.2.2 Análise Comparativa dos Modelos com Dados Experimentais

As Figuras 15, 16 e 17 mostram os gráficos de caixa do desempenho dos modelos de predição aplicados à aquisição de habilidade motora para a tarefa de traçado nos dados obtidos no experimento final.

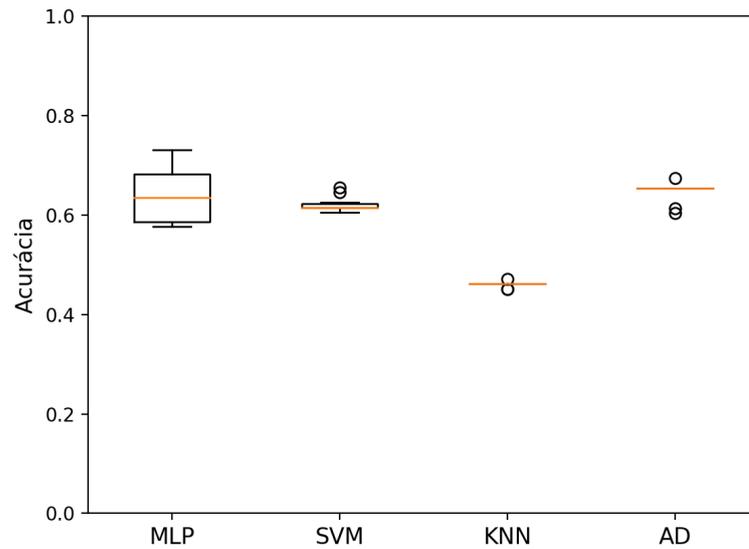
Na Tabela 9, estão dispostos os resultados do experimento para a métrica acurácia.

Tabela 9 – Resultados da acurácia do experimento

Algoritmo	Máximo	Média	Mínimo	Desvio padrão
MLP	0,7307692308	0,6384615385	0,5769230769	0,0549938733
KNN	0,4715384615	0,4605384615	0,4515384615	0,0056764621
SVM	0,6553846154	0,6223846154	0,6053846154	0,0156702124
AD	0,6738461538	0,6468461538	0,6038461538	0,0211081869

A acurácia média para as técnicas MLP, SVM e árvore de decisão foi bem próxima, porém, como pode-se perceber na Figura 15 e na Tabela 9, o valor máximo da MLP foi

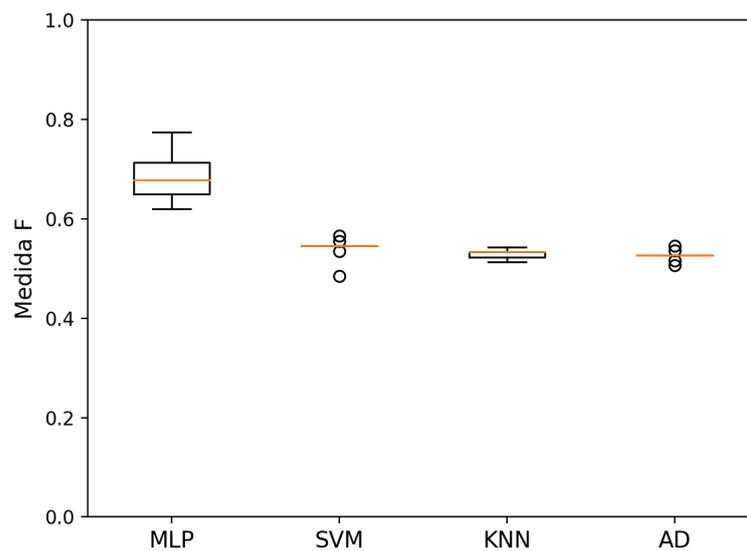
Figura 15 – Acurácia dos modelos



MLP representa o desempenho da rede neural, SVM representa o desempenho da Máquina com Suporte Vetorial, KNN representa o algoritmo dos K vizinhos mais próximos e AD representa o desempenho da Árvore de Decisão.
Fonte: Próprio Autor.

superior aos demais. O KNN apresentou desempenho inferior às demais técnicas quanto a acurácia.

Figura 16 – Medida F dos modelos



MLP representa o desempenho da rede neural, SVM representa o desempenho da Máquina com Suporte Vetorial, KNN representa o algoritmo dos K vizinhos mais próximos e AD representa o desempenho da Árvore de Decisão.
Fonte: Próprio Autor.

Na Tabela 10, estão dispostos os resultados do experimento para a métrica medida F.

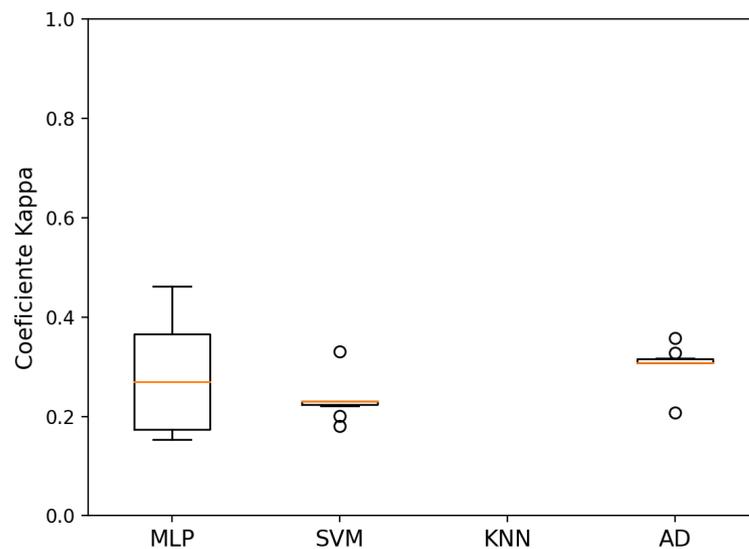
Na Figura 16, que trata da medida F, observa-se que a rede neural MLP apresentou um melhor desempenho que às demais técnicas. Enquanto que as demais técnicas, SVM,

Tabela 10 – Resultados da medida F do experimento

Algoritmo	Máximo	Média	Mínimo	Desvio padrão
MLP	0,7741935484	0,6841019122	0,6206896552	0,0497804281
KNN	0,5433333333	0,5283333333	0,5133333333	0,0097182532
SVM	0,5654545455	0,5414545455	0,4854545455	0,0211869981
AD	0,5463157895	0,5263157895	0,5063157895	0,0105409255

KNN e Árvore de Decisão, apresentaram desempenho semelhante.

Figura 17 – Coeficiente kappa dos modelos



MLP representa o desempenho da rede neural, SVM representa o desempenho da Máquina com Suporte Vetorial, KNN representa o algoritmo dos K vizinhos mais próximos e AD representa o desempenho da Árvore de Decisão.
Fonte: Próprio Autor.

Na Tabela 11, estão dispostos os resultados do experimento para a métrica coeficiente kappa.

Tabela 11 – Resultados do coeficiente kappa do experimento

Algoritmo	Máximo	Média	Mínimo	Desvio padrão
MLP	0,4615384615	0,2769230769	0,1538461538	0,1099877466
KNN	0	-0,0779230769	-0,0969230769	0,0099442893
SVM	0,3307692308	0,2317692308	0,1807692308	0,0387154864
AD	0,3576923077	0,3056923077	0,2076923077	0,00144

O coeficiente kappa é mostrado na Figura 17. Nesta figura e na Tabela 11 é possível observar que, em média, houve pouca diferença entre o desempenho dos modelos MLP, SVM e Árvore de decisão. Entretanto, o coeficiente kappa da melhor execução da rede neural MLP apresentou resultado bem superior aos demais com kappa igual a 0,46. O algoritmo KNN apresentou desempenho insatisfatório quanto ao coeficiente kappa, com todos os valores abaixo de 0, por isso não aparece na Figura 17.

Na Tabela 12, tem-se o resultado do teste estatístico de análise de variância (ANOVA) para checar se existe diferença entre os desempenhos dos algoritmos. Para as análises estatísticas, foi usada a linguagem de programação R. Em todos os testes estatísticos, o valor-p $< 0,05$ foi considerado como estatisticamente relevante.

Tabela 12 – Teste ANOVA dos modelos de classificação do experimento final

Métrica	Df	Df.res	Sum Sq	Sum Sq.res	Mean Sq	valor-p
Acurácia	3	36	0,233	0,033	0,077	< 0.001
Medida F	3	36	0,174	0,028	0,058	< 0.001
Coeficiente kappa	3	36	0,943	0,136	0,314	< 0.001

Na Tabela 12, todos os testes apresentaram valor-p menor que 0,01. Isto sinaliza que há uma diferença estatística entre o desempenho dos algoritmos no geral. Fez-se também um teste post hoc para verificar a diferença estatística entre os pares de algoritmos. O teste utilizado foi o "Tukey com múltiplas comparações entre médias". O resultado do valor-p para todas as métricas e modelos de classificação está disponível na Tabela 13.

Tabela 13 – Test post hoc - Tukey de múltiplas comparações entre médias

Algoritmos	p (Acurácia)	p (Medida F)	p (Kappa)
KNN-AD	0,0000000	0,9984802	0,0000000
MLP-AD	0,9274347	0,0000000	0,7239393
SVM-AD	0,2960432	0,6248643	0,0506931
MLP-KNN	0,0000000	0,0000000	0,0000000
SVM-KNN	0,0000000	0,7224017	0,0000000
SVM-MLP	0,6465444	0,0000000	0,3690339

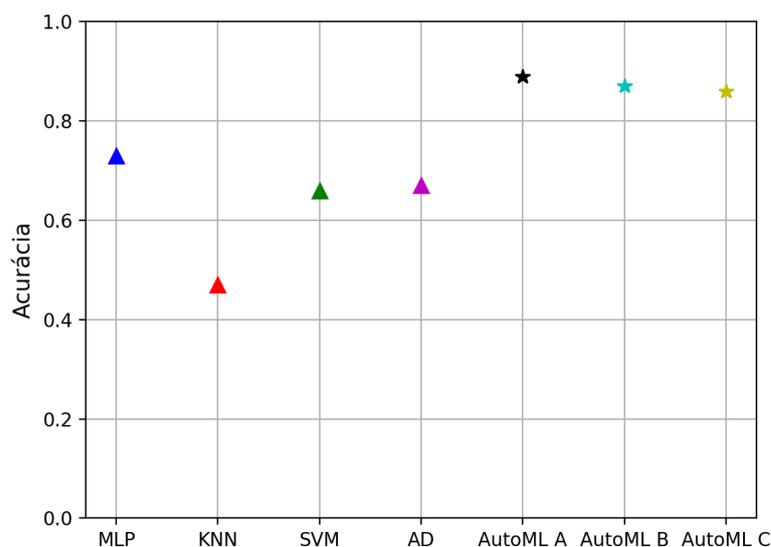
Na Tabela 13, nota-se algumas das combinações não houve diferença estatística relevante (valor-p $> 0,05$). Não houve diferença estatística entre a acurácia e coeficiente kappa nos pares MLP-Árvore de decisão e SVM-MLP. Não houve diferença estatística para nenhuma das métricas na dupla SVM e Árvore de decisão. Não houve diferença estatística entre a medida F nos pares SVM-KNN e KNN-Árvore de decisão.

Para as três métricas, a rede neural MLP, SVM e árvore de decisão apresentaram resultados resultados bem próximos e sem diferença estatisticamente relevante. Porém a acurácia ainda está abaixo do esperado quando comparado com os trabalhos [Sato et al. \(2016\)](#) e [Lin et al. \(2018\)](#). O coeficiente kappa da melhor execução da rede neural MLP apresentou nível de concordância moderado, que segundo [Altman \(1999\)](#), encontra-se entre 0.40 e 0.60. Resultados melhores poderão ocorrer caso o número de amostras aumente.

4.2.3 Aplicação de AutoML no Aprendizado Motor

Os três melhores resultados da aplicação de AutoML do Auto Weka são mostrados nas Figuras 18, 19 e 20. Nestas figuras também podem-se ver os melhores resultados das abordagens tradicionais para efeito de comparação com a aplicação de AutoML. Os resultados gerais dos modelos selecionados por AutoML estão dispostos nos Apêndices A, B e C.

Figura 18 – Acurácia das abordagens tradicionais e AutoML



Os triângulos são os melhores resultados das abordagens tradicionais. As estrelas são os resultados médios dos melhores resultados da aplicação de AutoML. AutoML A: resultado do Classificador A; AutoML B: resultado do Classificador B; AutoML C: resultado do Classificador C.

Fonte: Próprio Autor.

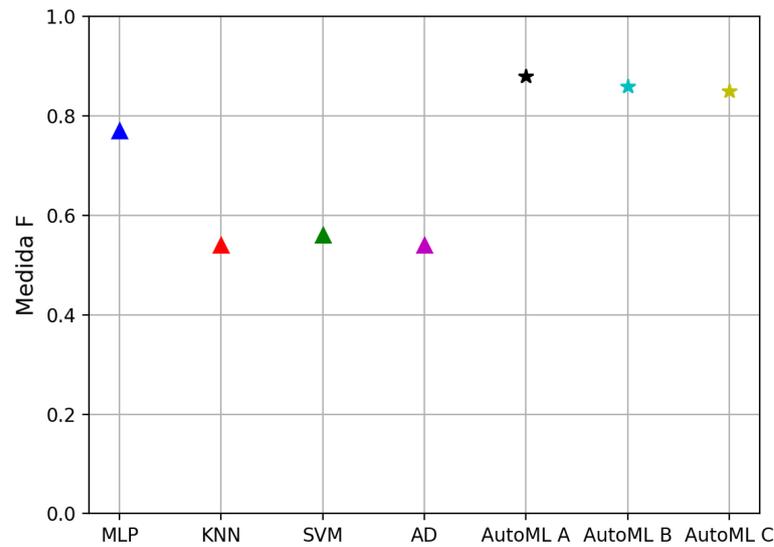
Através das Figuras 18, 19 e 20 percebe-se uma superioridade dos modelos selecionados pela aplicação de AutoML em comparação com os modelos implementados para as três métricas (Acurácia, Medida F e Coeficiente Kappa). Na Tabela 14, estão dispostos os resultados dos modelos.

Tabela 14 – Resultados da aplicação de AutoML e metodologias tradicionais

Modelo	Acurácia	Medida F	Kappa
AutoML A	0,89	0,88	0,78
AutoML B	0,86	0,86	0,73
AutoML C	0,85	0,85	0,71
MLP	0,73	0,77	0,46
KNN	0,47	0,54	0
SVM	0,65	0,56	0,33
AD	0,67	0,54	0,35

O desempenho geral dos modelos selecionados por AutoML foi superior para as três métricas apresentadas. O resultado é altamente relevante quando comparado com a

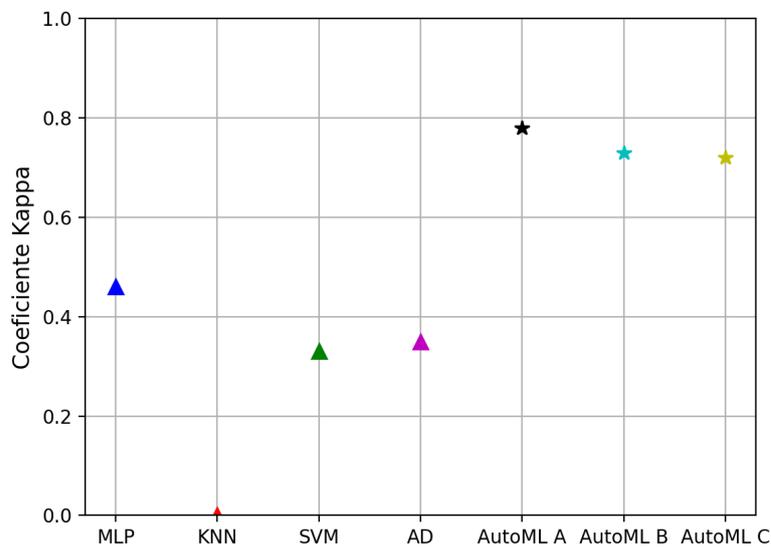
Figura 19 – Medida F das abordagens tradicionais e AutoML



Os triângulos são os melhores resultados das abordagens tradicionais. As estrelas são os resultados médios dos melhores resultados da aplicação de AutoML. AutoML A: resultado do Classificador A; AutoML B: resultado do Classificador B; AutoML C: resultado do Classificador C.

Fonte: Próprio Autor.

Figura 20 – Coeficiente Kappa das abordagens tradicionais e AutoML



Os triângulos são os melhores resultados das abordagens tradicionais. As estrelas são os resultados médios dos melhores resultados da aplicação de AutoML. AutoML A: resultado do Classificador A; AutoML B: resultado do Classificador B; AutoML C: resultado do Classificador C.

Fonte: Próprio Autor.

literatura, referenciada por [Sonoda et al. \(2005\)](#), [Sato et al. \(2016\)](#) e [Lin et al. \(2018\)](#). Outro resultado relevante é relacionado ao coeficiente kappa. Para os três classificadores, o coeficiente kappa ficou entre 0,6 e 0,8, o que, segundo [Altman \(1999\)](#) representa uma boa concordância na estimação de valores.

5 Conclusão

O objetivo principal deste trabalho foi propor uma abordagem para estimar a quantidade de sessões necessárias para aprender uma tarefa motora por meio de algoritmos de aprendizado de máquina.

Durante esta pesquisa foram elaborados dois experimentos, um de prova de conceito e outro final. O objetivo do primeiro experimento foi observar características do aprendizado motor em uma amostragem pequena dos dados coletados de três participantes e gerar dados simulados para verificar como os algoritmos de classificação se comportariam em termos de desempenho com uma amostragem maior. O objetivo do experimento final foi observar características do aprendizado motor, em uma amostragem maior, e utilizar apenas os dados coletados de oito participantes para prever a quantidade de sessões necessárias para atingir determinado desempenho, objetivo principal deste trabalho.

Nos dois experimentos, foi possível perceber a característica do aperfeiçoamento motor. No primeiro, mostrou-se o aperfeiçoamento entre blocos e no segundo entre sessões. No segundo experimento, conseguiu-se também observar a consistência motora, uma importante característica do aprendizado motor. Estes resultados corroboram com as teorias acerca destas duas características presentes em [Kitago e Krakauer \(2013\)](#), [Reis et al. \(2009\)](#) e [Dayan e Cohen \(2011\)](#).

A partir da geração de novas amostras através da adição de ruídos gaussianos com média zero e desvio padrão fixo foi concebida uma base de dados 18 participantes a partir dos três participantes do experimento piloto. Através desses dados simulados foi possível aplicar técnicas de classificação com o objetivo de prever em qual bloco os participantes se encontravam de acordo com seu desempenho em termos de tempo de movimento e erro percentual de execução. Neste experimento foram utilizados os algoritmos de classificação: KNN, SVM, rede neural MLP e árvore de decisão. Através dos dados do experimento piloto obteve-se os melhores resultados utilizando o algoritmo Árvore de Decisão com acurácia média de 97%.

Após a coleta de dados do experimento final, foi possível aplicar os algoritmos de classificação diretamente aos dados dos oito participantes. Os modelos de classificação utilizados no experimento final foram: KNN, rede neural MLP, árvore de decisão, SVM e aprendizagem de máquina automática (AutoML) com "Auto Weka". Utilizando os dados do experimento final, oito participantes, obtiveram-se os melhores resultados com o AutoML, atingindo uma acurácia de 89%. Destaca-se também que os resultados obtidos pela abordagem AutoML possuem coeficiente kappa entre 0,60 e 0,80, que indica uma boa concordância na predição. Segundo [Altman \(1999\)](#) isto é um fator essencial ao suporte

à decisão clínica. Os resultados também apresentam desempenho superior a trabalhos relacionados ([SONODA et al., 2005](#); [SATO et al., 2016](#); [LIN et al., 2018](#)).

Como trabalhos futuros sugere-se fazer o experimento com a quantidade igual ou maior que 18 participantes saudáveis. Recomenda-se testar mais técnicas e compará-las ao desempenho dos classificadores utilizados neste trabalho. Propõe-se também, aplicar este experimento a pacientes de doenças neurológicas que sofreram dano motor, porém não perderam a capacidade de aprendizagem motora. Além disso, esta técnica também pode vir a ser utilizada para a predição de progresso da coordenação motora para fins pedagógicos.

Referências

- ALTMAN, D. G. *Practical statistics for medical research*. [S.l.]: CRC press, 1999. Citado 4 vezes nas páginas 36, 46, 48 e 49.
- BERGSTRA, J.; BENGIO, Y. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, v. 13, n. Feb, p. 281–305, 2012. Citado na página 27.
- CARVALHO, A.; FACELI, K.; LORENA, A.; GAMA, J. *Inteligência Artificial—uma abordagem de aprendizado de máquina*. [S.l.]: Rio de Janeiro: LTC, 2011. Citado 5 vezes nas páginas 21, 23, 25, 26 e 36.
- CARVALHO, M. B. F. de. *Plataforma de Avaliação do Aprendizado de Habilidade Motora Automatizada*. Monografia (Graduação) — Universidade Federal do Maranhão, São Luís, 2018. Citado 3 vezes nas páginas 28, 29 e 30.
- CHICCO, D. Ten quick tips for machine learning in computational biology. *BioData mining*, BioMed Central, v. 10, n. 1, p. 35, 2017. Citado na página 26.
- DAYAN, E.; COHEN, L. G. Neuroplasticity subserving motor skill learning. *Neuron*, Elsevier Science, v. 72, 2011. Citado 3 vezes nas páginas 18, 42 e 49.
- ESPÍRITO-SANTO, H.; PIRES, C. F.; GARCIA, I. Q.; DANIEL, F.; SILVA, A. G. d.; FAZIO, R. L. Preliminary validation of the portuguese edinburgh handedness inventory in an adult sample. *Applied Neuropsychology: Adult*, Taylor & Francis, v. 24, n. 3, p. 275–287, 2017. Citado 2 vezes nas páginas 28 e 29.
- FORKERT, N. D.; VERLEGER, T.; CHENG, B.; THOMALLA, G.; HILGETAG, C. C.; FIEHLER, J. Multiclass support vector machine-based lesion mapping predicts functional outcome in ischemic stroke patients. *PloS one*, Public Library of Science, v. 10, n. 6, p. e0129569, 2015. Citado na página 16.
- FRANK, E.; HALL, M.; WITTEN, I. The weka workbench. *Data mining: Practical machine learning tools and techniques*, Morgan Kaufmann MA, US, v. 4, 2016. Citado na página 37.
- GONZALEZ, R. C.; WOODS, R. E. *Processamento de imagens digitais*. 3. ed. [S.l.]: Pearson Education, 2010. Citado na página 21.
- HAHNE, J. M.; BIESSMANN, F.; JIANG, N.; REHBAUM, H.; FARINA, D.; MEINECKE, F.; MÜLLER, K.-R.; PARRA, L. Linear and nonlinear regression techniques for simultaneous and proportional myoelectric control. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, IEEE, v. 22, n. 2, p. 269–279, 2014. Citado 2 vezes nas páginas 15 e 35.
- HALLETT, M.; GRAFMAN, J. Executive function and motor skill learning. *International review of neurobiology*, Elsevier, v. 41, p. 297–323, 1997. Citado na página 18.
- HAYKIN, S. *Neural Networks: A comprehensive foundation*. [S.l.]: Bookman, 2004. v. 2. 41 p. Citado na página 24.

- HONG, W.-C. Chaotic particle swarm optimization algorithm in a support vector regression electric load forecasting model. *Energy Conversion and Management*, Elsevier, v. 50, n. 1, p. 105–117, 2009. Citado na página 27.
- HSUEH, P.; LIN, J.-H.; JENG, J.-S.; HSIEH, C.-L. Comparison of the psychometric characteristics of the functional independence measure, 5 item barthel index, and 10 item barthel index in patients with stroke. *Journal of Neurology, Neurosurgery & Psychiatry*, BMJ Publishing Group Ltd, v. 73, n. 2, p. 188–190, 2002. Citado 2 vezes nas páginas 14 e 17.
- JORDAN, M. I.; MITCHELL, T. M. Machine learning: Trends, perspectives, and prospects. *Science*, American Association for the Advancement of Science, v. 349, n. 6245, p. 255–260, 2015. Citado 2 vezes nas páginas 14 e 15.
- JULIANJATSONO, R.; FERDIANA, R.; HARTANTO, R. High-resolution automated fughl-meyer assessment using sensor data and regression model. In: IEEE. *3rd International Conference on Science and Technology-Computer (ICST), 2017*. [S.l.], 2017. p. 28–32. Citado na página 16.
- KARNI, A.; MEYER, G.; REY-HIPOLITO, C.; JEZZARD, P.; ADAMS, M. M.; TURNER, R.; UNGERLEIDER, L. G. The acquisition of skilled motor performance: fast and slow experience-driven changes in primary motor cortex. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 95, n. 3, p. 861–868, 1998. Citado na página 18.
- KITAGO, T.; KRAKAUER, J. W. Motor learning principles for neurorehabilitation. *Handb Clin Neurol*, v. 110, p. 93–103, 2013. Citado 6 vezes nas páginas 14, 18, 19, 38, 42 e 49.
- KORMAN, M.; RAZ, N.; FLASH, T.; KARNI, A. Multiple shifts in the representation of a motor sequence during the acquisition of skilled performance. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 100, n. 21, p. 12492–12497, 2003. Citado na página 19.
- KOTTHOFF, L.; THORNTON, C.; HOOS, H. H.; HUTTER, F.; LEYTON-BROWN, K. Auto-weka 2.0: Automatic model selection and hyperparameter optimization in weka. *The Journal of Machine Learning Research*, JMLR. org, v. 18, n. 1, p. 826–830, 2017. Citado 2 vezes nas páginas 27 e 37.
- KRAKAUER, J. W. Motor learning: its relevance to stroke recovery and neurorehabilitation. *Current Opinion in Neurology*, Lippincott Williams and Wilkins, v. 19, 02 2006. Citado na página 19.
- LEMM, S.; BLANKERTZ, B.; DICKHAUS, T.; MÜLLER, K.-R. Introduction to machine learning for brain imaging. *Neuroimage*, Elsevier, v. 56, n. 2, p. 387–399, 2011. Citado na página 35.
- LI, M.; SUOHAI, F. Forex prediction based on svr optimized by artificial fish swarm algorithm. In: *2013 Fourth Global Congress on Intelligent Systems*. [S.l.: s.n.], 2013. p. 47–52. ISSN 2155-6083. Citado na página 27.

- LIN, W.-Y.; CHEN, C.-H.; TSENG, Y.-J.; TSAI, Y.-T.; CHANG, C.-Y.; WANG, H.-Y.; CHEN, C.-K. Predicting post-stroke activities of daily living through a machine learning-based approach on initiating rehabilitation. *International Journal of Medical Informatics*, Elsevier, 2018. Citado 7 vezes nas páginas 14, 16, 17, 41, 46, 48 e 50.
- MARSLAND, S. *Machine learning: an algorithmic perspective*. [S.l.]: CRC press, 2015. Citado 4 vezes nas páginas 21, 22, 24 e 25.
- MITCHELL, T. M. *Machine learning*. [S.l.]: Burr Ridge, IL: McGraw Hill, 1997. v. 45. Citado na página 20.
- NORVIG, P.; RUSSELL, S. *Inteligência Artificial*. 3. ed. [S.l.]: Elsevier Brasil, 2014. v. 1. Citado na página 35.
- PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Citado na página 33.
- PRICHARD, G.; WEILLER, C.; FRITSCH, B.; REIS, J. Effects of different electrical brain stimulation protocols on subcomponents of motor skill learning. *Brain stimulation*, Elsevier, v. 7, n. 4, p. 532–540, 2014. Citado 6 vezes nas páginas 14, 15, 19, 29, 30 e 31.
- REIS, J.; SCHAMBRA, H. M.; COHEN, L. G.; BUCH, E. R.; FRITSCH, B.; ZARAHN, E.; CELNIK, P. A.; KRAKAUER, J. W. Noninvasive cortical stimulation enhances motor skill acquisition over multiple days through an effect on consolidation. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 106, n. 5, p. 1590–1595, 2009. Citado 4 vezes nas páginas 19, 29, 38 e 49.
- ROSSUM, G. *Python Reference Manual*. Amsterdam, The Netherlands, The Netherlands, 1995. Citado 2 vezes nas páginas 29 e 33.
- SALE, P.; FERRIERO, G.; CIABATTONI, L.; CORTESE, A. M.; FERRACUTI, F.; ROMEO, L.; PICCIONE, F.; MASIERO, S. Predicting motor and cognitive improvement through machine learning algorithm in human subject that underwent a rehabilitation treatment in the early stage of stroke. *Journal of Stroke and Cerebrovascular Diseases*, Elsevier, 2018. Citado na página 16.
- SATO, A.; FUJITA, T.; OHASHI, Y.; YAMAMOTO, Y.; SUZUKI, K.; OTSUKI, K. A prediction model for activities of daily living for stroke patients in a convalescent rehabilitation ward. *Journal of allied health sciences*, Society of Allied Health Sciences, v. 7, n. 1, p. 1–6, 2016. Citado 4 vezes nas páginas 16, 46, 48 e 50.
- SHINNERS, P. *PyGame*. 2011. <<http://pygame.org/>>. Citado na página 29.
- SHMUELOF, L.; KRAKAUER, J. W. Recent insights into perceptual and motor skill learning. *Frontiers in human neuroscience*, Frontiers Media SA, v. 8, 2014. Citado na página 18.
- SONODA, S.; SAITOH, E.; NAGAI, S.; OKUYAMA, Y.; SUZUKI, T.; SUZUKI, M. Stroke outcome prediction using reciprocal number of initial activities of daily living status. *Journal of Stroke and Cerebrovascular Diseases*, Elsevier, v. 14, n. 1, p. 8–11, 2005. Citado 3 vezes nas páginas 15, 48 e 50.

- SPRINT, G.; COOK, D. J.; WEEKS, D. L.; BORISOV, V. Predicting functional independence measure scores during rehabilitation with wearable inertial sensors. *IEEE Access*, IEEE, v. 3, p. 1350–1366, 2015. Citado na página 16.
- THORNTON, C.; HUTTER, F.; HOOS, H. H.; LEYTON-BROWN, K. Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In: ACM. *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.], 2013. p. 847–855. Citado na página 37.
- TSUJI, T.; LIU, M.; SONODA, S.; DOMEN, K.; CHINO, N. The stroke impairment assessment set: its internal consistency and predictive validity. *Archives of physical medicine and rehabilitation*, Elsevier, v. 81, n. 7, p. 863–868, 2000. Citado na página 15.
- VAPNIK, V. *Statistical Learning Theory*. [S.l.]: NY: Wiley, 1998. Citado 2 vezes nas páginas 22 e 23.
- ZHANG, L.; MISTRY, K.; LIM, C. P.; NEOH, S. C. Feature selection using firefly optimization for classification and regression models. *Decision Support Systems*, Elsevier, v. 106, p. 64–85, 2018. Citado na página 27.

Apêndices

APÊNDICE A – Classificador A

Figura 21 – Buffer do Auto Weka - classificador A

Arquivo: /home/turing/Documentos/Auto weka runs/res17 BOM Página 1 de 1

```
Auto-WEKA result:
best classifier: weka.classifiers.meta.Bagging
arguments: [-P, 65, -I, 9, -S, 1, -W, weka.classifiers.rules.OneR, --, -B, 1]
attribute search: weka.attributeSelection.GreedyStepwise
attribute search arguments: [-R]
attribute evaluation: weka.attributeSelection.CfsSubsetEval
attribute evaluation arguments: [-M, -L]
metric: weightedFMeasure
estimated weightedFMeasure: 0.9140152671755726
training time on evaluation dataset: 0.005 seconds

You can use the chosen classifier in your own code as follows:

AttributeSelection as = new AttributeSelection();
ASearch asSearch = ASearch.forName("weka.attributeSelection.GreedyStepwise", new
String[]{"-R"});
as.setSearch(asSearch);
ASEvaluation asEval = ASEvaluation.forName
("weka.attributeSelection.CfsSubsetEval", new String[]{"-M", "-L"});
as.setEvaluator(asEval);
as.SelectAttributes(instances);
instances = as.reduceDimensionality(instances);
Classifier classifier = AbstractClassifier.forName
("weka.classifiers.meta.Bagging", new String[]{"-P", "65", "-I", "9", "-S", "1", "-
W", "weka.classifiers.rules.OneR", "--", "-B", "1"});
classifier.buildClassifier(instances);

Correctly Classified Instances          114           89.0625 %
Incorrectly Classified Instances        14           10.9375 %
Kappa statistic                        0.7813
Mean absolute error                    0.2292
Root mean squared error                 0.3005
Relative absolute error                 45.8333 %
Root relative squared error            60.0925 %
Total Number of Instances              128

=== Confusion Matrix ===
   a  b  <-- classified as
55  9  | a = 2
 5 59 | b = 3

=== Detailed Accuracy By Class ===
Area  PRC Area  TP Rate  FP Rate  Precision  Recall  F-Measure  MCC    ROC
0,963  0,958    0,859   0,078   0,917     0,859   0,887     0,783
          2
0,963  0,952    0,922   0,141   0,868     0,922   0,894     0,783
          3
Weighted Avg. 0,891   0,109   0,892     0,891   0,891     0,783
0,963  0,955
```

Temporary run directories:
/tmp/autoweka6564005790405308318/

For better performance, try giving Auto-WEKA more time.

APÊNDICE B – Classificador B

Figura 22 – Buffer do Auto Weka - classificador B

Arquivo: /home/turing/Documentos/Auto weka runs/res1 Página 1 de 2

```

Auto-WEKA result:
best classifier: weka.classifiers.rules.OneR
arguments: [-B, 2]
attribute search: weka.attributeSelection.GreedyStepwise
attribute search arguments: [-C, -B, -R]
attribute evaluation: weka.attributeSelection.CfsSubsetEval
attribute evaluation arguments: [-M, -L]
metric: errorRate
estimated errorRate: 0.1328125
training time on evaluation dataset: 0.004 seconds

You can use the chosen classifier in your own code as follows:

AttributeSelection as = new AttributeSelection();
ASearch asSearch = ASearch.forName("weka.attributeSelection.GreedyStepwise", new
String[]{"-C", "-B", "-R"});
as.setSearch(asSearch);
ASEvaluation asEval = ASEvaluation.forName
("weka.attributeSelection.CfsSubsetEval", new String[]{"-M", "-L"});
as.setEvaluator(asEval);
as.SelectAttributes(instances);
instances = as.reduceDimensionality(instances);
Classifier classifier = AbstractClassifier.forName("weka.classifiers.rules.OneR",
new String[]{"-B", "2"});
classifier.buildClassifier(instances);

Correctly Classified Instances      111          86.7188 %
Incorrectly Classified Instances    17           13.2813 %
Kappa statistic                    0.7344
Mean absolute error                 0.1328
Root mean squared error             0.3644
Relative absolute error             26.5625 %
Root relative squared error        72.8869 %
Total Number of Instances          128

=== Confusion Matrix ===
   a  b  <-- classified as
56  8  |  a = 2
 9 55  |  b = 3

=== Detailed Accuracy By Class ===
Area  PRC Area      TP Rate  FP Rate  Precision  Recall   F-Measure  MCC     ROC
Class
0,867  0,816  0,875  0,141  0,862     0,875   0,868     0,734
2
0,867  0,821  0,859  0,125  0,873     0,859   0,866     0,734
3
Weighted Avg. 0,867  0,133  0,867     0,867   0,867     0,734
0,867  0,818

----- 2 BEST CONFIGURATIONS -----

These are the 2 best configurations, as ranked by SMAC
Please note that this list only contains configurations evaluated on at least 10
folds,
If you need more configurations, please consider running Auto-WEKA for a longer
time.

Configuration #1:
SMAC Score: 0.40576923076923077
Argument String:
-_0_weka.classifiers.rules.one_r_00_INT_B 2 -

```

APÊNDICE C – Classificador C

Figura 23 – Buffer do Auto Weka - classificador C

Arquivo: /home/turing/Documentos/Auto weka runs/res20 Página 1 de 1

```
Auto-WEKA result:
best classifier: weka.classifiers.meta.Bagging
arguments: [-P, 52, -I, 9, -S, 1, -W, weka.classifiers.rules.OneR, --, -B, 1]
attribute search: weka.attributeSelection.GreedyStepwise
attribute search arguments: [-C, -B, -R]
attribute evaluation: weka.attributeSelection.CfsSubsetEval
attribute evaluation arguments: [-M, -L]
metric: errorRate
estimated errorRate: 0.140625
training time on evaluation dataset: 0.007 seconds

You can use the chosen classifier in your own code as follows:

AttributeSelection as = new AttributeSelection();
ASearch asSearch = ASearch.forName("weka.attributeSelection.GreedyStepwise", new
String[]{"-C", "-B", "-R"});
as.setSearch(asSearch);
ASEvaluation asEval = ASEvaluation.forName
("weka.attributeSelection.CfsSubsetEval", new String[]{"-M", "-L"});
as.setEvaluator(asEval);
as.SelectAttributes(instances);
instances = as.reduceDimensionality(instances);
Classifier classifier = AbstractClassifier.forName
("weka.classifiers.meta.Bagging", new String[]{"-P", "52", "-I", "9", "-S", "1", "-
W", "weka.classifiers.rules.OneR", "--", "-B", "1"});
classifier.buildClassifier(instances);

Correctly Classified Instances          110           85.9375 %
Incorrectly Classified Instances        18           14.0625 %
Kappa statistic                        0.7188
Mean absolute error                    0.2622
Root mean squared error                 0.3316
Relative absolute error                 52.4306 %
Root relative squared error            66.3185 %
Total Number of Instances              128

=== Confusion Matrix ===
   a  b  <-- classified as
54 10 | a = 2
 8 56 | b = 3

=== Detailed Accuracy By Class ===
Area  PRC Area  TP Rate  FP Rate  Precision  Recall  F-Measure  MCC  ROC
Class
0,844  0,125  0,871  0,844  0,857  0,719
2
0,939  0,931  0,875  0,156  0,848  0,875  0,862  0,719
3
0,939  0,924  0,859  0,141  0,860  0,859  0,859  0,719
Weighted Avg.
0,939  0,927
```

Temporary run directories:
/tmp/autoweka7264352767004115046/

For better performance, try giving Auto-WEKA more time.

Anexos

ANEXO A – Edinburgh Inventory

Edinburgh Handedness Inventory

(Versão original: Oldfield, 1971)

(Adaptação portuguesa: Espírito Santo et al., 2017)*

Indique qual das mãos usa preferencialmente na execução das atividades que lhe vão ser apresentadas de seguida.

Para tal, coloque um "+" na coluna que corresponde à **mão que usa preferencialmente** na execução dessas atividades.

Quando **a sua preferência for tão forte** que nunca usa a outra mão, a não ser que seja forçado/a, marque "++".

Se o uso de uma ou de outra **mão for indiferente**, marque "+" **nas duas colunas**.

Algumas atividades exigem o uso de ambas as mãos. Nesses casos, o objeto para o qual deve considerar o uso preferencial da mão é indicado entre parêntesis.

Por favor responda a todas as questões.

Atividades	Mão	
	Esquerda	Direita
Escrever		
Desenhar		
Atirar/Lançar		
Usar a tesoura		
Segurar a escova de dentes		
Cortar com uma faca		
Usar uma colher		
Varrer (cimo da vassoura)		
Segurar num fósforo para o acender/riscar (fósforo)		
Segurar na tampa para abrir uma caixa (tampa)		

***Referência Provisória:** Espírito-Santo, H., Pires, A. C., Queiroz Garcia, I., Daniel, F., Silva, A., & Fazio, R. (2017). Preliminary validation of the Portuguese Edinburgh Handedness Inventory in an adult sample. *Applied Neuropsychology: Adult*, XX(X), XX-XX. doi: XXX

ANEXO B – Beck Depression Inventory



INVENTÁRIO DE DE PRESSÃO DE BECK - BDI

Nome: _____ Idade: _____ Data: ____ / ____ / ____

Este questionário consiste em 21 grupos de afirmações. Depois de ler cuidadosamente cada grupo, faça um círculo em torno do número (0, 1, 2 ou 3) próximo à afirmação, em cada grupo, que descreve **melhor** a maneira que você tem se sentido na **última semana, incluindo hoje**. Se várias afirmações num grupo parecerem se aplicar igualmente bem, faça um círculo em cada uma. **Tome cuidado de ler todas as afirmações, em cada grupo, antes de fazer sua escolha.**

1	0 Não me sinto triste 1 Eu me sinto triste 2 Estou sempre triste e não consigo sair disto 3 Estou tão triste ou infeliz que não consigo suportar	7	0 Não me sinto decepcionado comigo mesmo 1 Estou decepcionado comigo mesmo 2 Estou enojado de mim 3 Eu me odeio
2	0 Não estou especialmente desanimado quanto ao futuro 1 Eu me sinto desanimado quanto ao futuro 2 Acho que nada tenho a esperar 3 Acho o futuro sem esperanças e tenho a impressão de que as coisas não podem melhorar	8	0 Não me sinto de qualquer modo pior que os outros 1 Sou crítico em relação a mim por minhas fraquezas ou erros 2 Eu me culpo sempre por minhas falhas 3 Eu me culpo por tudo de mal que acontece
3	0 Não me sinto um fracasso 1 Acho que fracassei mais do que uma pessoa comum 2 Quando olho pra trás, na minha vida, tudo o que posso ver é um monte de fracassos 3 Acho que, como pessoa, sou um completo fracasso	9	0 Não tenho quaisquer idéias de me matar 1 Tenho idéias de me matar, mas não as executaria 2 Gostaria de me matar 3 Eu me mataria se tivesse oportunidade
4	0 Tenho tanto prazer em tudo como antes 1 Não sinto mais prazer nas coisas como antes 2 Não encontro um prazer real em mais nada 3 Estou insatisfeito ou aborrecido com tudo	10	0 Não choro mais que o habitual 1 Choro mais agora do que costumava 2 Agora, choro o tempo todo 3 Costumava ser capaz de chorar, mas agora não consigo, mesmo que o queria
5	0 Não me sinto especialmente culpado 1 Eu me sinto culpado grande parte do tempo 2 Eu me sinto culpado na maior parte do tempo 3 Eu me sinto sempre culpado	11	0 Não sou mais irritado agora do que já fui 1 Fico aborrecido ou irritado mais facilmente do que costumava 2 Agora, eu me sinto irritado o tempo todo 3 Não me irrita mais com coisas que costumavam me irritar
6	0 Não acho que esteja sendo punido 1 Acho que posso ser punido 2 Creio que vou ser punido 3 Acho que estou sendo punido	12	0 Não perdi o interesse pelas outras pessoas 1 Estou menos interessado pelas outras pessoas do que costumava estar 2 Perdi a maior parte do meu interesse pelas outras pessoas 3 Perdi todo o interesse pelas outras pessoas



INVENTÁRIO DE DE PRESSÃO DE BECK - BDI

13	0 Tomo decisões tão bem quanto antes 1 Adio as tomadas de decisões mais do que costumava 2 Tenho mais dificuldades de tomar decisões do que antes 3 Absolutamente não consigo mais tomar decisões	18	0 O meu apetite não está pior do que o habitual 1 Meu apetite não é tão bom como costumava ser 2 Meu apetite é muito pior agora 3 Absolutamente não tenho mais apetite
14	0 Não acho que de qualquer modo pareço pior do que antes 1 Estou preocupado em estar parecendo velho ou sem atrativo 2 Acho que há mudanças permanentes na minha aparência, que me fazem parecer sem atrativo 3 Acredito que pareço feio	19	0 Não tenho perdido muito peso se é que perdi algum recentemente 1 Perdi mais do que 2 quilos e meio 2 Perdi mais do que 5 quilos 3 Perdi mais do que 7 quilos Estou tentando perder peso de propósito, comendo menos: Sim _____ Não _____
15	0 Posso trabalhar tão bem quanto antes 1 É preciso algum esforço extra para fazer alguma coisa 2 Tenho que me esforçar muito para fazer alguma coisa 3 Não consigo mais fazer qualquer trabalho	20	0 Não estou mais preocupado com a minha saúde do que o habitual 1 Estou preocupado com problemas físicos, tais como dores, indisposição do estômago ou constipação 2 Estou muito preocupado com problemas físicos e é difícil pensar em outra coisa 3 Estou tão preocupado com meus problemas físicos que não consigo pensar em qualquer outra coisa
16	0 Consigo dormir tão bem como o habitual 1 Não durmo tão bem como costumava 2 Acordo 1 a 2 horas mais cedo do que habitualmente e acho difícil voltar a dormir 3 Acordo várias horas mais cedo do que costumava e não consigo voltar a dormir	21	0 Não notei qualquer mudança recente no meu interesse por sexo 1 Estou menos interessado por sexo do que costumava 2 Estou muito menos interessado por sexo agora 3 Perdi completamente o interesse por sexo
17	0 Não fico mais cansado do que o habitual 1 Fico cansado mais facilmente do que costumava 2 Fico cansado em fazer qualquer coisa 3 Estou cansado demais para fazer qualquer coisa		