UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

José Rodrigues de Oliveira Neto

Uma abordagem de Game Learning Analytics para identificação de habilidades de leitura e escrita no ensino infantil.

Recife

2019

José Rodrigues de Oliveira Neto

Uma abordagem de Game Learning Analytics para identificação de habilidades de leitura e escrita no ensino infantil.

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Universidade Federal Rural de Pernambuco – UFRPE

Departamento de Estatística e Informática

Curso de Bacharelado em Sistemas de Informação

Orientador: Rodrigo Lins Rodrigues

Coorientador: Américo Nobre Amorim

Recife

2019

À Deus, minha famíla e amigos.

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema Integrado de Bibliotecas da UFRPE
Biblioteca Central, Recife-PE, Brasil

O48u Oliveira Neto, José Rodrigues de
Uma abordagem de Game Learning Analytics para identificação de habilidades de leitura e escrita no ensino infantil / José Rodrigues de Oliveira Neto. – Recife, 2018.
69 f.: il.

Orientador: Rodrigo Lins Rodrigues.
Coorientador: Américo Nobre Gonçalves Ferreira Amorim.
Trabalho de Conclusão de Curso (Graduação em Sistemas de informação) – Universidade Federal Rural de Pernambuco, Departamento de Estatística e Informática, Recife, BR-PE, 2019.
Inclui referências, anexo(s) e apêndice(s).

1. Ensino auxiliado por computador 2. Aprendizagem - Jogos para computador 3. Mineração de dados (Computação) 4. Leitura - Estudo e ensino 5. Incentivo à leitura I. Rodrigues, Rodrigo Lins, orient. II. Amorim, Américo Nobre Gonçalves Ferreira, coorient. III. Título

CDD 004.03

Agradecimentos

Agradeço à minha família, em especial os meus pais, Rejane e Jurandir, que sempre fizeram o possível e o impossível para que eu pudesse ter as condições de ser o que sou hoje. Agradeço também os meus irmãos, Bárbara e Vinícius, por todo apoio que me dão em todos os momentos da minha vida. Sem vocês, nada do que faço faria algum sentido.

À minha namorada, Tatiane Lima, por estar do meu lado nessa maravilhosa jornada chamada vida, além de todo o apoio e companheirismo fundamentais para que eu pudesse fechar esse importante ciclo de graduação.

A todos os meus amigos e companheiros que fiz aqui na Rural, especialmente a François, Igor, Matheus, Raissa, Ewerton, Delando, Diego e Ricardo. Vocês fizeram com que a Universidade fosse uma experiência ainda mais divertida e enriquecedora para mim.

Agradeço aos meus amigos de longa data, especialmente a Felipe Aurélio e Juliana Ferraz, por estarem sempre me apoiando, torcendo pelo meu sucesso e por acreditarem no meu potencial, até quando eu mesmo não acreditava.

A todos os professores que fazem o Bacharelado em Sistemas de Informação da UFRPE acontecer. Vocês são incríveis. Esses foram os cinco anos de maior crescimento pessoal e profissional que eu já tive.

E por fim, agradeço a Rodrigo Lins Rodrigues e Américo Amorim, meus orientadores neste trabalho, pela paciência e empenho em me guiar durante esta pesquisa. Graças ao esforço e apoio de vocês, este trabalho foi possível.

“Se você está sempre seguro sobre as escolhas que faz, você não cresce.”
(Heath Ledger)

“A vida nunca está completa sem seus desafios.”
(Stan Lee)

“Impossível é apenas uma palavra usada pelos fracos que acham mais fácil viver no mundo que lhes foi determinado do que explorar o poder que possuem para muda-lo. Nada é impossível.”
(Muhammad Ali)

Resumo

O poder que os vídeo games têm de capturar atenção de seus jogadores trouxe consigo a ideia de usá-los tendo como objetivo principal o reforço no aprendizado na educação. Estudos recentes demonstram que é possível analisar as interações dos jogadores com tais jogos, chamados de Serious Games, para tirar conclusões e mensurar o aprendizado obtido durante a interação com tais jogos. Dado esse contexto, este se propõe a fazer uma análise de dados obtidos a partir da interação de jogadores com um dos jogos, dentre 20, aplicados durante uma pesquisa que comprovou o impacto positivo deles no desenvolvimento de habilidades de leitura e escrita de crianças de 4 anos de idade. Foram selecionados três classificadores: Naive Bayes, Support Vector Machines (SVM) e Regressão Logística, que foram treinados com os dados resultantes da interação desses jogadores com o jogo e demonstrada a taxa de acerto de cada um dos classificadores. Além disso, este trabalho também faz uma análise das interações consideradas mais relevantes na classificação de um dos modelos, encontrando relações entre as palavras propostas como desafio no teste e às presentes no jogo, levantando reflexões que podem ser levadas em consideração durante a produção de um jogo educacional que objetive aperfeiçoar habilidades de leitura e escrita de crianças no ensino infantil.

Palavras-chave: análise de aprendizagem, jogos, educação, mineração de dados

Abstract

The power that video games have to capture their players' attention has brought with it the idea of using them with the main objective of reinforcing learning in educational context. Recent studies demonstrate that it is possible to analyze the interactions of players in such games, called Serious Games, to conclude and measure the learning obtained during interaction in those games. Given this context, this work aims to develop an analysis of data obtained from the interaction of players in one game, out of 20, applied during a research that proved their positive impact on the development of reading and writing skills of 4-years-old children. Three classifiers were selected: Naive Bayes, Support Vector Machines (SVM) and Logistic Regression, which were trained with the data resulting from the interaction of these players with the game and demonstrated the hit rate of each of the classifiers. In addition, this work also makes an analysis of the interactions considered more relevant by one of the models, finding relationships between the words proposed as challenge in the test and those present in the game, raising reflections that can be taken into account during the development of a educational game that aims to improve children's reading and writing skills in early childhood education.

Keywords: game learning analytics, learning analytics, education, games, data mining

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

BBC	*British Broadcasting Corporation*.
IBGE	Instituto Brasileiro de Geografia e Estatística.
INEP	Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira.
IG	*Internet Group*.
LR	*Logistic Regression*.
NB	*Naive Bayes*.
OCDE	Organização para a Cooperação e Desenvolvimento Econômico.
PCA	*Principal Component Analysis*.
PNE	Plano Nacional de Educação.
SL	*Scikit Learn*.
SVC	*C-Support Vector Classification*.
SVM	*Support Vector Machines*.
TAE	Teoria do Aprendizado Estatístico.
TTS	*Text-to-Speech*.
UNICEF	*United Nations International Children's Emergency Fund*.

Sumário

1 Introdução

1.1 Justificativa

Apesar dos grandes avanços nos últimos anos, o Brasil ainda tem cerca de 11,5 milhões de analfabetos, o que corresponde a 7,0% da população de 15 anos ou mais, segundo IBGE (2018). Além disso, ainda segundo IBGE (2018), 13 das 27 unidades da federação não conseguiram alcançar a meta do Plano Nacional de Educação (PNE) de 6.5% de analfabetismo até 2017. Todos os estados da região nordeste estão entre as 13 unidades da federação que não bateram a meta. A região ainda registrou a maior taxa entre as regiões: 14,5%.

O fato deve ser visto como grave dado que, segundo UNICEF (2009), toda criança deve ler e escrever até os 8 anos de idade, porém o número de crianças e adolescentes de 7 a 14 anos que são analfabetos é superior a 2,3 milhões, o que corresponde a 8,3% do total. Muitos estudantes com mais de 8 anos frequentam as salas de aula, mas não sabem ler e escrever.

O número de professores no Brasil passa de 2.5 milhões, segundo INEP (2018) referentes a 2017. Desse universo, 340 mil professores estavam atuando.

Por consequência, segundo matéria publicada por BBC (2018), de acordo com o documento publicado em 2018 pela Organização para a Cooperação e Desenvolvimento Econômico (OCDE), as escolas públicas do Brasil têm 37 alunos por sala de aula no primeiro ano do ensino médio e têm também um dos números mais elevados de alunos por professor, 22. Os dois dados têm influência direta sobre o volume de trabalho dos professores, o que implica na qualidade de ensino.

Paralelo a isso, diversas pesquisas têm sido feitas comprovando os efeitos positivos dos jogos no aprendizado. De acordo com Granic, Lobel e Engels (2014), várias revisões já existem sobre os resultados da aprendizagem associados a jogos educativos, e uma meta-análise feita por Vogel et al. (2006) concluiu que os jogos podem ser uma ferramente importante para promover avanços na reforma educacional necessários para lidar com os desafios de aprendizagem do próximo século.

Também foi verificada uma certa escassez de trabalhos na literatura voltados à aplicação de métodos analíticos na medição do impacto de jogos educacionais na aprendizagem e desenvolvimento de habilidades de leitura e escrita, principalmente até os 5 anos de idade.

1.2 Objetivos

1.2.1 Objetivos Gerais

O objetivo deste trabalho é descobrir relações entre as interações da criança com um jogo educacional desenvolvido para reforçar o aprendizado de aliteração e sua nota em um teste de leitura e escrita.

1.2.2 Objetivos Específicos

1. Desenvolver uma metodologia de análise de resultados das interações de outros jogos semelhantes ao aqui estudado.
2. Verificar se é possível, a partir dos dados coletados através da interações de jogadores com um jogo de aliteração, treinar um classificador capaz de descobrir se a nota da criança foi superior ou inferior à média esperada para sua idade. Dispensando o uso de um pós teste manual.
3. Encontrar quais são as interações mais relevantes no processo de classificação, e levantar discussões sobre as possíveis explicações destes resultados.

1.2.3 Estrutura do Trabalho

- O capítulo 1 trata da introdução ao trabalho, mostrando sua motivação e objetivos.
- O capítulo 2 é dedicado ao embasamento teórico do trabalho, tratando dos benefícios dos jogos educacionais, da análise de dados obtidos a partir de jogos educacionais, técnicas de mineração de dados usando classificadores e citando trabalhos relacionados.
- No capítulo 3 descrevemos como os jogos educacionais da pesquisa foram desenvolvidos, a estrutura da plataforma de desenvolvimento de tais jogos e também dos próprios jogos, descreve detalhes sobre o jogo analisado neste trabalho, bem como os eventos produzidos por ele. O capítulo também trata de como foi feita a extração de dados.
- O capítulo 4 se destina à apresentação dos resultados obtidos e às discussões de tais resultados.
- Por fim, no capítulo 5 é feita uma conclusão dos resultados e discussões apresentados. Além disso, são colocadas as limitações desta pesquisa e são apontados caminhos possíveis para futuros trabalhos.

2 Referencial Teórico

2.1 Aprendizagem baseada em jogos

Essa seção trata das descobertas relacionadas aos benefícios de jogos eletrônicos no geral, do porquê de jogos serem uma plataforma de boa aceitação pelos alunos comparada às outras para objetivos educacionais e da definição de *Serious Games*.

2.1.1 Os benefícios de jogos eletrônicos

Os benefícios do ato de se envolver com um jogo (ou brincar) de forma geral vem sendo estudado por décadas. O impacto positivo de jogar é um tema recorrente entre grandes pesquisadores da área de educação. Esses impactos se mostram nas mais diversas áreas, como no desenvolvimento de habilidades emocionais, cognitivas, sociais e motivacionais (GRANIC; LOBEL; ENGELS, 2014).

Ao contrário da crença de que jogar jogos eletrônicos é uma atividade intelectuamente preguiçosa, Granic, Lobel e Engels (2014) afirma que jogar jogos eletrônicos promove o desenvolvimento de uma ampla gama de habilidades cognitivas. Ainda segundo Granic, Lobel e Engels (2014), os jogos eletrônicos são a plataforma de treinamento ideal para desenvolver a teoria da inteligência incremental, descrita por Dweck (2005). A teoria de inteligência incremental é um credo que a criança desenvolve sobre sua própria inteligência caso receba recompensas por seu próprio esforço em atividades que desenvolve. Ela passa a crer que sua inteligência é algo maleável, que pode ser desenvolvida através de seu próprio esforço.

Na área de educacional, os jogos eletrônicos também podem desenvolver habilidades importantes para graduados, como comunicação, resolução de problemas e desenvoltura. De acordo com Barr (2017), jogos digitais comerciais podem ter impactos positivos nessas habilidades, sugerindo que os jogos podem ter um papel na educação superior. O estudo ainda sugere que esse tipo de habilidade pode ser aprimorada em um espaço de tempo relativamente curto.

O estudo de Amorim (2018), que envolveu cerca de 700 estudantes de 4 anos, sugere que os jogos digitais educacionais podem ser uma ferramenta importante no desenvolvimento de habilidades de leitura e escrita, uma vez que o grupo de estudantes que interagiu com os jogos educacionais em sua pesquisa conseguiu resultados superiores em relação ao grupo que prosseguiu com as atividades comuns da escola, e que não interagiu com os jogos. Além de demonstrar que os professores conseguiram aprender como usar jogos educacionais para o reforço da aprendizagem dos seus

alunos.

Esses trabalhos demonstram que ao interagir com jogos eletrônicos, estamos exercitando e desenvolvendo certas habilidades, mesmo se tratando de jogos desenvolvidos com o propósito de entretenimento.

2.1.2 Jogos de video game comparado a outras atividades lúdicas

A caracteristica mais distintiva entre video games e outras mídias, como livros, televisão ou filme é que os jogos são interativos. Os jogadores não podem se render passivamente ao enredo de um jogo. Em vez disso, jogos de video game são desenvolvidos e pensados para que os jogadores ativamente se envolvam nos seus sistemas e para que esses sistemas reajam aos comportamentos e ações dos jogadores.

Existem milhões de jogos diferentes, com os mais vastos temas e finalidades. Esses jogos podem ser jogados cooperativamente ou competitivamente, sozinho, com outros jogadores fisicamente presentes, ou com milhares de outros jogadores online. Além de poderem ser jogados nas mais diversas plataformas, desde consoles (como o Playstation ou o Xbox) a computadores ou celulares. Devido a essa diversidade em termos de generos e às várias plataformas onde jogos digitais podem ser acessados, uma definição compreensível do que são jogos contemporâneos é bastante complicada de se desenvolver (GRANIC; LOBEL; ENGELS, 2014).

Em jogos de video game, os jogadores sempre enfrentam um desafio que tem um objetivo especifico, para a resolução desse desafio eles precisam enfrentar um certo tipo de obstáculo ou força oposta. Adicionalmente, videogames provêem não somente os meios e as regras do jogo, como também são um ambiente interativo de jogo, o que não é encontrado em muitos outros jogos, além do próprio ambiente interativo ser sempre virtual (FABRICATORE, 2000).

2.1.3 *Serious Games*

Apesar de não existir uma única definição simples do que significam *Serious Games*, o termo está se tornando cada vez mais popular, e vem se estabelecendo. Em termos gerais, *Serious Games* é um termo associado a jogos que não têm o entretenimento como propósito principal, mas também pode se referir a jogos usados para treinamento, propaganda, simulação, ou educação que são desenvolvidos para funcionar num computador pessoal ou em consoles de vídeo game (SUSI; JOHANNESSON; BACKLUND, 2007).

Entre os anos de 1950 e 2000, somente 10% dos jogos identificados como *Serious Games* foram desenvolvidos com o propósito de treinamento e desenvolvimento de habilidades específicas. Isso se deve ao fato de que jogos com o único propósito de

propagar mensagens específicas eram mais fáceis e baratos de se desenvolver, pois os programadores não precisavam desenvolver um componente de avaliação. Também não era necessário se preocupar demasiadamente com a precisão que a mensagem era passada ou com o quão compreensível ela era (LOH; SHENG; IFENTHALER, 2015).

A Figura 1 sumariza graficamente os achados de Loh, Sheng e Ifenthaler (2015).

Figura 1 – Fluxograma de definição de um Serious Game

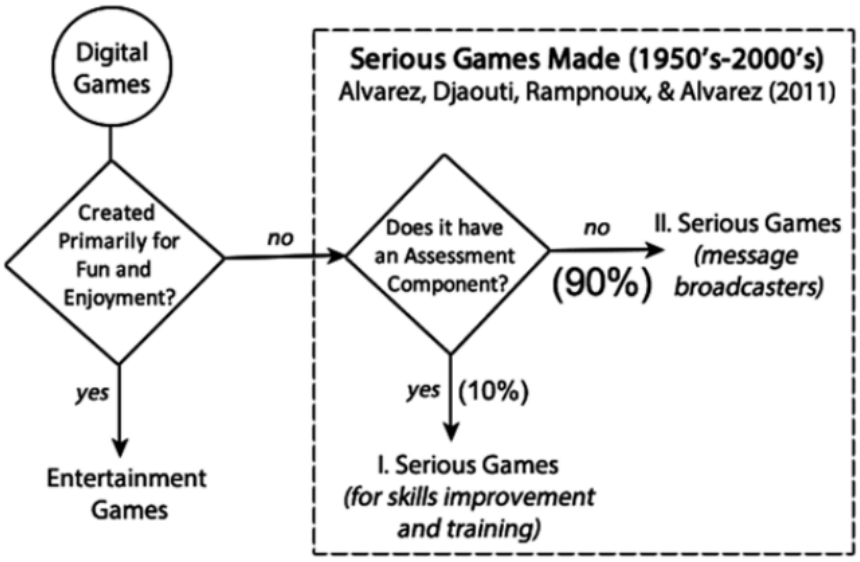


Imagem retirada do artigo de Loh, Sheng e Ifenthaler (2015)

Ainda segundo Loh, Sheng e Ifenthaler (2015), por definição de Scientists (2006), as características esperadas em um Serious Game são as seguintes:

1) Objetivos claros;

2) Tarefas repetitivas, para que se possa desenvolver uma certa maestria;

3) Monitoramento do progresso no aprendizado do jogador;

4) Encorajar o investimento de tempo do jogador nas tarefas propostas, usando artifícios motivacionais;

5) Ajuste do nível de dificuldade das tarefas para prover compatibilidade com o atual nível de desenvolvimento do jogador;

Embora os *Serious Games* sejam bastante utilizados no contexto de educação, desde temas tratados em escola ao ensino de habilidades mais específicas, surge a necessidade de obter mais informações através da utilização do jogo que não seriam possíveis de se obter facilmente a partir da prova tradicional como, por exemplo, a identificação de assuntos ou perguntas específicas que podem ser consideradas mais

difíceis para os alunos. E isso pode ser feito através da análise das ações dos jogadores dentro do jogo.

2.2 Game Learning Analytics

Esta seção trata de técnicas e tipos de dados coletados com mais frequência nos estudos sobre interações com jogos, bem como formas de obter informações relevantes a partir desses dados.

2.2.1 Medições de dados em Serious Games

2.2.1.1 Tipos de dados gerados por usuários

Antes de medir o que os jogadores fazem dentro de um ambiente de jogo, precisamos encontrar uma forma de coletar os dados gerados pelos jogadores. Existem dois tipos de dados gerados em serious games: dados *in situ* e *ex situ*.

Dados *ex situ* são coletados fora do sistema onde o objeto ou evento em observação está. Um questionário, por exemplo, é um tipo de dado dessa categoria uma vez que ele é tipicamente feito no mundo real e não dentro do ambiente de jogo. Normalmente, os pesquisadores coletam dados *ex situ* por conveniência ou devido a restrições. Essas restrições normalmente incluem custos da pesquisa ou condições caixa preta onde não se pode acessar os detalhes internos do sistema (LOH, 2015).

Figura 2 – Análise *ex situ*, onde o jogo avaliado é considerado uma caixa preta

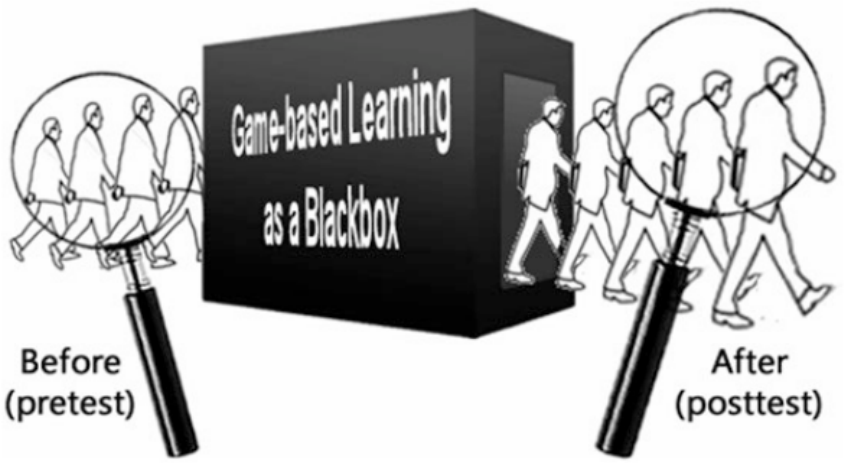


Imagem retirada do artigo de Loh (2015)

Diferentemente do *ex situ*, dados *in situ* são coletados ”no habitat natural ou no sistema” onde o objeto interativo vive. Portanto, nessa abordagem o objeto digital é visto como uma caixa branca, aberta para a manipulação de conteúdo e coleta

dos dados *in situ*. Um bom programador pode criar estruturas e usá-las também para automatizar o processo de coleta dos dados in-situ (LOH, 2015).

Obviamente, os métodos de coleta *in situ* são preferíveis aos *ex situ*, uma vez que eles eliminam muitos dados subjetivos coletados em pesquisas, entrevistas e relatórios.

2.2.1.2 Event Listeners

Para que o jogo saiba que evento de aprendizado ou de jogo aconteceu, é necessário o uso de uma função de event listener. Quase todas as ferramentas de jogos disponibilizam algum tipo de event listener para que o programa consiga identificar uma ampla gama de eventos que podem ocorrer dentro do sistema. A função de event listening é parte essencial da análise de dados em serious games e deve ser incorporada no ambiente de jogo se possível, ou o mais cedo possível no processo de desenvolvimento de jogos digitais (LOH, 2015).

2.2.1.3 Análise de dados de Serious Games

Segundo Loh (2015), baseado nos estudos de Moura, Nasr e Shaw (2011), usando mineração de dados e técnicas de categorização de comportamento, as ações de usuário coletadas a partir de eventos podem ser agregadas em padrões que não são facilmente detectadas por métodos tradicionais.

Baseado na combinação de categorias de comportamento, pesquisadores podem desenvolver perfis de comportamento de jogadores usando técnicas de aprendizado de máquina supervisionada ou não supervisionada, para treinar e produzir novas políticas ou *insights*: como sugestões de melhoria no design de jogos. Além disso, se corretamente verificado, um modelo preditivo poderia ser usado para avaliar a performance de novos jogadores a partir de uma estrutura ou algoritmo treinado com os dados previamente coletados (LOH, 2015).

2.2.1.4 Dados de Log

Alguns jogos oferecem oportunidades de customizar sua experiência dentro dele, através de opções como ”escolha seu avatar” até a possibilidade do jogador fazer escolhas que mudam completamente o rumo da aventura vivida pelo jogador dentro do jogo. Isso leva a um certo problema na análise e avaliação de padrões de interação, uma vez que cada experiência pode ser bastante distinta. Recentemente, pesquisadores estão estudando esse cenário através dos logs (ou registros) que o sistema gera (SNOW; MCNAMARA, 2015).

Os dados de log possuem o potencial de capturar essas múltiplas faces das decisões de usuários dentro dos jogos, o que levou pesquisadores a intecionalmente fazer com que seus ambientes de jogo registrassem todas as interações dos jogadores com o jogo. Um dos benefícios de se desenvolver um jogo com que registra as ações dos jogadores é a possibilidade de coletar dados relevantes sem interromper a interação do usuário com os jogos. Em outras palavras, essas medidas coletadas são virtualmente invisíveis para o jogador (SNOW; MCNAMARA, 2015).

2.2.1.5 Análises em dados de log

Um importante objetivo ao analisar dados de log é estabelecer metodos de avaliar e quantificar variações que se manifestam dentro dos dados de log. Esses metodos quantitativos possibilitarão aos pesquisadores avaliar qual interação ou padrão de comportamento pode explicar quais experiências dos jogadores dentro de um ambiente de jogo e como as variações nessa experiência podem influenciar no aprendizado observado (SNOW; MCNAMARA, 2015).

2.2.2 Técnicas de coleta de dados

Segundo a análise feita por Smith, Blackmore e Nesbitt (2015), a partir de 299 trabalhos avaliados sobre serious games no período de 1981 a 2012, foram encontradas cerca de 510 técnicas diferentes de coleta de dados em 188 desses estudos. Destas 510 técnicas, 33% ocorreram antes dos jogos (pré-teste), 21% durante a interação das pessoas com os jogos e 46% foram feitas após a interação com os jogos (pós-teste).

No que se refere a técnicas específicas, o questionário é a mais popular técnica de pré-teste, aparecendo em 52% dos estudos, 42% usaram alguma outra forma de teste, 4% usaram a entrevista e somente 2% usaram uma observação indireta da interação.

Para a fase de pós-teste, o questionário ainda é a forma mais popular de avaliação, com 46% de presença nos estudos. Além dele, 37% usaram uma outra forma diferente de teste e 13% usaram uma entrevista.

Assim, Smith, Blackmore e Nesbitt (2015) concluiu que a quantidade de dados coletados é ampla em escopo, medindo habilidades de desempenho desejados ou fatores comportamentais relacionados ao processo e resultados. Por exemplo, os dados podem objetivar medir mudanças no conhecimento, comportamento, habilidades ou atitudes. A revisão também demonstrou que a maioria dos dados são coletados após a interação com os jogos, e que a forma mais incomum de coleta de dados é durante a interação das pessoas com os jogos. Isso pode refletir uma dificuldade comum na

captura de dados durante as interações.

2.3 Mineração de Dados

Da área de mineração de dados, descrevemos nessa seção uma parte bastante específica: O funcionamento dos classificadores, alguns algoritmos usados na tarefa de classificação, metodologias de avaliação de classificadores e a análise de componentes principais.

A técnica de classificação foi escolhida para analisar os dados deste trabalho pela sua capacidade de satisfazer todos os objetivos propostos.

2.3.1 Classificadores

Comumente denomina-se classificação o processo pelo qual se determina um mapeamento capaz de indicar a qual classe pertence qualquer exemplar de um domínio sob análise, com base em um conjunto de dados já classificado. A tarefa de classificação pode ser descrita como a busca por uma função capaz de mapear um conjunto X de vetores de entrada (ou exemplares) para um conjunto finito C de rótulos. A tarefa de classificação pode ser dividida em, pelo menos, duas categorias: classificação binária e classificação multiclasse. Na classificação binária, a cardinalidade de C é 2 e para os casos em que a cardinalidade de C > 2, o problema é considerado de múltiplas classes.(SILVA; PERES; BOSCARIOLI, 2017)

Tabela 1 – Exemplo de conjunto de vetores e rótulo de um classificador binário.

Vetor 1	Vetor 2	Vetor 3	Vetor 4	Rótulo
1	Brasil	.955	12	0
2	Argentina	.938	15	0
3	EUA	.937	13	0
4	Paraguai	.921	10	1
5	Chile	.920	25	1

Um classificador binário se aplica a casos como o deste trabalho, onde se quer descobrir se um determinado estudante teve desempenho superior ou inferior à média no pós teste escrito. Já o classificador de múltiplas classes poderia auxiliar um avaliador à classificar um serviço dentro de um conjunto finito de opções como: excelente, bom, regular e ruim.

Dentre os diversos algoritmos existentes capazes de gerar um modelo classificador, podemos citar :

- Naive Bayes

- Máquina de Vetores Suporte (SVM)
- Regressão Logística

2.3.2 Classificador Naive Bayes

O Classificador Naive Bayes simplifica bastante o aprendizado do algoritmo a partir do presuposto de que as características dos exemplares (chamadas de features) são classes independentes umas das outras. Apesar da independência ser geralmente um presuposto muito pobre, na prática os modelos gerados pelo algoritmo de Naive Bayes apresentam resultados competitivos comparado a classificadores mais sofisticados (RISH, 2001).

O modelo independente de Naive Bayes é baseado na seguinte estimativa:

$$R = \frac{P(i \mid X)}{P(j \mid B)} = \frac{P(i)P(X \mid i)}{P(j)P(B \mid j)} = \frac{P(i)\prod P(X \mid i)}{P(j)\prod P(B \mid j)} \tag{2.1}$$

Comparando essas duas probabilidades (i e j), a maior delas indica qual valor do rótulo é mais provável de acontecer.

Uma vez que o classificador de Bayes usa operações de produto para calcular as probabilidades P(X,i), ele é especialmente propenso a ser indevidamente impactado por probabilidades zero. Isso pode ser evitado usando o estimador de Laplace ou estimadores m, adicionando um em todos os numeradores e adicionando ao denominador o valor equivalente à soma de todos os uns adicionados no numerador. A maior vantagem do uso do classificador Naive Bayes é o baixo tempo computacional para se fazer o treinamento (KOTSIANTIS, 2001).

2.3.3 Máquina de Vetores de Suporte (SVM)

As SVMs são baseadas na Teoria do Aprendizado Estatístico (TAE), que visa estabelecer condições matemáticas que permitam a escolha de um classificador f' capaz de ter um bom desempenho para os conjuntos de dados de treinamento e de teste, sem dar atenção exarcebada a qualquer ponto individual do mesmo. Os resultados da aplicação dessa técnica são comparáveis aos obtidos por modelos de Redes Neurais e até superiores em algumas tarefas específicas, como detecção de faces em imagens, categorização de textos e aplicações de Bioinformática (LORENA, 2003).

O funcionamento das SVMs gira em torno da noção de uma margem, que é um dos lados de um hiperplano que separa dados de duas classes diferentes. Uma SVM busca maximizar essa margem e, assim, determinar a maior distância possível entre o hiperplano definido e as instâncias de cada lado dele (KOTSIANTIS, 2001).

Figura 3 – Demonstração de um Hiperplano ótimo e suas margens

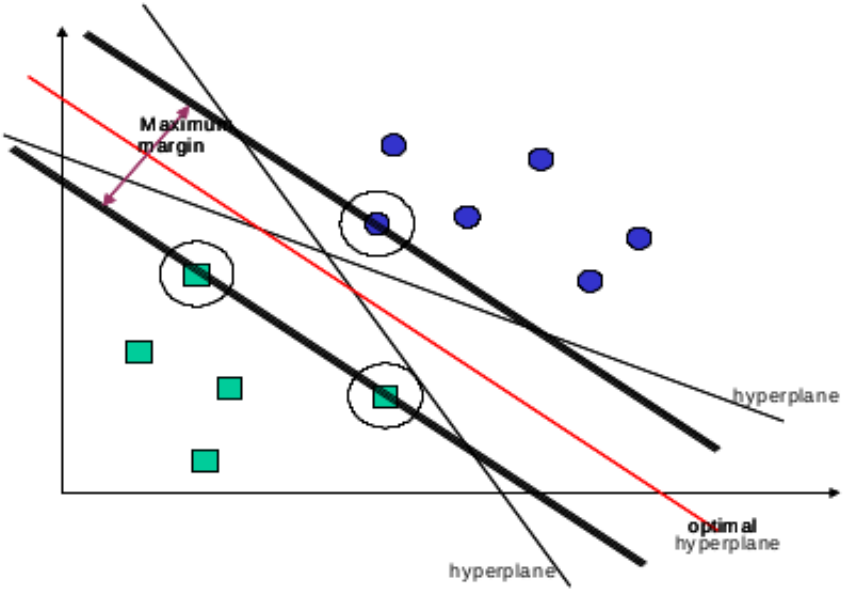


Imagem retirada do artigo de Kotsiantis (2001)

2.3.3.1 SVMs Lineares

A formulação mais simples de SVM, com margens rígidas, lida com problemas linearmente separáveis. Essa formulação foi posteriormente estendida para definir fronteiras lineares sobre conjuntos de dados mais gerais usando margens suaves. A partir desses conceitos iniciais, pode-se obter fronteiras não lineares com SVMs, por meio de uma extensão das SVMs lineares. (LORENA, 2003)

Seja T um conjunto de treinamento com n dados $x_i \in X$ e seus respectivos rótulos $y_i \in Y$, em que X constitui o espaço dos dados e $Y = \{-1, +1\}$. T é linearmente separável se é possível separar os dados das classes +1 e −1 por um hiperplano. A equação do hiperplano é apresentada abaixo:

$$f(x) = w \cdot x + b = 0 \tag{2.2}$$

Onde $w \cdot x$ é o produto escalar entre os vetores w e x. E $w \in X$ é o vetor normal ao hiperplano descrito e $\frac{b}{||w||}$ corresponde à distância do hiperplano em relação à origem, com $b \in \mathbf{R}$. Essa equação define o espaço de dados X em duas regiões. Onde $w \cdot x + b > 0$ e $w \cdot x + b < 0$. Assim, é possível usar uma função sinal que classifica +1 para o primeiro caso e -1 para o segundo (LORENA, 2003).

Mesmo que a margem máxima permita que o SVM encontre a melhor separação, dentre vários hiperplanos, para vários conjuntos de dados, existem casos onde o SVM não consegue achar nenhum hiperplano separador devido ao fato do conjunto de

dados conter instâncias mal classificadas ou outliers. O problema pode ser abordado através do uso de margens suaves, que aceitem algumas exceções nas instâncias de treinamento (KOTSIANTIS, 2001).

Isso pode ser feito através da introdução de variáveis de folga positivas ξ_i, onde $i = 1, \ldots, N$ nas restrições, que passam a ser:

$$w \cdot x - b \geq 1 - \xi \tag{2.3}$$

$$w \cdot x - b \leq -1 + \xi \tag{2.4}$$

$$\xi \geq 0 \tag{2.5}$$

2.3.3.2 SVMs não lineares

As SVMs lidam com problemas não lineares mapeando o conjunto de treinamento de seu espaço original, referenciado como de entradas, para um novo espaço de dimensão superior, chamado de espaço de características ou feature space. Dado o mapeamento $\Phi : X \rightarrow \Im$, onde X é o espaço de entradas e $\Im$ é o espaço de características, se escolhemos um Φ apropriado, temos um conjunto de treinamento mapeado em $\Im$ que pode ser separado por uma SVM linear, como demonstrado na Figura (4).

O uso desse procedimento é motivado pelo teorema de Cover, que determina que dado um conjunto de dados não linear no espaço de entradas X, pode-se transformar X em um espaço de características $\Im$ no qual existe uma alta probabilidade de se ter dados linearmente separáveis. Para isso duas condições devem ser satisfeitas: A transformação precisa ser não linear e a dimensão do espaço de características deve ser suficientemente alta (LORENA, 2003).

2.3.4 Regressão Logística

Na classificação de dados, objetiva-se predizer um dado rótulo para um exemplar qualquer que não pertence ao conjunto de dados usado no treinamento do modelo de classificação. Quando o rótulo Y é do tipo numérico, seja contínuo ou discreto, temos um problema de regressão ou predição numérica. A regressão é usada para estimar valores a partir de um conjunto de dados históricos como, por exemplo, em problemas de indicadores econômicos ou de mercado futuro, onde se pretende prever o próximo valor analisando dados de atributos descritivos (SILVA; PERES; BOSCARIOLI, 2017).

Especificamente na regressão logística, é possível estimar a probabilidade de ocorrência de um evento diretamente. No caso de uma variável dependente Y ter cardi-

Figura 4 – (a) Conjunto de dados não linear; (b) Fronteira não linear no espaço de entradas; (c) Fronteira linear no espaço de características

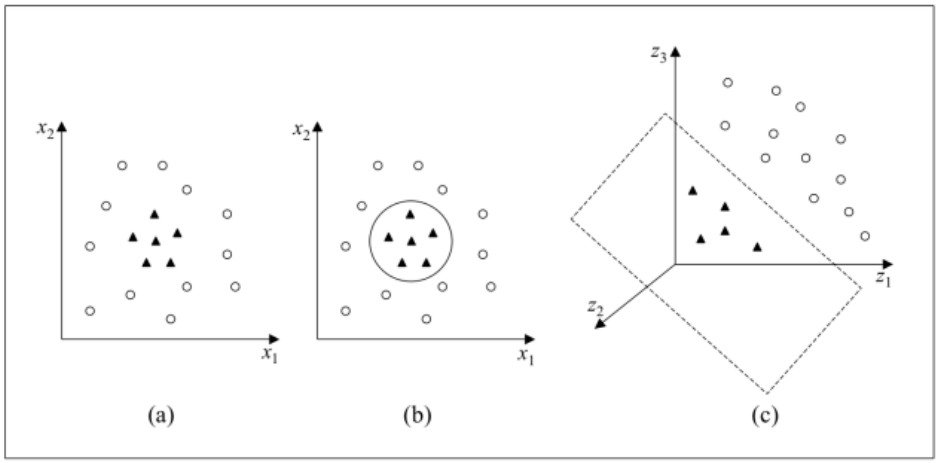


Imagem retirada do artigo de Lorena (2003)

nalidade 2 e haver um conjunto de p variáveis independentes $X_1, X_2, \ldots, X_p$, o modelo de regressão logística pode ser escrito da seguinte forma:

$$P(Y = 1) = \frac{1}{1 + e^{-g(x)}} \tag{2.6}$$

onde,

$$g(x) = B_0 + B_1 X_1 + \cdots + B_p X_p \tag{2.7}$$

Os coeficientes $B_0, B_1, ..., B_p$ são estimados a partir do conjunto de dados, através do método da máxima verossimilhança, que pode encontrar uma combinação de coeficientes com máxima probabilidade de uma amostra ter sido observada. A partir de uma combinação de coeficientes $B_0, B_1, ..., B_p$ e da variação dos valores de X, tem-se uma curva logística de comportamento probabilístico no formato de S, característico da regressão logística (MINUSSI; DAMACENA; JR, 2002).

Esse formato dá à regressão lógistica alto grau de generalização, aliado a aspectos como:

a) Quando g(x) $\rightarrow \infty$, então P (Y = 1) $\rightarrow 1$

b) Quando g(x) $\rightarrow -\infty$, então P (Y = 1) $\rightarrow 0$

Para utilizar o modelo de regressão logística para discriminação de dois grupos, a regra de classificação é:

1) Se P(Y=1) > 0,5 então classifica-se Y=1;

2) Em caso contrário classifica-se Y=0.

Figura 5 – Curva Sigmóide, característica da Regressão Logística

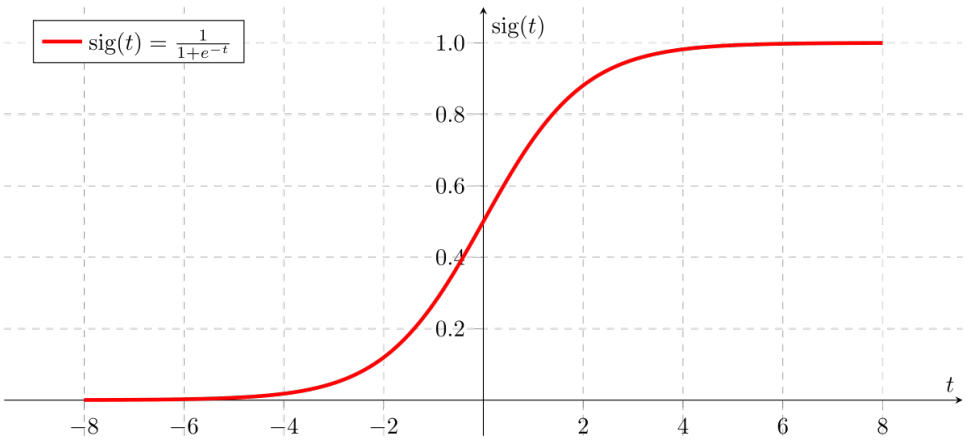


Imagem retirada da publicação de Swaminathan (2018)

2.3.5 Técnicas de Avaliação de Classificadores

Nesta seção são apresentadas técnicas tradicionais de medição do desempenho dos classificadores: Holdout e Validação Cruzada.

2.3.5.1 Holdout

Em sua forma mais simples, a estratégia de Holdout pressupõe a criação de dois subconjuntos de dados disjuntos, a partir do conjunto de dados disponível para uso no treinamento do modelo. Um dos subconjuntos será usado para o treinamento, também chamado de indução, e o segundo para testes após o término do treinamento e, consequentemente para a aplicação das medidas de avaliação do modelo. Tradicionalmente, os dois subconjuntos são gerados de forma que 70% dos exemplares do conjunto façam parte do subconjunto de treinamento e os outros 30% sejam alocados no subconjunto de teste. Também pode-se usar as porcentagens de 60/40. Os exemplares a serem alocados em cada um dos subconjuntos devem ser escolhidos aleatoriamente.

A execução dessa estratégia uma única vez não é suficiente, uma vez que a aleatoriedade imposta na geração dos subconjuntos pode levar a situações que beneficiem o teste do modelo. Sabendo que o conjunto de dados é uma amostra do universo de dados possível em certo contexto, é interessante maximizar as chances de gerar avaliações estatísticamente confiáveis. Então, a estratégia do holdout deveria ser aplicada N vezes, considerando os mesmos conjuntos de parâmetros determinados no algoritmo de geração dos modelos, porém, variando a composição dos subconjuntos de treinamento e de teste. A avaliação do modelo deveria ser gerada pela aplicação de medidas estatísticas (média, desvio-padrão, intervalos de confiança) ao conjunto de avaliações obtido para os modelos gerados nas N execuções da estratégia (SILVA;

PERES; BOSCARIOLI, 2017).

2.3.5.2 Validação Cruzada

Na estratégia de validação cruzada, todos os exemplares são parte, em algum momento, do conjunto de dados usado para teste do modelo preditivo. Para implementar essa situação, o conjunto de dados é dividido em K subconjuntos disjuntos, com alocação aleatória de exemplares para cada subconjunto. Assim, o conjunto de dados D é dividido nos subconjuntos $D_1, \ldots, D_k, \ldots, D_K$, esses subconjuntos são conhecidos como folds. A estratégia de validação cruzada é também conhecida como validação cruzada k-fold.

Então, um dos folds é reservado para ser usado como conjunto de testes, enquanto os outros K-1 conjuntos restantes compõe o conjunto de treinamento, esse procedimento é repetido K vezes, alterando o fold reservado para testes do modelo. Portanto, para cada execução desse procedimento, um modelo diferente é gerado e, a ele, devem ser aplicadas as medidas de avaliação de desempenho desejadas (SILVA; PERES; BOSCARIOLI, 2017).

A figura abaixo representa o funcionamento da validação cruzada.

Figura 6 – Uma representação gráfica de como funciona a validação cruzada.

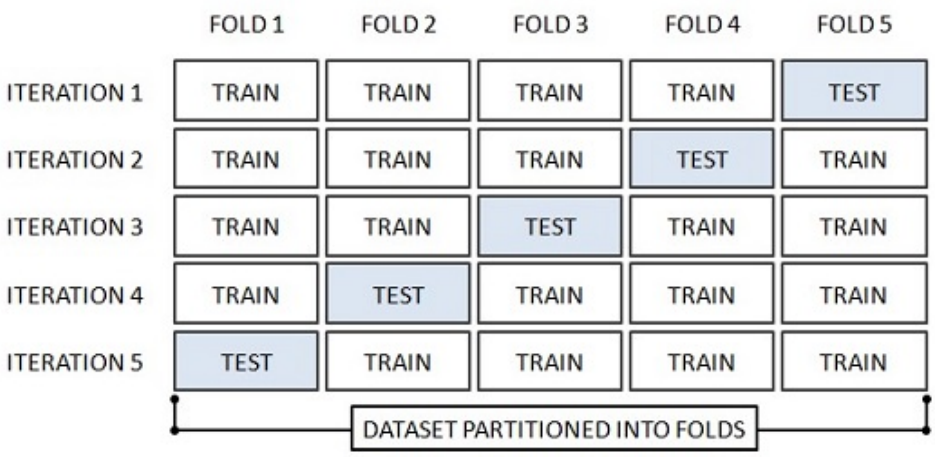


Imagem retirada da publicação de Mueller (2016)

A avaliação final referente à geração de modelos preditivos pode ser feita, pelo menos, de duas diferentes formas:

a) Aplicando medidas estatísticas, como média, desvio-padrão e intervalo de confiança.

b) Somando-se o desempenho obtido pelos K modelos gerados e dividindo essa soma pelo número de exemplares do conjunto de dados original.

A medida de avaliação mais comumente usada para classificadores é a Acurácia ou taxa de classificações corretas, sendo dada por:

$$|y - f(x) = 0| \tag{2.8}$$

em que |*| representa a contagem de vezes que * é verdadeiro, f é o modelo preditivo, x é o subconjunto de dados sob o qual o modelo está sendo avaliado, $f(*)$ é a classificação fornecida pelo modelo preditivo para cada um dos exemplares e y é a classe esperada como resposta. A acurácia também pode ser escrita em termos de erro de generalização e uma função de perda binária e, assim, ser interpretada como a probabilidade de ocorrer uma classificação correta (SILVA; PERES; BOSCARIOLI, 2017).

2.3.6 Análise de Componentes Principais

A análise de componentes principais é uma técnica da estatística multivariada que consiste em transformar um conjunto de variáveis originais em outro conjunto de variáveis de mesma dimensão denominadas de componentes principais. Esses componentes principais apresentam propriedades importantes:

1) Cada componente principal é uma combinação linear de todas as variáveis originais.

2) São independentes entre si e estimados com o propósito de reter, em ordem de estimação, o máximo de informação, em termos da variação total contida nos dados.

A análise de componentes principais é associada à ideia de redução de massa de dados, perdendo o mínimo possível de informação. Procura-se redistribuir a variação observada nos eixos originais de forma a se obter um conjunto de eixos ortogonais não correlacionados. Esta técnica pode ser utilizada para geração de índices e agrupamento de indivíduos. A análise agrupa os indivíduos segundo suas variâncias, ou seja, segundo seu comportamento dentro da população, representado pela variação do conjunto de características que define o indivíduo. Portanto a técnica agrupa os indivíduos de uma população segundo a variação de suas características.

Apesar das técnicas de análise multivariada terem sido desenvolvidas para resolver problemas específicos, principalmente de Biologia e Psicologia, podem ser também utilizadas para resolver problemas em diversas áreas do conhecimento. (VARELLA, 2008)

Uma análise de componentes principais começa com dados de p variáveis para n indivíduos, como indicado na tabela abaixo:

Segundo Araujo e Coelho (2009), uma análise de componentes principais envolve encontrar os autovalores de uma matriz de covariâncias amostral. A matriz de

Caso	X_1	X_2	...	X_p
1	a_{11}	a_{21}	...	a_{1p}
2	a_{21}	a_{22}	...	a_{2p}
⋮	⋮	⋮	⋮	⋮
n	a_{n1}	a_{n2}	...	a_{np}

covariância é simétrica e possui o seguinte formato:

$$\mathbf{C} = \begin{pmatrix} c_{11} & c_{12} & \dots & c_{1p} \\ c_{21} & c_{22} & \dots & c_{2p} \\ \vdots & \vdots & & \vdots \\ c_{p1} & c_{p2} & \dots & c_{pp} \end{pmatrix}$$

Em que o elemento c_{ii} na i-agonal é a variância de X_i e o termo fora da diagonal c_{ij} é a covariância entre as variáveis X_i X_j. As variâncias dos componentes principais são os autovalores da matriz C. Existem p destes autovalores, alguns dos quais podem ser zero. A fim de evitar uma ou duas variáveis tendo um indevida influência nos componentes principais, é usual codificar as variáveis $X_1, X_2, \dots, X_p$ para terem médias zero e variâncias iguais a 1 no início de uma análise. A matriz C então toma forma:

$$\mathbf{C} = \begin{pmatrix} 1 & c_{12} & \dots & c_{1p} \\ c_{21} & 1 & \dots & c_{2p} \\ \vdots & \vdots & & \vdots \\ c_{p1} & c_{p2} & \dots & 1 \end{pmatrix}$$

em que $c_{ij} = c_{ji}$ é a correlação entre X_i e X_j. Em outras palavras, a análise de componentes principais é feita sobre a matriz de correlação. Neste caso, a soma dos termos da diagonal, e, portanto, a soma dos autovalores, é igual a p, o número de variávies X.

Os passos em uma análise de componentes principais podem agora ser estabelecidos:

- Comece codificando as variáveis $X_1, X_2, ..., X_p$ para terem médias zero e variâncias unitárias. isto é usual, mas é omitido em alguns casos em que se assume que a importância das variáveis é refletida em suas variâncias.
- Calcule a matriz de covariâncias C. Esta é uma matriz de correlações se o passo 1 foi feito.
- Encontre os autovalores $\lambda_1, \lambda_2, \dots, \lambda_p$ e os correspondentes autovetores $a_1, a_2, ..., a_p$. Os coeficientes do i-ésimo componente principal são então os elementos de a_i, enquanto que λ_i é sua variância.

- Descarte quaisquer componentes que explicam somente uma pequena proporção da variação nos dados. Por exemplo, começando com 20 variáveis, pode ser obtido que os primeiros três componentes expliquem 90% da variância total. Com base nisto, os outros 17 componentes podem ser razoavelmente ignorados.

2.4 Trabalhos Relacionados

Já existem diversos trabalhos relacionados à efetividade dos jogos na aprendizagem, seja na alfabetização, no ensino superior ou para o desenvolvimento de habilidades relacionadas à inteligência emocional.

Em relação à game learning analytics aplicados a jogos usados no contexto escolar, Fernández (2017) fez uma analíse exploratória completa de dados obtidos antes, durante e depois da interação de estudantes, um jogo educacional direcionado ao ensino de técnicas de primeiros socorros, objetivando estabelecer qual o nível de conhecimento dos estudantes sobre o tema a partir de suas interações com jogos.

Ainda sobre game learning analytics, voltado à tirar conclusões no mundo real a partir de uma interação digital, Keshtkar et al. (2014) usou técnicas de mineração e processamento de linguagem natural para detectar automaticamente personalidade e comportamentos de estudantes dentro de um jogo educacional onde estudantes atuavam como estagiários em uma empresa de planejamento urbano e discutiam em grupo suas ideais. O estudo procurou definir a personalidade de cada aluno a partir da forma que ele se comunica no jogo, e isso foi feito usando algoritmos de aprendizado de máquina tais como Naive Bayes, Support Vector Machine (SVM) e Árvores de Decisão.

Na área educacional, especialmente no ensino superior, Slimani et al. (2018) apresentou técnicas de mineração de dados educacionais para discutir análises de aprendizado através de serious games, além de prover uma análise de dados da experiência dos jogadores coletada através do jogo educacional chamado ELISA, usado para ensinar a estudantes de biologia uma técnica imunológica para a determinação de anticorpos ANTI-HIV. Fazendo, também, uma avaliação crítica dos seus resultados, incluindo limitações no estudo e fazendo sugestões para futuras pesquisas relacionadas à análise de aprendizado e serious games.

E na área de educação infantil e alfabetização, Amorim (2018) desenvolveu 20 jogos, dentre eles o jogo analisado neste trabalho, aplicou-os em diversas escolas e demonstrou, através de testes ex situ, a relevância da interação dos estudantes com estes jogos, apontando que eles possuem impacto no desenvolvimento das habilidades de leitura e escrita dos estudantes. Além disso, fez uma análise inicial dos dados coletados in situ pelos jogos.

Esses trabalhos demonstram que existe a possibilidade de medir a evolução de um jogador numa habilidade específica através da análise da interação dele com um jogo eletrônico.

3 Metodologia

3.1 Estrutura da Plataforma

Os 20 jogos desenvolvidos para a pesquisa de Amorim (2018) foram aplicados em 18 escolas da cidade onde a pesquisa foi feita. A pesquisa foi viabilizada através de duas plataformas distintas: Uma que permitia que os pesquisadores desenvolvessem os jogos educacionais e classificassem tais jogos em agrupamentos (ou pacotes) a serem distribuídos aos alunos, chamado de LAB, e um aplicativo mobile que viabilizava o acesso aos pacotes de jogos, o jogo em si e a coleta de dados, chamado de APP.

A primeira plataforma, ou LAB, proporcionava aos pesquisadores uma gama de mecânicas de jogos como ponto de partida, para permitir que os pesquisadores se concentrassem unicamente no teor educacional dos jogos a serem desenvolvidos. O LAB ainda possui formas internas de compartilhar tais jogos, e a principal usada na pesquisa é o compartilhamento para usuários do aplicativo. O banco de dados do servidor do LAB também é responsável pelo armazenamento dos logs dos eventos gerados no APP

Já o APP tem como principal responsabilidade servir os jogos produzidos e dar feedbacks sobre a performance das crianças através da exibição de medalhas (ouro, prata e bronze), além de identificar uma pré determinada lista de ações do usuário que deve ser armazenada no LAB, essas ações também são chamadas de eventos. Algumas escolas participantes da pesquisa não possuiam internet Wi-Fi até a data em que ela foi feita, portanto foi necessário desenvolver no APP a funcionalidade de armazenar internamente todos os eventos produzidos pelos usuários, enviando posteriormente todos eles quando uma conexão com a internet estiver disponível.

3.2 Estrutura dos jogos

Dentro do LAB, os jogos são identificados como projetos. Cada um dos projetos é composto por telas que possuem propriedades editáveis.

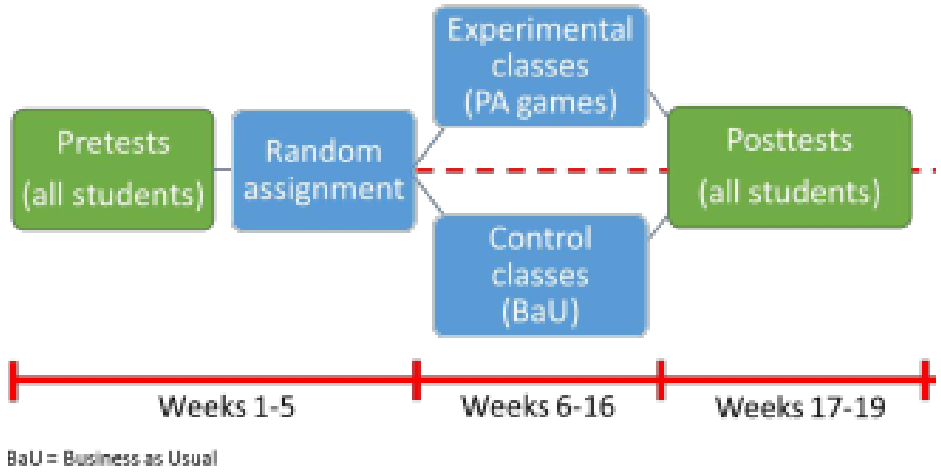
Além disso, o LAB também proporciona templates de projetos com uma listagem de telas pré definida e a possibilidade de permitir qualquer texto ser lido por ferramentas de leitura presente em navegadores, como o Google Text-to-Speech (TTS), presente no Google Chrome. Nos casos em que a leitura da ferramenta não estivesse compreensível, o LAB permitiu que os pesquisadores gravassem áudios a serem tocados no lugar da leitura dos textos.

Os pesquisadores usaram essas possibilidades para desenvolver os jogos da pesquisa, dentre os quais está o jogo analisado neste trabalho. Em sua grande maioria, cada tela contém uma pergunta que é lida para o jogador, em alguns casos as respostas também foram lidas antes que o jogador informasse sua resposta.

3.3 Origem dos dados

Os dados aqui analisados foram gerados durante a pesquisa de Amorim (2018). Seu estudo envolveu 17 escolas da região nordeste do Brasil, sendo 62 turmas e 749 estudantes. A pesquisa seguiu o roteiro abaixo.

Figura 7 – Roteiro da pesquisa de Amorim (2018)



Retirado da pesquisa de Amorim (2018)

Amorim (2018) dividiu igualmente os 749 estudantes em dois grupos: um que não interagiu com os jogos (chamado de grupo de controle) e um que interagiu com os jogos digitais desenvolvidos (chamado de grupo de intervenção). Ao final da pesquisa, constatou que o grupo de intervenção de sua pesquisa desenvolveu mais as habilidades de leitura e escrita do que o grupo de controle.

Cada um dos jogos coletava e enviava para um servidor os dados referentes a interação das crianças com eles. Esta coleta foi possível através da implementação de *event listeners* nos jogos.

A diferença de aprendizado foi aferida através da aplicação de um pré teste e de um pós teste com todos os alunos, para leitura e escrita. Nos testes de leitura, um dos pesquisadores mostrava para a criança uma palavra, que a criança deveria ler e dizer a sua maneira o que estava escrito. Nos testes de escrita, o pesquisador pronunciava uma palavra para a criança que, novamente, deveria escrever em uma folha como ela acreditava que a palavra era escrita.

Estes testes foram desenvolvidos por Pazeto et al. (2012), e foram uma das principais formas de constatar a evolução das crianças no trabalho desenvolvido por Amorim (2018).

3.4 Jogo escolhido

O jogo escolhido para ser analisado neste estudo se chama ”Gol da Aliteração”, e ele possui cinco tipos de tela diferentes: tela de abertura, tela de instruções, telas de mecânica, telas de fim de nível e tela de fim de jogo. Todas as telas do jogo são listadas na tabela abaixo.

Tabela 2 – Ordem das telas do jogo Gol da Aliteração e seus respectivos tipos

Ordem	Identificador da tela	Tipo de tela	Palavra alvo
1	589df5017e917	Abertura	–
2	ce6db391-10ea-4191-a26e-ef6cb610d285	Instruções	–
3	877d2eb4-5ae0-481a-8cb7-1e2c4f207bf0	Mecânica	FACE
4	59826b9c3647c	Mecânica	PIÃO
5	3211c25f-bb7b-4c92-b26d-a11806678005	Mecânica	PIRULITO
6	b48ce9a2-b87f-49d9-b646-e94ac77ca011	Mecânica	CAVALO
7	3706ca23-aaf4-4970-a8d9-d63a39b615a7	Mecânica	RAMO
8	6bc89348-ce92-40bb-8d98-379c4d7727aa	Fim de nível	–
9	89ab600a-a45a-4fc5-9cb5-79ecd355cd18	Mecânica	FORMA
10	c87d5d54-2dd1-4d23-af5d-eafc5c1641cd	Mecânica	BONÉ
11	678ac328-e8b1-433b-b88b-45154136adf6	Mecânica	TRABALHO
12	e939cc86-7868-414f-b0b7-ad4fc834719d	Mecânica	FACA
13	10ccc029-8f76-4c6a-8e5b-5acd1300ae53	Fim de nível	–
14	8712dbe8-1ed0-4eae-a485-781c4a3b870c	Mecânica	PAÇO
15	f4322dfd-0657-44c5-b829-d15c017e919	Mecânica	TESOURO
16	facfa596-1d83-4215-b3ab-643f41583f06	Fim de nível	–
17	398423b8-c5bb-4594-a8e5-fbd4acc6d998	Fim de Jogo	–

O Gol da Aliteração foi escolhido para este trabalho por utilizar somente um tipo de tela de mecânica, ou seja, só existe uma forma de interação entre o usuário e o jogo, mudando apenas o conteúdo desta tela de mecânica, como veremos adiante. Esse fato possibilita fazer mais facilmente a análise de palavras mais relevantes, uma vez que não existe diferença dentre um desafio e outro, somente as palavras que são propostas como desafio.

Outros jogos feitos no trabalho de Amorim (2018) eram mais diretamente relacionados com leitura e escrita, porém as formas que o jogador interagia com o jogo eram diferentes, fato que tornava mais complexa a comparação do conteúdo deles uma vez que podem existir desafios (ou mecânicas) considerados mais fáceis ou mais complexos pelos jogadores.

3.4.1 Tela de Abertura

Trata-se de uma tela com a imagem de uma bola de futebol na rede, além do próprio título e um botão para iniciar o jogo. É a primeira tela exibida ao acessar o jogo, e seu título é lido pelo TTS. A representação gráfica desta tela está representada na figura (8).

Figura 8 – Tela de abertura do jogo ”Gol da Aliteração”

Imagem retirada da plataforma desenvolvida durante a pesquisa de Amorim (2018)

3.4.2 Tela de instruções

É a segunda tela do jogo, possui textos que definem as regras do jogo para a criança: identificar palavras que começam com o mesmo som da palavra alvo da tela de mecânica. A mensagem completa pode ser conferida na Figura (9).

A tela também possui um botão de play que redireciona o jogador para a próxima tela.

Figura 9 – Tela de instruções do jogo ”Gol da Aliteração”

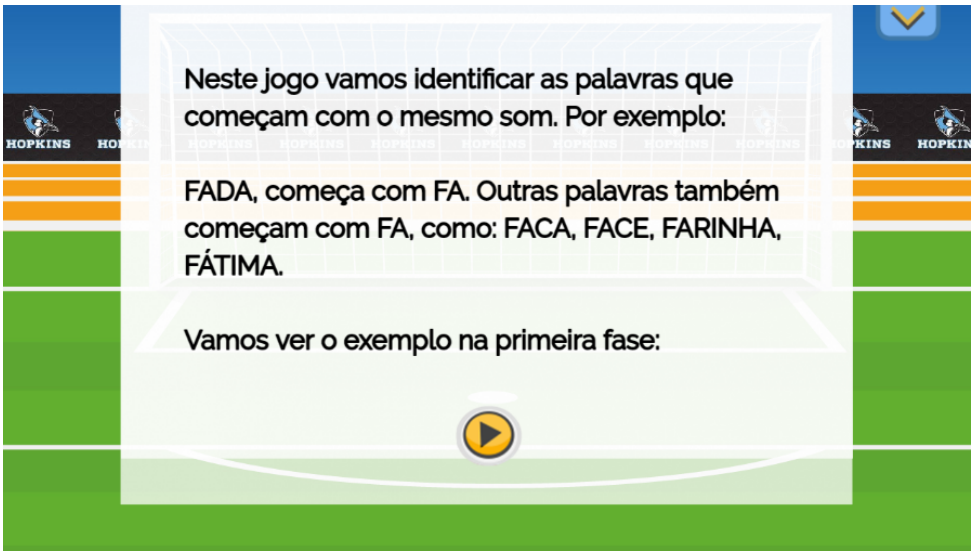


Imagem retirada da plataforma desenvolvida durante a pesquisa de Amorim (2018)

3.4.3 Telas de mecânica

São as telas que definem as perguntas aos jogadores. Nelas sempre estão definidos os seguintes elementos: A palavra alvo, as respostas, o personagem interativo, a bola e o botão de dica. Além do texto da pergunta, que é lido ao iniciar a fase. Essa mecânica foi desenhada para se assemelhar a uma cobrança de pênalti no futebol.

O texto da pergunta segue o seguinte padrão: ”O que começa com o mesmo som da palavra...” e então é lida a palavra alvo. Caso o jogador pressione o botão de dica, o mesmo texto é lido novamente.

As respostas possíveis são sempre imagens, o jogador deve conseguir identificar que palavra representa a imagem e se essa palavra começa com o mesmo som da palavra alvo. Ao pressionar uma resposta, a bola localizada na parte inferior da tela é lançada em direção a resposta selecionada. Caso a resposta seja errada, o personagem interativo faz uma defesa e lança a bola de volta ao seu ponto inicial, e é apresentado um feedback de erro para o jogador. Caso a resposta seja correta, o personagem interativo se lança para outro lado diferente ao que a bola está indo, um

feedback de acerto é apresentado, e o texto ”goooool!” é exibido, passando para a próxima tela do jogo.

As posições das respostas são sempre fixadas nos quatro cantos da barra, e a cada vez que a tela é reiniciada as respostas trocam de lugar entre si de forma a evitar que o jogador apenas decore a posição das respostas. Na tabela abaixo é mostrada todas as palavras alvo de telas de mecânica, bem como quais eram as opções possíveis de resposta.

Tabela 3 – Palavras alvo e respostas possíveis das telas de mecânica

Palavra alvo	Resposta 1	Resposta 2	Resposta 3	Resposta 4
FACE	ELEFANTE	FADA	LÁPIS	APONTADOR
PIÃO	HAMBÚRGUER	PIZZA	MACARRÃO	HOT DOG
PIRULITO	CARRO	ESQUILO	GALINHA	PIPOCA
CAVALO	CARRO	MOTO	URSO	AVIÃO
RAMO	BARCO	CANGURU	RATO	AVIÃO
FORMA	LEÃO	FORMIGA	ELEFANTE	CACHORRO
BONÉ	JACARÉ	HIPOPÓTAMO	PORCO	BOI
TRABALHO	TUCANO	TRAPEZISTA	SOLDADO	AVESTRUZ
FACA	FADA	ELEFANTE	LÁPIS	APONTADOR
PAÇO	CARRO	PATO	QUADRO	LÁPIS
TESOURO	CARTA	CANETA	TESOURA	ZEBRA

A primeira tela de mecânica que segue a tela de instruções retrata um exemplo semelhante ao dado na instrução: A palavra alvo é **FACE**, uma das respostas possíveis é uma figura de uma **Fada** e somente nessa fase existe uma figura apontando para a fada, mostrando ao jogador que é onde ele deve pressionar para responder corretamente.

As telas de mecânica possuem duração máxima de 2 minutos, caso esse tempo seja ultrapassado e o jogador ainda não tenha encontrado a resposta correta, a tela de mecânica é reiniciada. Um exemplo dessa tela pode ser visto na Figura (10).

3.4.4 Tela de fim de nível

É uma tela simples que exibe estrelas, de 1 até 3, para o jogador de acordo com sua performance num determinado agrupamento de telas de mecânica.

Tem três utilidades dentro do jogo: definir níveis dentro de um jogo, dar um feedback visual para a criança sobre seu progresso e faz uma verificação da performance do jogador no nível.

Todas as telas entre telas de fim de nível ou entre o inicio do jogo e uma tela de fim de nível são considerados níveis.

Figura 10 – Exemplo de tela de mecânica do jogo ”Gol da Aliteração”.

Imagem retirada da plataforma desenvolvida durante a pesquisa de Amorim (2018)

Se, por exemplo, o jogador passar por muitas telas do nível sem dar uma resposta ou sem conseguir vencer o desafio proposto no tempo determinado, ela não exibe nenhuma estrela e bloqueia a passagem para o próximo nível, obrigando o jogador à reiniciar o nível jogado. Uma representação dessa tela pode ser vista na Figura (11).

Figura 11 – Uma das telas de fim de fase do jogo ”Gol da Aliteração”.

Imagem retirada da plataforma desenvolvida durante a pesquisa de Amorim (2018)

3.4.5 Tela de fim de jogo

Exibe algumas informações para o jogador ao fim do jogo, como: Quantidade de estrelas conquistadas nas telas de fim de nível, tempo total de jogo, e uma certa quantidade de pontos que o jogador conquistou por seus acertos. Além disso, exibe

uma medalha de acordo com a performance do jogador, dentre os seguintes tipos: ouro, prata e bronze. A Figura (12) mostra um exemplo de tela de fim de jogo.

Figura 12 – Tela de fim de jogo do Gol da Aliteração.

Imagem retirada da plataforma desenvolvida durante a pesquisa de Amorim (2018)

3.5 Eventos produzidos pelo APP

Existem seis eventos que o APP identifica e envia ao servidor, são eles:

1. Lançar: Enviado sempre que o jogador inicia um determinado jogo.
2. Visualizar: Enviado sempre que o jogador visualiza qualquer elemento dentro do jogo. Esse elemento pode ser desde uma imagem na tela até a própria tela em si.
3. Sair: Enviado sempre que o jogador realiza todas as ações necessárias para finalizar uma tela.
4. Passar: Enviado sempre que o jogador seleciona uma das respostas corretas de uma tela.
5. Falhar: Enviado sempre que o jogador seleciona uma das respostas incorretas de uma tela.
6. Pontuar: Enviado sempre que o jogador finaliza um determinado jogo. Nesse evento o aplicativo também envia uma pontuação que o jogador obteve, que é calculada de acordo com as respostas certas e erradas dadas pelo jogador.

Cada um desses eventos tem uma estrutura de detalhamento única, em formato JSON que é salva dentro do banco de dados do LAB. Para a mineração feita nessa pesquisa, somente o evento de lançar não foi usado, uma vez que a quantidade de vezes que o jogador iniciou uma sessão de jogo não faz parte das métricas escolhidas para a análise proposta neste trabalho.

3.6 Detalhes relevantes na estrutura dos eventos

O evento de visualizar informa que elementos foram visualizados, é possível identificar a visualização por tela contando a quantidade desses eventos onde o objeto é a tela em si.

O evento de sair possui o identificador da tela, juntamente com o tempo no formato ISO 8601 que o jogador demorou para sair daquela tela.

Os eventos de passar e falhar indicam quais elementos o usuário selecionou para responder a uma pergunta de uma tela, e o evento de pontuar somente é contado para um determinado jogo. O evento de pontuar foi usado para identificar a quantidade de vezes que um usuário finalizou completamente um determinado jogo.

Um exemplo de uma entrada de log pode ser feita a seguir:

Tabela 4 – Exemplo de entrada de log no banco de dados do LAB

id	guid	verb_id	object_id	actor_id	share_id	parent_id	project_id	timestamp	JSON	ip
3482	31430a5b	1	48995	2732	1120982974	0	267	2019-01-10 19:09:40	{"exemplo":1}	127.0.0.1

Destes, os mais relevantes para a nossa pesquisa são os actor_id's, verb_id's, project_id's e o JSON. O actor_id é o identificador do jogador, o verb_id é a identificação do tipo de evento, o project_id é o identificador do jogo e o JSON contém informações adicionais do evento dependendo do tipo de evento.

3.7 Extração de dados

Foi feito um script em Python para extrair os dados dos logs. O banco de logs dos eventos possuem cerca de 2,5 milhões de entradas.

Esses eventos foram identificados por meio de uma característica inserida propositalmente no sistema: todos os participantes da pesquisa possuíam o identificador da pesquisa no meio dos seus nomes cadastrados internamente, e somente os participantes da pesquisa possuem essa característica. A partir disso, bastou selecionar os eventos de usuários que satisfaziam esses requisitos, e cujo identificador do projeto era o projeto que analisamos neste trabalho, para todos os tipos de evento mencionados, excetuando apenas o de lançar.

Os dados extraídos foram, por tela:

- Quantidade de acertos
- Quantidade de erros
- Quantidade de visualizações
- Tempo médio, em segundos, por tela
- Quantidade de acertos líquidos (acertos - erros)
- Porcentagem de acertos

Além desses, também foi adicionado a quantidade de vezes que o jogador finalizou completamente um jogo.

Os três primeiros dados foram obtidos a partir dos seus respectivos eventos (acerto, erro e visualização). A única ressalva feita foi que as visualizações foram contadas somente se o objeto visualizado fosse a própria tela do jogo. Para os acertos e erros não foi feita nenhuma filtragem, somente a contagem de eventos registrada para cada tela.

O tempo médio foi obtido nos eventos de ”Sair”, todos os eventos com tempo registrado em ISO 8601 foram convertidos para segundos e foi feita a média.

Os acertos líquidos e porcentagem de acertos foram obtidos posteriormente à contabilização dos eventos acima, quando o script já tinha registrado todos os eventos de acertos e erros por tela e por jogador.

Além desses dados, para cada jogador foi adicionado o resultado nos pós testes de escrita ou de leitura. Estes resultados foram coletados durante a pesquisa de Amorim (2018) e estavam separados em planilhas.

A estrutura das colunas segue o seguinte padrão: screen <identificador da tela> <identificador do evento contabilizado>. Os identificadores dos eventos contabilizados são: V para visualização, R para acertos, W para erros, Average Time (in seconds) para o tempo médio, Liquid para acertos líquidos, Right Percent para porcentagem de acertos.

O único dado coletado que não segue o padrão é o de finalizações, ele está no formato: <identificador do jogo> F. Para simplificar a visualização de todas as métricas extraídas por jogador e seus respectivos significados, foi elaborada a Tabela 5.

Tabela 5 – Descrição das métricas extraídas para cada jogador

Métricas	Descrição
internal_id	Identificador interno usado na pesquisa
class_id	Identificador da turma do aluno
R2Score	Indica se o jogador está acima da média no teste de leitura
W2Score	Indica se o jogador está acima da média no teste de escrita
screen d2a5503c-142a-4127-9607-af9a6f6f9a3a V	Quantidade de visualizações para a tela fim de jogo removida
screen 89ab600a-a45a-4fc5-9cb5-79ecd355cd18 V	Quantidade de visualizações para a tela #9 (FORMA)
screen e939cc86-7868-414f-b0b7-ad4fc834719d V	Quantidade de visualizações para a tela #12 (FACA)
screen b48ce9a2-b87f-49d9-b646-e94ac77ca011 V	Quantidade de visualizações para a tela #6 (CAVALO)
screen 589df5017e917 V	Quantidade de visualizações para a tela #1
screen c87d5d54-2dd1-4d23-af5d-eafc5c1641cd V	Quantidade de visualizações para a tela #10 (BONÉ)
screen 3211c25f-bb7b-4c92-b26d-a11806678005 V	Quantidade de visualizações para a tela #5 (PIRULITO)
screen 3706ca23-aaf4-4970-a8d9-d63a39b615a7 V	Quantidade de visualizações para a tela #7 (RAMO)
screen 398423b8-c5bb-4594-a8e5-fbd4acc6d998 V	Quantidade de visualizações para a tela #17
screen ce6db391-10ea-4191-a26e-ef6cb610d285 V	Quantidade de visualizações para a tela #2
screen 8712dbe8-1ed0-4eae-a485-781c4a3b870c V	Quantidade de visualizações para a tela #14 (PAÇO)
screen 877d2eb4-5ae0-481a-8cb7-1e2c4f207bf0 V	Quantidade de visualizações para a tela #3 (FACE)
screen facfa596-1d83-4215-b3ab-643f41583f06 V	Quantidade de visualizações para a tela #16
screen 6bc89348-ce92-40bb-8d98-379c4d7727aa V	Quantidade de visualizações para a tela #8
screen 10ccc029-8f76-4c6a-8e5b-5acd1300ae53 V	Quantidade de visualizações para a tela #13
screen f4322dfd-0657-44c5-b829-d15c017e9196 V	Quantidade de visualizações para a tela #15 (TESOURO)
screen 678ac328-e8b1-433b-b88b-45154136adf6 V	Quantidade de visualizações para a tela #11 (TRABALHO)
screen 59826b9c3647c V	Quantidade de visualizações para a tela #4 (PIÃO)
screen e939cc86-7868-414f-b0b7-ad4fc834719d R	Quantidade de respostas corretas na tela #12 (FACA)
screen 8712dbe8-1ed0-4eae-a485-781c4a3b870c R	Quantidade de respostas corretas na tela #14 (PAÇO)
screen 59826b9c3647c R	Quantidade de respostas corretas na tela #4 (PIÃO)
screen c87d5d54-2dd1-4d23-af5d-eafc5c1641cd R	Quantidade de respostas corretas na tela #10 (BONÉ)
screen 3706ca23-aaf4-4970-a8d9-d63a39b615a7 R	Quantidade de respostas corretas na tela #7 (RAMO)
screen 678ac328-e8b1-433b-b88b-45154136adf6 R	Quantidade de respostas corretas na tela #11 (TRABALHO)
screen 3211c25f-bb7b-4c92-b26d-a11806678005 R	Quantidade de respostas corretas na tela #5 (PIRULITO)
screen b48ce9a2-b87f-49d9-b646-e94ac77ca011 R	Quantidade de respostas corretas na tela #6 (CAVALO)
screen 877d2eb4-5ae0-481a-8cb7-1e2c4f207bf0 R	Quantidade de respostas corretas na tela #3 (FACE)
screen f4322dfd-0657-44c5-b829-d15c017e9196 R	Quantidade de respostas corretas na tela #15 (TESOURO)
screen 89ab600a-a45a-4fc5-9cb5-79ecd355cd18 R	Quantidade de respostas corretas na tela #9 (FORMA)
screen 89ab600a-a45a-4fc5-9cb5-79ecd355cd18 W	Quantidade de respostas incorretas na tela #9 (FORMA)
screen f4322dfd-0657-44c5-b829-d15c017e9196 W	Quantidade de respostas incorretas na tela #15 (TESOURO)
screen 8712dbe8-1ed0-4eae-a485-781c4a3b870c W	Quantidade de respostas incorretas na tela #14 (PAÇO)
screen 877d2eb4-5ae0-481a-8cb7-1e2c4f207bf0 W	Quantidade de respostas incorretas na tela #3 (FACE)
screen 3706ca23-aaf4-4970-a8d9-d63a39b615a7 W	Quantidade de respostas incorretas na tela #7 (RAMO)
screen c87d5d54-2dd1-4d23-af5d-eafc5c1641cd W	Quantidade de respostas incorretas na tela #10 (BONÉ)
screen 3211c25f-bb7b-4c92-b26d-a11806678005 W	Quantidade de respostas incorretas na tela #5 (PIRULITO)
screen b48ce9a2-b87f-49d9-b646-e94ac77ca011 W	Quantidade de respostas incorretas na tela #6 (CAVALO)
screen e939cc86-7868-414f-b0b7-ad4fc834719d W	Quantidade de respostas incorretas na tela #12 (FACA)
screen 59826b9c3647c W	Quantidade de respostas incorretas na tela #4 (PIÃO)
screen 678ac328-e8b1-433b-b88b-45154136adf6 W	Quantidade de respostas incorretas na tela #11 (TRABALHO)
screen 678ac328-e8b1-433b-b88b-45154136adf6 Liquid	Quantidade de (acertos - erros) na tela #11 (TRABALHO)
screen b48ce9a2-b87f-49d9-b646-e94ac77ca011 Liquid	Quantidade de (acertos - erros) na tela #6 (CAVALO)
screen 89ab600a-a45a-4fc5-9cb5-79ecd355cd18 Liquid	Quantidade de (acertos - erros) na tela #9 (FORMA)
screen 3706ca23-aaf4-4970-a8d9-d63a39b615a7 Liquid	Quantidade de (acertos - erros) na tela #7 (RAMO)
screen e939cc86-7868-414f-b0b7-ad4fc834719d Liquid	Quantidade de (acertos - erros) na tela #12 (FACA)
screen 8712dbe8-1ed0-4eae-a485-781c4a3b870c Liquid	Quantidade de (acertos - erros) na tela #14 (PAÇO)
screen 3211c25f-bb7b-4c92-b26d-a11806678005 Liquid	Quantidade de (acertos - erros) na tela #5 (PIRULITO)
screen 59826b9c3647c Liquid	Quantidade de (acertos - erros) na tela #4 (PIÃO)
screen f4322dfd-0657-44c5-b829-d15c017e9196 Liquid	Quantidade de (acertos - erros) na tela #15 (TESOURO)
screen c87d5d54-2dd1-4d23-af5d-eafc5c1641cd Liquid	Quantidade de (acertos - erros) na tela #10 (BONÉ)
screen 877d2eb4-5ae0-481a-8cb7-1e2c4f207bf0 Liquid	Quantidade de (acertos - erros) na tela #3 (FACE)
screen 3706ca23-aaf4-4970-a8d9-d63a39b615a7 Right Percent	Porcentagem de acertos do jogador na tela #7 (RAMO)
screen c87d5d54-2dd1-4d23-af5d-eafc5c1641cd Right Percent	Porcentagem de acertos do jogador na tela #10 (BONÉ)
screen 59826b9c3647c Right Percent	Porcentagem de acertos do jogador na tela #4 (PIÃO)
screen 8712dbe8-1ed0-4eae-a485-781c4a3b870c Right Percent	Porcentagem de acertos do jogador na tela #14 (PAÇO)
screen 877d2eb4-5ae0-481a-8cb7-1e2c4f207bf0 Right Percent	Porcentagem de acertos do jogador na tela #3 (FACE)
screen 89ab600a-a45a-4fc5-9cb5-79ecd355cd18 Right Percent	Porcentagem de acertos do jogador na tela #9 (FORMA)
screen f4322dfd-0657-44c5-b829-d15c017e9196 Right Percent	Porcentagem de acertos do jogador na tela #15 (TESOURO)
screen 3211c25f-bb7b-4c92-b26d-a11806678005 Right Percent	Porcentagem de acertos do jogador na tela #5 (PIRULITO)
screen 678ac328-e8b1-433b-b88b-45154136adf6 Right Percent	Porcentagem de acertos do jogador na tela #11 (TRABALHO)
screen e939cc86-7868-414f-b0b7-ad4fc834719d Right Percent	Porcentagem de acertos do jogador na tela #12 (FACA)
screen b48ce9a2-b87f-49d9-b646-e94ac77ca011 Right Percent	Porcentagem de acertos do jogador na tela #6 (CAVALO)
screen 398423b8-c5bb-4594-a8e5-fbd4acc6d998 Average Time (in seconds)	Tempo médio de jogo na tela #17
screen c87d5d54-2dd1-4d23-af5d-eafc5c1641cd Average Time (in seconds)	Tempo médio de jogo na tela #10 (BONÉ)
screen 678ac328-e8b1-433b-b88b-45154136adf6 Average Time (in seconds)	Tempo médio de jogo na tela #11 (TRABALHO)
screen 89ab600a-a45a-4fc5-9cb5-79ecd355cd18 Average Time (in seconds)	Tempo médio de jogo na tela #9 (FORMA)
screen e939cc86-7868-414f-b0b7-ad4fc834719d Average Time (in seconds)	Tempo médio de jogo na tela #12 (FACA)
screen 877d2eb4-5ae0-481a-8cb7-1e2c4f207bf0 Average Time (in seconds)	Tempo médio de jogo na tela #3 (FACE)
screen ce6db391-10ea-4191-a26e-ef6cb610d285 Average Time (in seconds)	Tempo médio de jogo na tela #2
screen 3706ca23-aaf4-4970-a8d9-d63a39b615a7 Average Time (in seconds)	Tempo médio de jogo na tela #7 (RAMO)
screen 10ccc029-8f76-4c6a-8e5b-5acd1300ae53 Average Time (in seconds)	Tempo médio de jogo na tela #13
screen 3211c25f-bb7b-4c92-b26d-a11806678005 Average Time (in seconds)	Tempo médio de jogo na tela #5 (PIRULITO)
screen facfa596-1d83-4215-b3ab-643f41583f06 Average Time (in seconds)	Tempo médio de jogo na tela #16
screen 6bc89348-ce92-40bb-8d98-379c4d7727aa Average Time (in seconds)	Tempo médio de jogo na tela #8
screen f4322dfd-0657-44c5-b829-d15c017e9196 Average Time (in seconds)	Tempo médio de jogo na tela #15 (TESOURO)
screen 8712dbe8-1ed0-4eae-a485-781c4a3b870c Average Time (in seconds)	Tempo médio de jogo na tela #14 (PAÇO)
screen d2a5503c-142a-4127-9607-af9a6f6f9a3a Average Time (in seconds)	Tempo médio de jogo na tela fim de jogo removida
screen b48ce9a2-b87f-49d9-b646-e94ac77ca011 Average Time (in seconds)	Tempo médio de jogo na tela #6 (CAVALO)
screen 589df5017e917 Average Time (in seconds)	Tempo médio de jogo na tela #1
screen 59826b9c3647c Average Time (in seconds)	Tempo médio de jogo na tela #4 (PIÃO)
1795 F	Quantidade de vezes que o jogador finalizou o jogo

Foram desconsiderados todos os eventos que não cumpriram os seguintes requisitos:

- Caso seu identificador do sistema não estivesse entre os identificadores dos jogadores da pesquisa
- Caso seu identificador da pesquisa não estivesse entre aqueles que tiveram seus resultados de pré e pós teste coletados
- Para os eventos de visualização, caso o objeto visualizado não fosse uma tela

Ao final do processo, foram gerados dois arquivos .csv, onde a única mudança foi a variável alvo, um deles tendo como variável alvo a nota no pós teste de escrita e outra contendo a nota no pós teste de leitura. Cada arquivo final tem dados de 318 alunos, além de 84 características deles, dentre as quais está o resultado no pós-teste de leitura e de escrita.

3.8 Mineração dos dados e validação

Para sumarizar melhor os processo de metodologia como um todo, foi elaborada a figura abaixo.

Figura 13 – Metodologia de extração dos dados e treinamento dos classificadores.

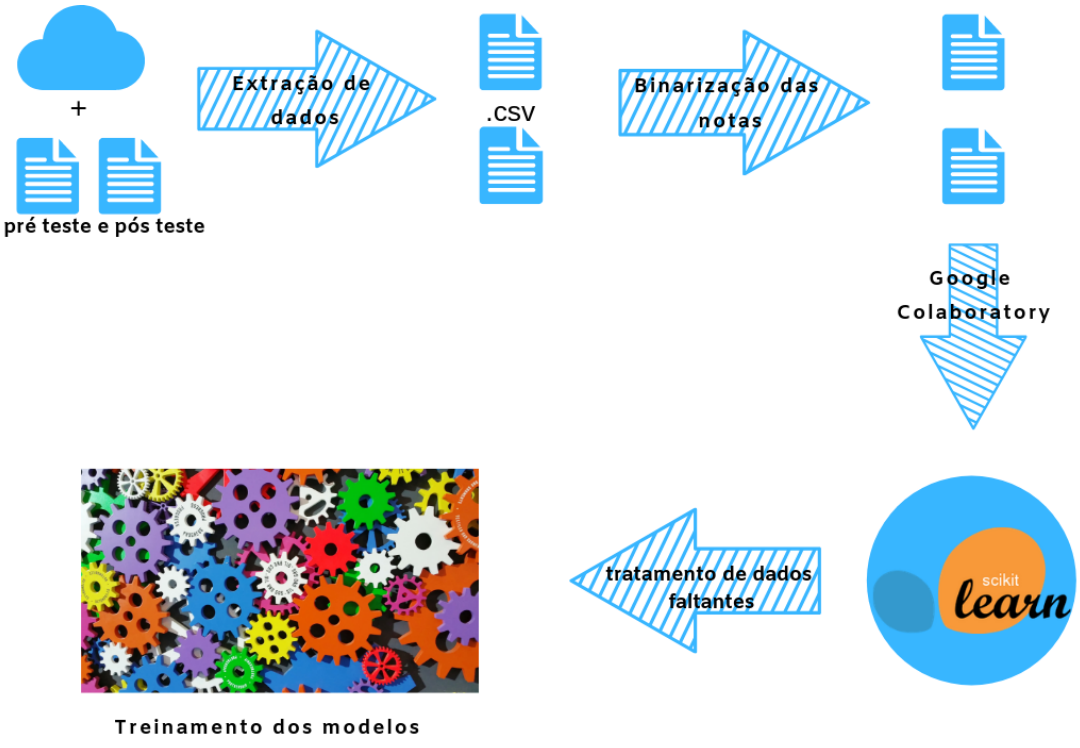


Imagem elaborada pelo autor do trabalho

Esta seção adiciona maiores detalhes relacionados ao processo de binarização das notas e treinamento dos classificadores, bem como justifica a escolha de cada um dos algoritmos usados no trabalho.

3.8.1 Pré-Processamento

Para os dois arquivos .csv gerados, foi feita uma binarização do resultado do pós teste. O pós teste usado neste trabalho foi desenvolvido e aplicado no sudeste do Brasil por Pazeto et al. (2012), que também definiu médias de pontuação esperadas para crianças de 4 anos na região sudeste do Brasil. No trabalho de Amorim (2018) foi aplicado o mesmo teste no nordeste do país obtendo as seguintes médias de pontuação para estudante de 4 anos de idade: 19.84 para leitura e 23.57 para escrita no grupo de controle de sua pesquisa, ou seja, o grupo que não passou pela intervenção dos jogos. Ainda segundo Amorim (2018) não existe um teste de leitura e escrita padrão. E essas foram as referências encontradas e usadas para este trabalho no que se refere a aplicação dos testes desenvolvido por Pazeto et al. (2012).

Foram encontradas dois identificadores de tela que não estavam presentes no jogo atualmente, e fazendo uma pesquisa no banco de dados identificou-se que apesar de terem identificadores diferentes elas correspondem respectivamente à uma tela de mecânica com a palavra alvo TESOURO e uma tela de fim de jogo que foi removida, os dados de referentes a estas duas telas foi mantido entre as métricas.

Portanto, adotamos neste trabalho como pontuações esperadas de leitura e escrita para crianças de 4 anos os valores de 20 e 24, respectivamente. A nossa variável alvo foi binarizada de forma que assumisse o valor 1 se a criança obteve uma pontuação acima do esperado e 0 se estiver abaixo.

Também foi feito o preenchimento dos dados faltantes com o valor zero. Essa decisão foi tomada porque as características que temos são todas númericas, e pelo fato do valor zero representar bem a ausência de tais dados, como visualização, acerto em uma tela, ou erro.

3.8.2 Escolha dos algoritmos

O Naive Bayes foi escolhido por estar bastante presente em trabalhos relacionados, além de, segundo Kotsiantis (2001), baseado em trabalhos de Domingos e Pazzani (1997), mesmo assumindo inicialmente que as variáveis são completamente independentes, o algoritmo de Naive Bayes consegue resultados tão bons ou até melhores que algoritmos mais sofisticados em comparações de larga escala. Nesse estudo é usada a implementação GaussianNB do Scikit Learn.

O algoritmo de Regressão Logística foi selecionado devido a um dos objetivos

da pesquisa, que é encontrar quais dentre as interações coletadas dos usuários são as mais relevantes ou impactantes no resultado do pós teste, para que sejam feitas discussões do ponto de vista educacional a partir disso. O algoritmo de Regressão Logística permite encontrar facilmente quais são as características mais impactantes na sua predição, tornando esse objetivo possível. No estudo é usada a classe LogisticRegression do Scikit Learn.

O algoritmo SVM foi escolhido também por pesquisas na literatura. Segundo Kotsiantis (2001), a complexidade do modelo de uma SVM não é afetada pela quantidade de características dos dados de treinamento, dado que o número de vetores suporte selecionados pelo algoritmo é usualmente pequeno. Por essa razão, as SVMs são consideradas adequadas para tarefas de treinamento onde o número de características é grande em relação ao número de instâncias de treinamento. O que acaba sendo o nosso caso uma vez que temos relativamente poucos exemplos e várias características deles. Para o experimento usamos a implementação SVC (C-Support Vector Classification) do Scikit Learn.

3.8.3 Treinamento dos modelos

O treinamento foi feito na ferramenta *Google Colaboratory*[1] e os modelos escolhidos são implementados e disponibilizados pela biblioteca Python chamada *Scikit Learn (SL)*[2], acessível dentro do Colaboratory.

Para avaliar a performance dos modelos, foi usada a acurácia como métrica e o método *Cross Validation*. A quantidade de *folds* escolhida foi 10, por ser o valor mais comumente usado em trabalhos que aplicam a técnica de validação cruzada. As métricas apresentadas nos resultados são as médias dos 10 testes realizados com cada modelo.

Para o modelo de regressão logística, a biblioteca do SL permite selecionar qual o *Solver* a ser usado, que segundo a *Documentação oficial do Scikit Learn*[3] é um algoritmo usado internamente para melhorar o desempenho do modelo de acordo com certas características dos dados usados no treinamento.

Já na implementação do SVC, é possível escolher qual o *kernel* a ser utilizado. O *kernel* no SVC é o algoritmo responsável por encontrar as margens e o hiperplano ótimo.

Devido à possibilidade de fazer essas escolhas que são impactantes na performance final dos dois algoritmos, usamos 4 tipos diferentes de *solver* que o scikit

[1] https://colab.research.google.com/
[2] https://scikit-learn.org/
[3] https://scikit-learn.org/stable/documentation.html

learn disponibiliza: *lbfgs, liblinear, sag e saga*[4] . Além de 3 tipos diferentes de *kernel* disponíveis: *sigmoid, rbf e linear*[5]

Além disso, também foi feita a análise de componentes principais, usando a biblioteca PCA do SL, com dois objetivos: verificar se o uso de componentes principais afeta positivamente a performance dos modelos e selecionar quais características mais explicam a variância dos dados, sendo assim considerados os mais relevantes.

Decidimos usar 40 componentes principais, porque foi observada a possibilidade de ter cerca de 95% da variância dos dados com essa quantidade de componentes. Todos os testes usando PCA consideram o uso de 40 componentes principais.

Para cada *solver* e cada *kernel*, foi feita a validação cruzada sem o uso da análise de componentes principais e com ela.

Por fim, a análise de características mais relevantes foi feita com o PCA e com o *solver* de regressão logística que teve melhor performance.

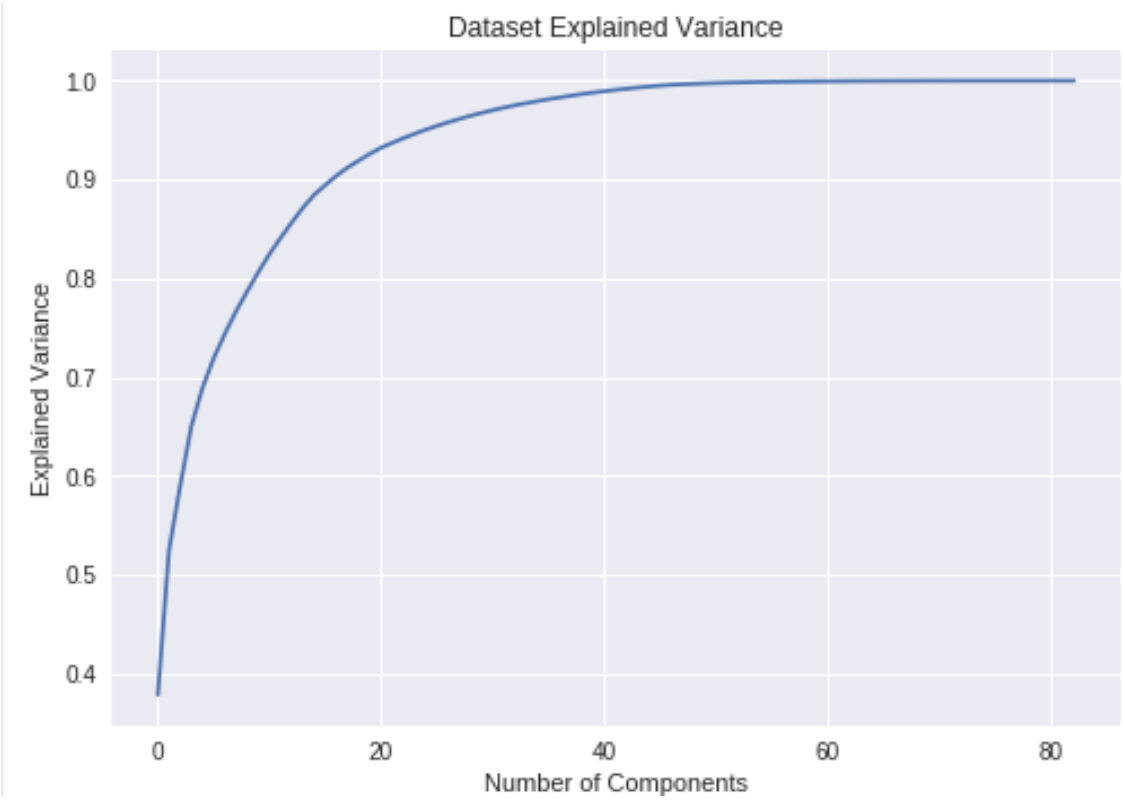
[4] https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
[5] https://scikit-learn.org/stable/modules/svm.html#svm-classification.

4 Resultados e Discussões

Essa seção descreve os resultados obtidos no estudo, além de discussões acerca de tais resultados.

Para a escolha da quantidade de componentes principais, foi feita uma normalização dos dados usando a biblioteca *MinMaxScaler*, definindo um range de zero a um e montando um gráfico de variância explicada por quantidade de componentes que pode ser conferido a seguir.

Figura 14 – Variância explicada por quantidade de componentes principais



Elaborada pelo autor do trabalho.

A partir do gráfico apresentado na Figura (14), foi tomada a decisão de usar 40 componentes principais, o que diminuiria a quantidade de atributos em mais da metade dos 83 atributos originais além de conseguir manter uma variância de cerca de acuna de 95% dos dados.

4.0.1 Testes de escrita

Aqui são apresentados aqui o desempenho de cada algoritmo para escrita. Para cada algoritmo, foi elaborada uma tabela que apresenta os resultados para cada variação do mesmo: para o SVM, é mostrada a performance quando se varia o kernel do algoritmo e para a LR é mostrada a performance quando se muda o tipo de *solver* usado para treinar o modelo. No caso do modelo NB, foram usadas as configurações default da implementação GaussianNB. Abaixo de cada tabela é feita uma discussão sobre os resultados encontrados e ao final da discussão apresentamos um gráfico que facilita a visualização dos resultados discutidos.

Para o algoritmo de regressão logística e seus diferentes *solvers*, foram encontrados os seguintes resultados para o teste de escrita.

Tabela 6 – Resultados para o modelo LR nos testes de escrita

(a) Resultados sem PCA

solver	média	desvio padrão
libfgs	0.536	0.09
liblinear	0.535	0.09
sag	0.602	0.06
saga	0.587	0.08

(b) Resultados com PCA

solver	média	desvio padrão
libfgs	0.558	0.07
liblinear	0.561	0.07
sag	0.583	0.09
saga	0.589	0.11

Para os *solvers* sag, o uso do PCA se mostrou prejudicial, já para o saga, libfgs e liblinear houve ganho na performance dos modelos pelo uso da análise de componentes principais. A figura abaixo facilita a compreensão desses resultados.

Figura 15 – Representação gráfica dos resultados da Regressão Logística

Elaborada pelo autor do trabalho.

A melhor performance do modelo de regressão logística foi com o uso do *solver* sag, sem o PCA, com cerca de 60% de taxa de acerto, medido através da acurácia.

Para o algoritmo SVM, foram encontrados os seguintes resultados.

Tabela 7 – Resultados para o modelo de SVM nos testes de escrita

(a) Resultados sem PCA

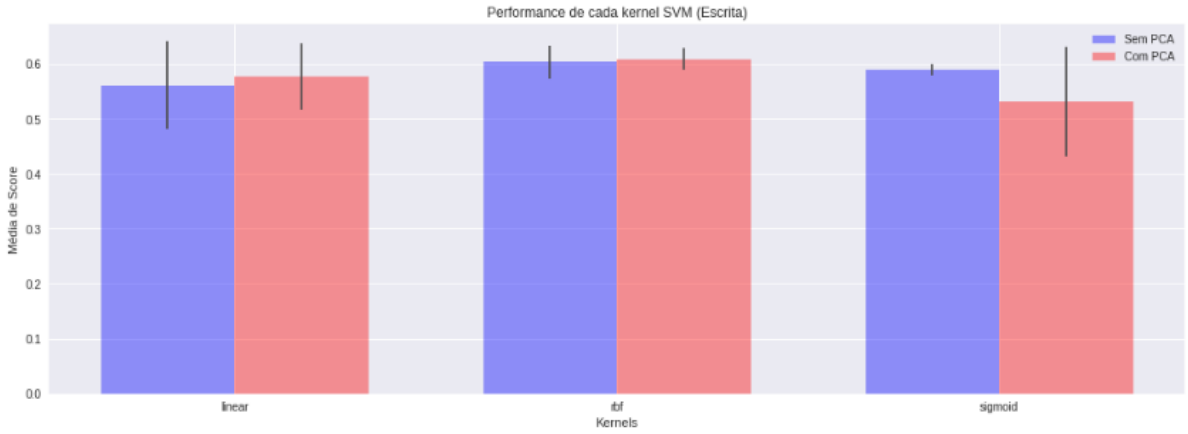
kernel	média	desvio padrão
linear	0.561	0.08
rbf	0.603	0.03
sigmoid	0.59	0.01

(b) Resultados com PCA

kernel	média	desvio padrão
linear	0.577	0.06
rbf	0.609	0.02
sigmoid	0.531	0.1

Para os kernels de SVM o uso de PCA mostrou-se benéfico para o kernel linear e rbf, mas prejudicou a performance do kernel sigmoid. A melhor performance do modelo SVM foi usando componentes principais e o kernel rbf, também com cerca de 60% de taxa de acerto.

Figura 16 – Resultados do modelo SVM para o teste de escrita



Elaborada pelo autor do trabalho.

Tal como o *solver* sag no caso da regressão logística, o uso do pca prejudicou tanto a performance quanto o desvio padrão do kernel sigmoid.

E para o algoritmo GaussianNB, em suas configurações padrão, os achados foram os seguintes.

Tabela 8 – Resultados para o modelo de NB nos testes de escrita

(a) Resultados sem PCA

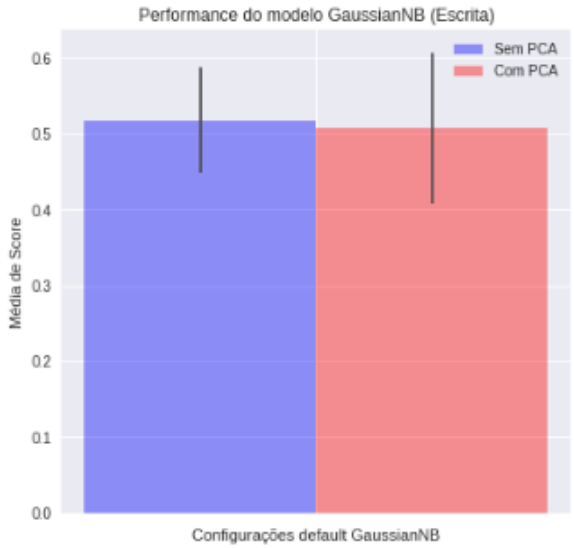
média	desvio padrão
0.517	0.07

(b) Resultados com PCA

média	desvio padrão
0.507	0.1

Dentre os modelos testados, o GaussianNB foi o que obteve a pior performance para escrita, tendo médias próximas de 51% antes do PCA e diminuindo ainda mais com o PCA, caindo 1%. O desvio padrão nos dois casos é alto.

Figura 17 – Resultados do modelo Naive Bayes para os testes de escrita



Elaborada pelo autor do trabalho.

Com esses resultados, pode-se dizer que o uso do modelo de NB, através da implementação GaussianNB, não vale a pena quando se objetiva desenvolver um algoritmo que possa chegar a substituir o pós teste no futuro. Uma performance próxima de 50% é bastante próxima a uma escolha aleatória, sendo quase equivalente a jogar uma moeda e decidir ao acaso.

4.0.2 Testes de leitura

Aqui são apresentados os resultados dos mesmos algoritmos para os testes de leitura, a estrutura segue o padrão anterior: Apresentamos uma tabela com os resultados, é feita uma descrição e discussão dos resultados e é apresentado um gráfico para facilitar a visualização de tais resultados para cada algoritmo.

Tabela 9 – Resultados para o modelo de LR nos testes de leitura

(a) Resultados sem PCA

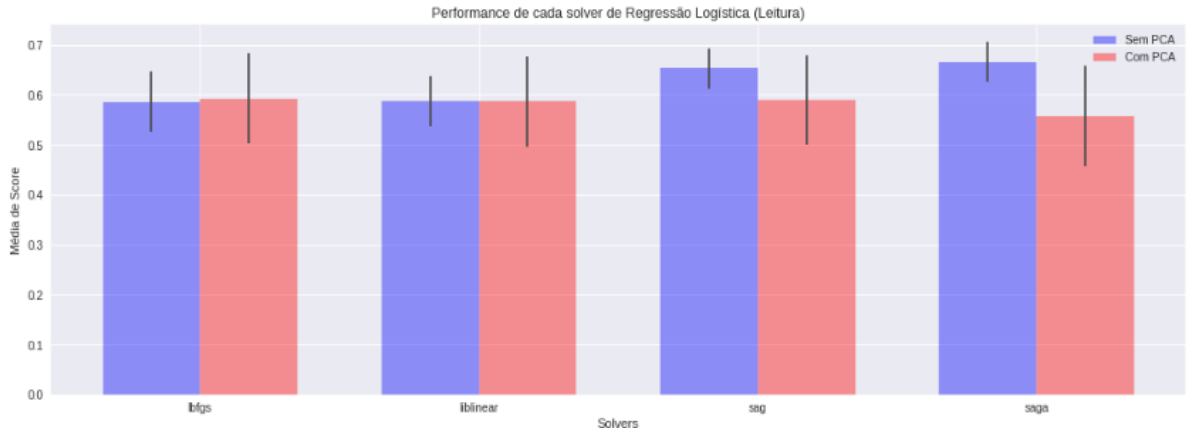
solver	média	desvio padrão
libfgs	0.586	0.06
liblinear	0.587	0.05
sag	0.653	0.04
saga	0.666	0.04

(b) Resultados com PCA

solver	média	desvio padrão
libfgs	0.593	0.09
liblinear	0.587	0.09
sag	0.59	0.09
saga	0.558	0.1

O comportamento é semelhante ao dos testes de escrita, no entando o *solver* saga se comporta melhor que os outros sem o uso do pca (66% de taxa de acerto) e se torna o pior *solver* considerando o uso de componentes principais, mantendo seu significativo aumento de desvio padrão pós pca. O gráfico abaixo sumariza os resultados apresentados na tabela.

Figura 18 – Resultados da Regressão Logística para o teste de leitura



Elaborada pelo autor do trabalho.

O *solver* sag apresenta um comportamento semelhante ao saga. O *solver* liblinear também não apresentou melhora usando componentes principais e o libfgs mostrou desempenho levemente superior usando PCA.

Para o algoritmo de SVM usando cada kernel escolhido, temos os seguintes resultados.

Tabela 10 – Resultados para o modelo de SVM nos testes de leitura

(a) Resultados sem PCA

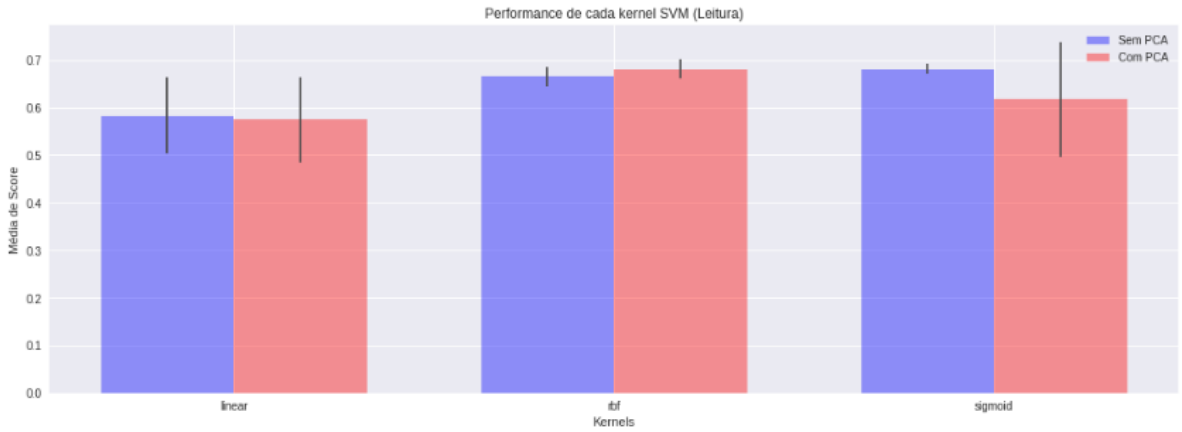
solver	média	desvio padrão
linear	0.583	0.08
rbf	0.665	0.02
sigmoid	0.681	0.01

(b) Resultados com PCA

solver	média	desvio padrão
linear	0.574	0.09
rbf	0.681	0.02
sigmoid	0.617	0.12

O melhor kernel SVM foi o sigmoid sem o uso de componentes principais. Considerando os componentes principais a melhor performance foi a do kernel rbf, tendo apresentado uma média igual ao kernel sigmoid sem PCA. Para proporcionar uma melhor visualização, os dados foram colocados no gráfico abaixo.

Figura 19 – Resultados do modelo SVM para o teste de leitura



Elaborada pelo autor do trabalho.

Somente o kernel rbf demonstrou melhora de desempenho usando componentes principais enquanto o kernel sigmoid manteve sua queda de performance identificada também nos testes de escrita quando se usa a análise de componentes principais.

Nos experimentos usando o algoritmo GaussianNB para leitura, observamos os seguintes resultados.

Tabela 11 – Resultados para o modelo de NB nos testes de leitura

(a) Resultados sem PCA

média	desvio padrão
0.472	0.06

(b) Resultados com PCA

média	desvio padrão
0.602	0.11

O algoritmo GaussianNB manteve-se como o modelo de pior performance do nosso experimento, com resultados bastante próximos aos testes de escrita. Porém para leitura, apresentou um ganho médio um pouco mais significativo.

4.1 Comparativo com o Estado da Arte

Nesta seção é feita uma comparação entre os desempenhos encontrados de cada algoritmo neste trabalho com o de outros trabalhos encontrados na literatura que usaram os mesmos algoritmos para o contexto de jogos. Dentre os trabalhos relacionados citados, o que mais se assemelha a este é o de Fernández (2017), que analisou um jogo que foi desenvolvido e aplicado para crianças um pouco mais velhas, entre 12 e 14 anos para ensinar técnicas de primeiros socorros. A metodologia aplicada foi bastante semelhante a do trabalho de Amorim (2018), tendo um pré teste, a interação com o jogo educacional e por fim a aplicação do pós teste.

Figura 20 – Resultados do modelo Naive Bayes para os testes de leitura

Elaborada pelo autor do trabalho.

Nos resultados deste trabalho, visivelmente o algoritmo NB estava um nível abaixo dos outros dois. O que não foi o caso para o trabalho de Fernández (2017), que encontrou resultados para o algoritmo de NB considerados bons o suficiente para cogitar a possibilidade de retirar o pós teste e usar somente o jogo como critério decisório para o aprendizado de técnicas de primeiros socorros.

Para o uso de regressão logística, neste trabalho a melhor acurácia encontrada foi de cerca de 66% no teste de leitura usando o *solver* saga. O que também não foi o caso no trabalho de Fernández (2017), que encontrou também taxas de predição consideradas altas para a LR. Não podemos comparar diretamente os valores achados uma vez que as métricas usadas nos dois trabalhos são diferentes.

Também foi feita uma análise de resultados do trabalho de Keshtkar et al. (2014), que aplicou algoritmos de SVM e NB para um problema diferente: a detecção de perfis comportamentais a partir da interação de jogadores com um jogo. É interessante notar que tal qual este trabalho, nos resultados obtidos por Keshtkar et al. (2014) os algoritimos de SVM tiveram performances visivelmente superiores aos de NB em termos de acurácia. Os algoritmos de NB deste trabalho tiveram uma acurácia máxima de 60%, obtida no teste de leitura usando componentes principais. Keshtkar et al. (2014) obteve resultados de 65% na aplicação do mesmo modelo em seu problema. Já para os algoritmos de SVM, os melhores resultados deste trabalho chegaram a 68% de acurácia, enquanto nos de Keshtkar et al. (2014) chegaram aos maiores valores de sua pesquisa: 84% de acurácia.

Cabe ressaltar algumas diferenças na natureza dos trabalhos. Além da própria temática e da idade dos jogadores serem bastante diferentes, o jogo analisado neste

trabalho faz parte de um conjunto de 20 jogos aplicados a crianças de 4 anos de idade enquanto no trabalho de Fernández (2017) só foi feita a interação com um jogo. Aqui nós buscamos encontrar qual a contribuição de um dos jogos nos resultados do pós teste.

Já no trabalho de Keshtkar et al. (2014) não houve um pré e um pós teste, e sim feitas análises manuais nas frases digitadas por uma certa quantidade de estudantes aleatoriamente e os resultados dessas análises alimentaram os modelos de previsão.

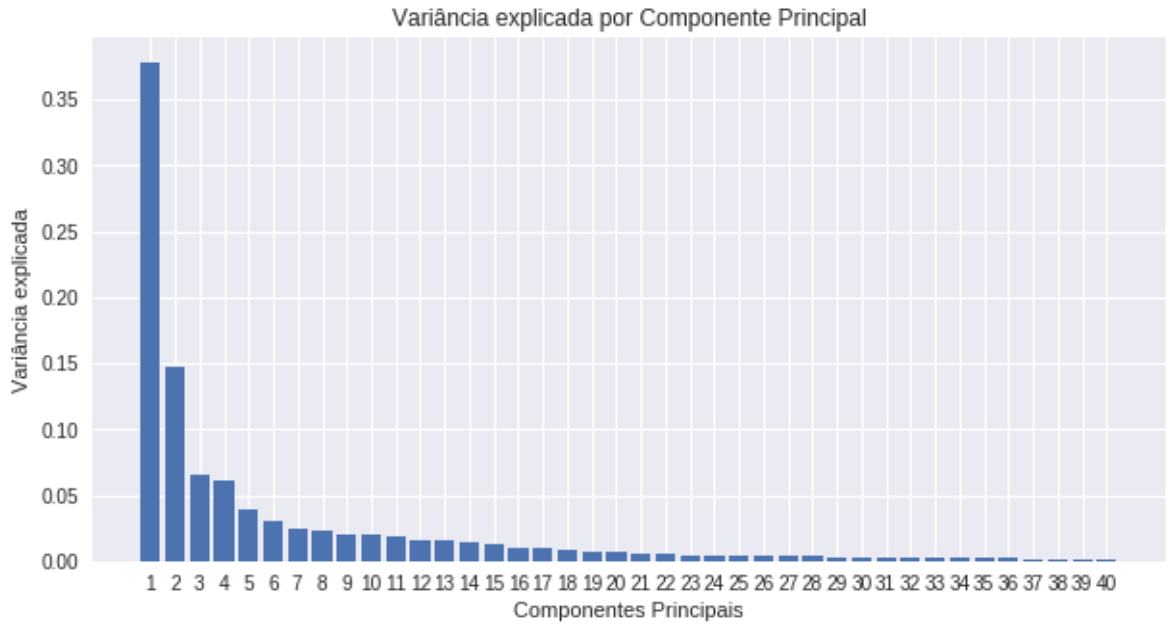
Outra diferença importante em relação à pesquisa de Fernández (2017) é a forma como os jogos foram aplicados. Em sua pesquisa, Amorim (2018) incentivou os jogadores a interagir com o jogo educacional em duplas porque esse fator poderia influenciar positivamente no aprendizado das crianças, enquanto na de Fernández (2017) as interações com os jogos foram individuais.

4.1.1 Componentes mais relevantes

A determinação dos componentes mais relevantes foi feita a partir de três métodos: A análise de componentes principais, o modelo de Regressão Logística usando o *solver* sag para os testes de escrita e o mesmo modelo com *solver* saga para os testes de leitura.

A partir do PCA, foi elaborado um gráfico para demonstrar quais dos 40 componentes principais encontrados são os que representam maior variância desses dados, e ele pode ser visto abaixo.

Figura 21 – Gráfico de variância por componente principal



Elaborada pelo autor do trabalho.

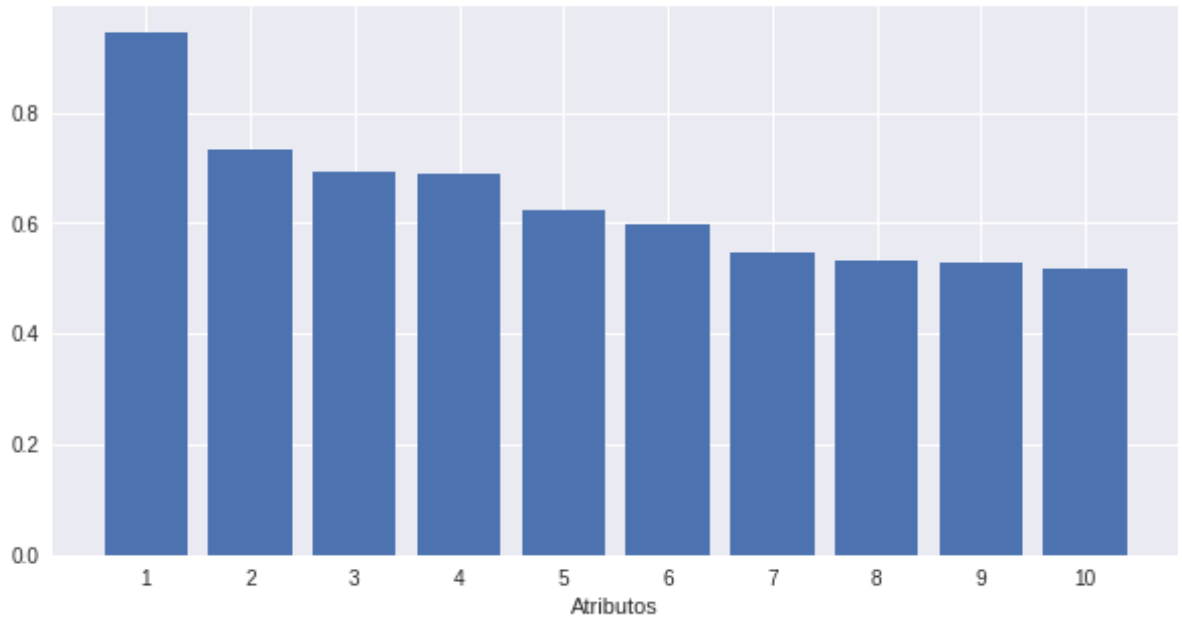
A partir do gráfico, temos que os quatro primeiros componentes principais explicam aproximadamente 70% da variância dos dados. A partir desses componentes, identificamos a variância de cada atributo inicial para os quatro primeiros componentes, selecionando os dez melhores atributos e chegamos aos seguintes resultados.

Tabela 12 – Identificação dos atributos da Figura (21)

Atributo do gráfico	Nome da métrica
1	Identificador da Turma
2	Porcentagem de acertos do jogador na tela #14 (PAÇO)
3	Identificador interno do jogador
4	Porcentagem de acertos na tela #15 (TESOURO)
5	Porcentagem de acertos do jogador na tela #12 (FACA)
6	Porcentagem de acertos do jogador na tela #6 (CAVALO)
7	Quantidade de respostas incorretas na tela #6 (CAVALO)
8	Porcentagem de acertos do jogador na tela #7 (RAMO)
9	Porcentagem de acertos do jogador na tela #11 (TRABALHO)
10	Porcentagem de acertos do jogador na tela #4 (PIÃO)

É possível verificar uma forte influência do ambiente escolar nos resultados, uma vez que o identificador da turma e o identificador do estudante aparecem entre os três valores mais relevantes. Alunos pertencentes a uma mesma escola e turma tem identificadores de valores próximos, portanto é compreensível que os dois apareçam juntos no caso de um dos fatores relevantes nos resultados ser o ambiente escolar. Mais detalhes podem ser verificados na Figura (22).

Figura 22 – Variância dos quatro melhores componentes principais por atributo inicial



Elaborada pelo autor do trabalho.

A grande maioria dos outros atributos são porcentagens de acertos em telas, tendo somente uma outra métrica de erros dentre os atributos mais relevantes para o PCA. Somente uma tela tem duas métricas dentre as mais relevantes, que é a tela #6, que tem CAVALO como palavra alvo.

Para a avaliação feita a partir do treinamento dos modelos, foi escolhido para o teste de escrita o *solver* sag e para o teste de leitura o *solver* saga, por terem sido os *solvers* de LR com melhor performance desconsiderando o uso da análise de componentes principais.

A partir do modelo treinado com o *solver* sag, foram coletados os dez atributos que mais influenciavam o modelo a classificar o jogador como acima da média nos testes de escrita, observamos os seguintes resultados.

Tabela 13 – Identificação dos atributos da Figura 21

Atributo do gráfico	Nome da métrica
1	Quantidade de (acertos - erros) na tela #3 (FACE)
2	Quantidade de (acertos - erros) na tela #6 (CAVALO)
3	Tempo médio de jogo na tela #7 (RAMO)
4	Tempo médio de jogo na tela #5 (PIRULITO)
5	Tempo médio de jogo na tela #2
6	Tempo médio de jogo na tela #14 (PAÇO)
7	Quantidade de (acertos - erros) na tela #9 (FORMA)
8	Quantidade de (acertos - erros) na tela #12 (FACA)
9	Quantidade de (acertos - erros) na tela #5 (PIRULITO)
10	Quantidade de (acertos - erros) na tela #10 (BONÉ)

A interação mais relevante na LR nos testes de escrita foram os acertos líquidos, que é o resultado do total de acertos do jogador menos os erros que ele cometeu na tela. A segunda foi o tempo médio que o jogador levou, em segundos, para finalizar o desafio proposto pela tela. Também é possível observar que existe um certo destaque para os três primeiros atributos, pois há uma queda perceptível comparado ao quarto atributo em diante. É possível verificar essa diferença na Figura 21, logo abaixo.

Figura 23 – Resultados de coeficiente por atributo para a LR nos testes de escrita

Elaborada pelo autor do trabalho.

Esses resultados confirmam os achados de Amorim (2018) em seu trabalho, uma vez que ele descreve os acertos líquidos como a métrica que mais explica a diferença na pontuação entre o pré teste e o pós teste de um jogador. Essa métrica

ter tido mais destaque nos testes de escrita pode ser um indicativo de que crianças com uma habilidade de leitura mais desenvolvida tendem a ter uma performance mais destacada também nos testes de escrita.

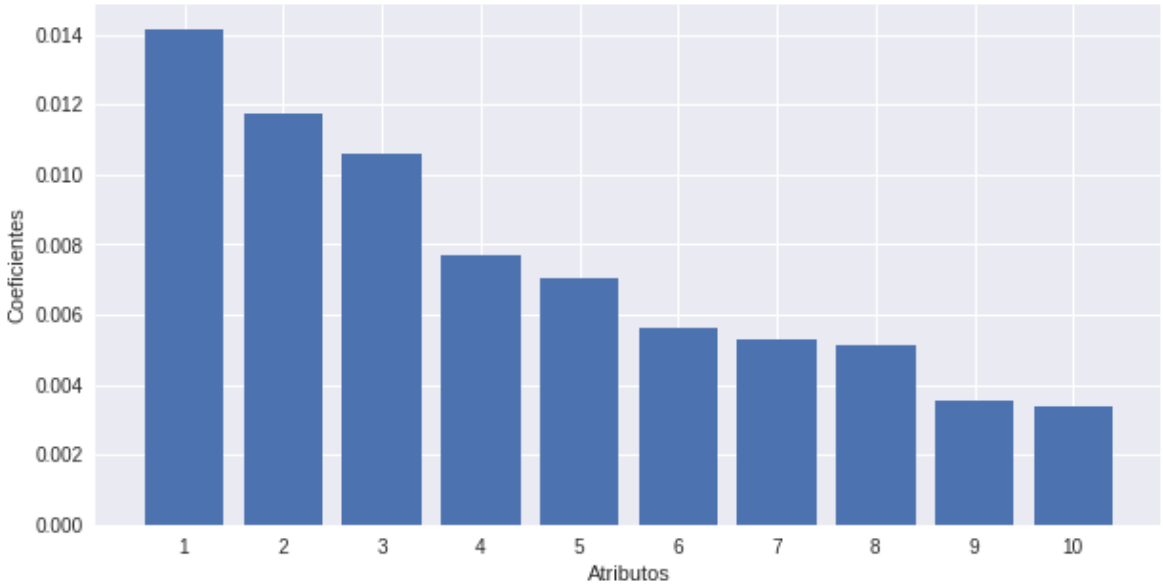
Para os testes de leitura, os resultados da LR com o *solver* saga foram os seguintes.

Tabela 14 – Identificação dos atributos da Figura (23)

Atributo do gráfico	Nome da métrica
1	Tempo médio de jogo na tela #7 (RAMO)
2	Tempo médio de jogo na tela #14 (PAÇO)
3	Tempo médio de jogo na tela #10 (BONÉ)
4	Tempo médio de jogo na tela #4 (PIÃO)
5	Tempo médio de jogo na tela #9 (FORMA)
6	Quantidade de (acertos - erros) na tela #3 (FACE)
7	Quantidade de (acertos - erros) na tela #6 (CAVALO)
8	Tempo médio de jogo na tela #12 (FACA)
9	Quantidade de respostas incorretas na tela #5 (PIRULITO)
10	Quantidade de (acertos - erros) na tela #15 (TESOURO)

Aqui as métricas de acertos líquidos e tempo médio também aparecem bastante, mas o tempo médio na tela se destaca mais que os acertos líquidos. Tais resultados levantam a possibilidade de que, no caso da leitura, crianças que estão se esforçando mais para ler e compreender a pergunta alvo tendem a ter resultados melhores no pós teste de leitura. A Figura 22 mostra as diferenças na influência de cada atributo na decisão dos testes de leitura.

Figura 24 – Resultados de coeficiente por atributo para a LR nos testes de leitura



Elaborada pelo autor do trabalho.

Para sumarizar os resultados encontrados, as tabelas abaixo descrevem quais as telas mais citadas pelas três análises feitas para detectar os componentes mais relevantes e também mostra quais as interações com maior frequência.

Tabela 15 – Sumarização das principais características encontradas

(a) Telas que tiveram mais interações citadas como relevantes

Descrição da Tela	Quantidade
Tela #4 (CAVALO)	4
Tela #12 (FACA)	3
Tela #7 (RAMO)	3
Tela #5 (PIRULITO)	3
Tela #14 (PAÇO)	3
Tela #3 (FACE)	2
Tela #9 (FORMA)	2
Tela #10 (BONÉ)	2
Tela #4 (PIÃO)	2
Tela #15 (TESOURO)	2
Identificador da Turma	1
Tela #2	1
Identificador interno do jogador	1
Tela #11 (TRABALHO)	1

(b) Interações mais citadas como relevantes

Tipo de Interação	Quantidade
Tempo médio (em segundos)	10
Acertos líquidos (total acertos - total erros)	9
Porcentagem de Acertos	7
Erros	2

A partir dessa sumarização, chama atenção o fato da tela com a palavra alvo CAVALO ter mais citações para influência positiva. O jogo avaliado aqui propõe como desafio encontrar palavras que iniciam com o mesmo som, portanto a sílaba CA pode ter uma influência destacada quando comparada às outras. Para buscar explicações disso, foi feita uma análise das palavra usadas no pós teste do trabalho de Amorim (2018).

No teste de escrita, três das dez palavras presentes na prova iniciavam com o fonema C: Casa, Cabide e Chuveiro. Dentre elas, em duas a sílaba CA está presente, além de ser a primeira sílaba da palavra. Verificou-se também a presença da palavra Pila no pós teste, o que pode explicar a relevância da palavra PIRULITO, que teve duas métricas citadas no modelo para o teste de escrita.

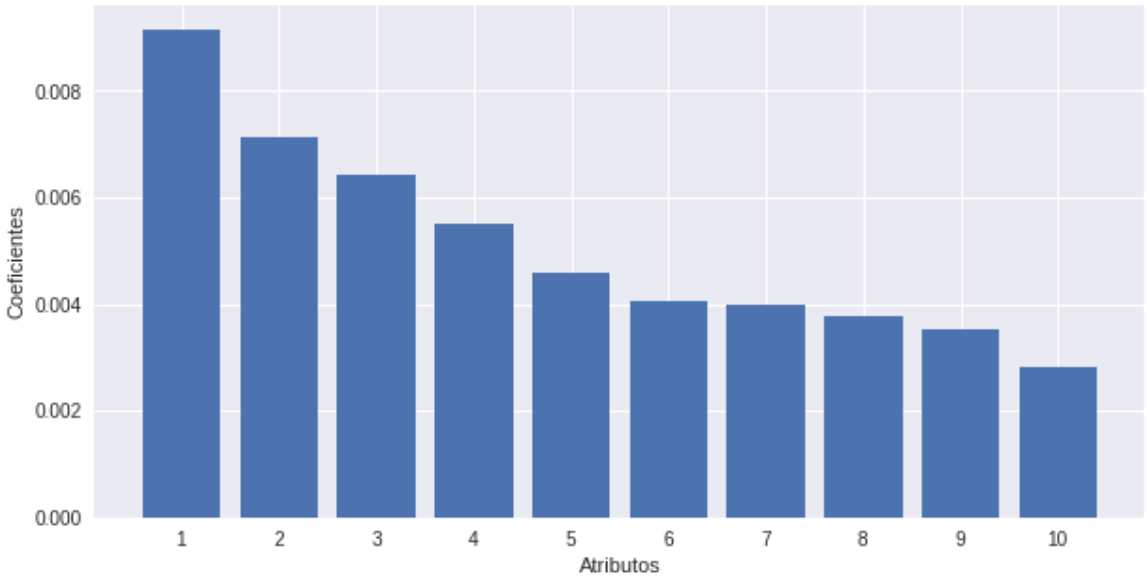
Usando ainda os coeficientes da LR é possível verificar quais os atributos que influenciam negativamente a classificação, ou seja, aqueles que quanto mais crescem, mais tendem a fazer o modelo decidir que o jogador estará abaixo da média no teste. Para os testes de leitura foram encontrados os seguintes resultados.

Atributo	Descrição
1	Tempo médio de jogo na tela #17 fim de jogo
2	Tempo médio de jogo na tela #16 fim de nível
3	Tempo médio de jogo na tela #3 (FACE)
4	Tempo médio de jogo na tela #1 de abertura
5	Tempo médio de jogo na tela #15 (TESOURO)
6	Tempo médio de jogo na tela de fim de jogo
7	Quantidade de respostas incorretas na tela #6 (CAVALO)
8	Quantidade de respostas incorretas na tela #10 (BONÉ)
9	Quantidade de visualizações para a tela #1 de abertura
10	Quantidade de (acertos - erros) na tela #4 (PIÃO)

Tabela 16 – Identificação dos atributos da Figura (25)

Aqui fica mais perceptível a presença de telas que não são de mecânica, além de métricas como as de erros em telas. A presença de telas de fim de nível pode significar que por vezes o jogador não conseguiu atingir a performance mínima para passar para outros níveis do jogo. A tela de fim de nível não tem seu conteúdo lido pelo TTS e se comporta diferente quando o usuário não consegue a pontuação mínima, exibindo um botão para que o jogador reinicie o nível que teve pontuação baixa, o que leva a crer que nesses casos os jogadores não souberam como reagir para tentar novamente. No gráfico a seguir é possível ver os valores absolutos dessas métricas.

Figura 25 – Impacto negativo de coeficientes por atributo nos testes de leitura



Elaborada pelo autor do trabalho.

Também vale destacar a presença da métrica de erros na palavra CAVALO, cujo acerto foi verificado como fator positivo para o modelo na análise anterior.

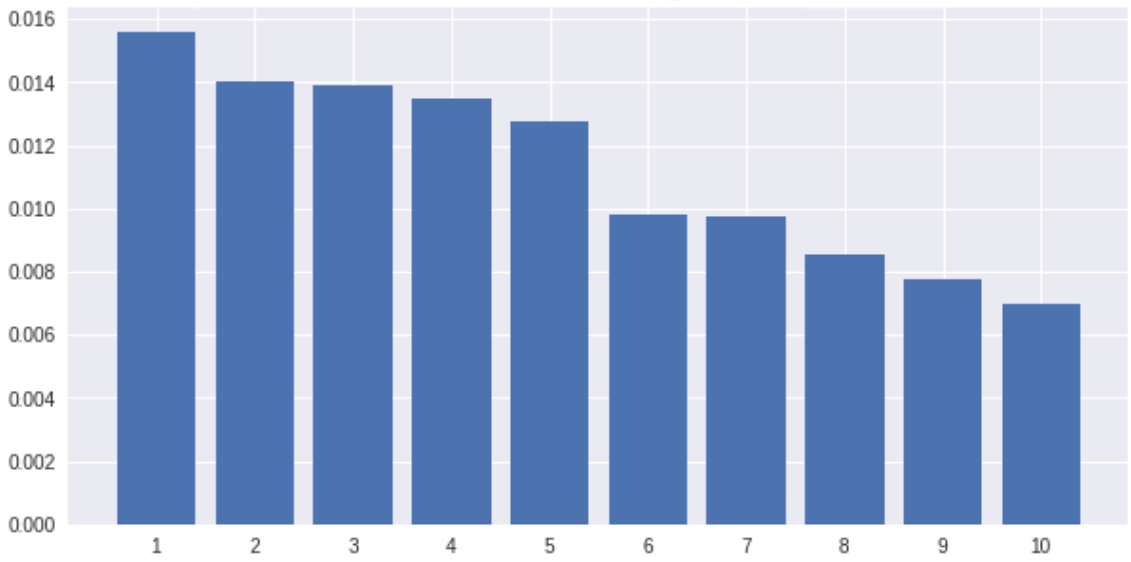
Para os impactos negativos nos resultados de escrita, encontramos os seguintes resultados.

Tabela 17 – Identificação dos atributos da Figura (26)

Atributo	Descrição
1	Tempo médio de jogo na tela #16 fim de nível
2	Quantidade de respostas incorretas na tela #6 (CAVALO)
3	Tempo médio de jogo na tela #8 fim de nível
4	Tempo médio de jogo na tela #17 fim de jogo
5	Quantidade de visualizações para a tela #1
6	Quantidade de respostas incorretas na tela #5 (PIRULITO)
7	Tempo médio de jogo na tela fim de jogo
8	Quantidade de respostas incorretas na tela #3 (FACE)
9	Quantidade de respostas incorretas na tela #4 (PIÃO)
10	Quantidade de visualizações para a tela #2

Aqui é possível ver uma predominância mais clara de métricas de tempo médio em telas de fim de jogo e de fase, além das métricas de respostas incorretas em telas. Esses resultados reforçam ainda mais as suspeitas de problemas enfrentados pelos jogadores durante a interação em telas de fim de nível e de jogo, além de oferecer mais indícios de que errar mais palavras é um fator preditor relevante para uma performance abaixo da média.

Figura 26 – Impacto negativo de coeficientes por atributo nos testes de escrita



Elaborada pelo autor do trabalho.

Os erros na tela da palavra CAVALO e PIRULITO são ainda mais relevantes para o teste de escrita, o que reforça as suspeitas levantadas na análise dos fatores que influenciam positivamente, já que é no teste de escrita que encontramos mais palavras que apresentam a sílaba CA e uma que apresenta a sílaba PI.

5 Conclusão e Trabalhos Futuros

Para o objetivo específico 1, este trabalho conseguiu estabelecer uma metodologia capaz de ser aplicada para os outros 19 jogos que fizeram parte da pesquisa de Amorim (2018). Fica confirmada a possibilidade de extração e análise dos dados de qualquer um desses jogos a partir dos arquivos de log armazenados.

Para o segundo objetivo específico deste trabalho, a performance dos algoritmos escolhidos foi considerada insatisfatória. Os melhores resultados foram 60,9% para os testes de escrita utilizando o modelo de SVM configurada para usar o *kernel rbf* com PCA e 68,1% também para o modelo de SVM configurada para usar ou o *kernel sigmoid* sem PCA, ou o *kernel rbf* com PCA. Tais valores não são suficientes para fazer uma substituição do pós teste.

Para o último dos objetivos específicos deste trabalho, o de fazer uma análise dos atributos mais relevantes, para o modelo LR encontramos mais traços que reforçam as suspeitas levantadas no trabalho de Amorim (2018), como a relevância positiva da métrica de acertos líquidos, resultante da subtração dos erros pelos acertos. Além de uma discussão inicial relativa à quantidade de métricas predominantes dentre as que impactam positivamente o resultado do modelo em cada um dos testes: Acertos líquidos para escrita e Tempo médio para leitura.

Como afirmado anteriormente, a presença da elevada quantidade de métricas de acertos líquidos nos testes de escrita pode indicar que as crianças que já possuem uma habilidade de leitura mais desenvolvida que as outras e que, portanto, acertam mais perguntas do que erram tendem a serem classificadas como acima da média pelo modelo, enquanto o alto número de métricas de tempo médio nos testes de leitura pode indicar que um bom fator preditor de boa performance de leitura possa ser a atenção empregada pela criança em resolver cada desafio proposto na tela, o que a faz levar mais tempo para responder o desafio.

Para as palavras em específico, foi encontrada uma relação relevante nos testes de escrita. Duas palavras chamaram atenção pela quantidade de métricas relacionadas a elas, e foram encontradas palavras no teste de escrita que são correspondentes por terem as primeiras sílabas iguais.

Já no caso das métricas que fazem o modelo classificar a criança como abaixo da média considerada adequada, chamou atenção a presença da métrica de tempo médio em telas que não são de mecânica e que não apresentam desafio ao jogador. Nesses casos, o jogador pode ter demorado por não ter compreendido bem o que deveria fazer quando estava naquela tela, o que consequentemente pode ter tirado

sua atenção da atividade. Outro problema encontrado é que as telas de fim de nível e de jogo não são lidas pelo TTS, o que reforça a hipótese do jogador ter perdido o foco por não ter compreendido qual a ação a ser executada ao se deparar com ela.

Tais hipóteses levantadas podem ser consideradas na produção dos próximos jogos, portanto o objetivo de achar interações relevantes é considerado cumprido.

O objetivo geral deste trabalho é considerado cumprido, uma vez que relações entre as interações dos jogadores foram descobertas e analisadas. As performances dos modelos, apesar de insatisfatórias para o objetivo específico 2, denotam que existe relação entre o resultado no pós teste e a interação com o jogo educacional analisado.

Em relação à performance insatisfatória dos algoritmos, existem limitações nos dados usados neste trabalho que precisam ser consideradas.

1. A amostra que tivemos acesso neste estudo tinha cerca de 300 entradas. Essa quantidade pode ser considerada pequena para a tarefa de classificação.

2. Os estudantes jogavam cada jogo em duplas, mas as diagnoses utilizadas para aferição do desempenho em leitura e escrita foram feitas individualmente. Assim, os dados coletados pelos jogos refletem a performance de dois estudantes trabalhando juntos, mas a diagnose reflete um desenho potencialmente menor, o de cada estudante trabalhando individualmente.

3. O resultado observado no teste é fruto de uma interação dos jogadores com 20 jogos diferentes, e não somente do que foi analisado aqui. O que encontramos foi o que pode ser somente a contribuição deste jogo para o resultado final. Os dados que temos refletem somente parte do impacto da interação das crianças com os jogos, e não o todo.

4. O jogo escolhido para ser estudado neste trabalho não tratava especificamente de leitura e escrita como um todo, e sim da habilidade de aliteração. O desafio proposto ao jogador era o de identificar palavras que iniciavam com o som de uma mesma sílaba, e essa é somente uma das habilidades necessárias para saber ler e escrever.

Para trabalhos futuros, é sugerida a análise dos outros jogos que foram aplicados durante o trabalho de Amorim (2018), não mais de forma individual mas de um conjunto de métricas de dois ou mais jogos para verificar se há diferença sugestiva na performance dos classificadores para substituir futuramente o pós teste. Outros caminhos são os de desenvolver diferentes jogos que levem em consideração as hipóteses levantadas aqui sobre os impactos positivos e negativos das interações encontradas.

Por fim, é sugerida também a condução de uma pesquisa que use tecnologia de *eye-tracking*, que possibilite o monitoramento de onde os jogadores estão olhando ao interagirem com o jogo digital. A análise de tais dados poderia adicionar informações ainda mais relevantes ao debate levantado sobre a possibilidade dos jogadores terem perdido o foco em telas que não apresentam algum desafio.

Referências

AMORIM, A. N. G. F. *The use of digital games by kindergarten students to enhance early literacy skills*. 2018. <https://drive.google.com/open?id=1LfYCXsg3a91UCmF1UvPAlrVbv67ExMkG>. Citado 18 vezes nas páginas 7, 14, 29, 31, 32, 33, 34, 35, 37, 38, 41, 44, 52, 54, 57, 60, 64 e 65.

ARAUJO, W. O. D.; COELHO, C. J. Análise de componentes principais (pca). *University Center of Anápolis, Annapolis Google Scholar*, 2009. Citado na página 27.

BARR, M. Video games can develop graduate skills in higher education students: A randomised trial. *Computers & Education*, Elsevier, v. 113, p. 86–97, 2017. Citado na página 14.

BBC, D. F. *Salas lotadas e pouca valorização: ranking global mostra desgaste dos professores no Brasil*. 2018. <https://www.bbc.com/portuguese/brasil-44436608>. Citado na página 12.

DOMINGOS, P.; PAZZANI, M. On the optimality of the simple bayesian classifier under zero-one loss. *Machine learning*, Springer, v. 29, n. 2-3, p. 103–130, 1997. Citado na página 44.

DWECK, D. C. M. C. S. *Self-theories: Their impact on competence motivation and acquisition.* 2005. Citado na página 14.

FABRICATORE, C. Learning and videogames: An unexploited synergy. 2000. Citado na página 15.

FERNÁNDEZ, C. A. *Applying data mining techniques to Game Learning Analytics*. [S.l.: s.n.], 2017. Citado 4 vezes nas páginas 29, 52, 53 e 54.

GRANIC, I.; LOBEL, A.; ENGELS, R. C. The benefits of playing video games. *American psychologist*, American Psychological Association, v. 69, n. 1, p. 66, 2014. Citado 3 vezes nas páginas 12, 14 e 15.

IBGE, A. de Notícias do. *Analfabetismo cai em 2017, mas segue acima da meta para 2015*. 2018. <https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/21255-analfabetismo-cai-em-2017-mas-segue-acima-da-meta-para-2015>. Citado na página 12.

INEP, A. de Comunicação Social do. *Censos Educacionais do Inep revelam mais de 2,5 milhões de professores no Brasil*. 2018. <http://portal.inep.gov.br/artigo/-/asset_publisher/B4AQV9zFY7Bv/content/censos-educacionais-do-inep-revelam-mais-de-2-5-milhoes-de-professores-no-brasil/21206>. Citado na página 12.

KESHTKAR, F. et al. Using data mining techniques to detect the personality of players in an educational game. In: *Educational Data Mining*. [S.l.]: Springer, 2014. p. 125–150. Citado 3 vezes nas páginas 29, 53 e 54.

KOTSIANTIS, S. B. *Supervised Machine Learning: A Review of Classification Techniques*. 2001. Citado 5 vezes nas páginas 21, 22, 23, 44 e 45.

LOH, C. S.; SHENG, Y.; IFENTHALER, D. Serious games analytics: Theoretical framework. In: *Serious games analytics*. [S.l.]: Springer, 2015. p. 3–29. Citado na página 16.

LOH, Y. S. C. S. *Measuring Expert Performance for Serious Games Analytics: From Data to Insights*. 2015. Citado 2 vezes nas páginas 17 e 18.

LORENA, A. C. P. L. F. d. C. A. C. *Introdução às Máquinas de Vetores Suporte (Support Vector Machines)*. 2003. Citado 4 vezes nas páginas 21, 22, 23 e 24.

MINUSSI, J. A.; DAMACENA, C.; JR, W. L. N. Um modelo de previsão de solvência utilizando regressão logística. *Revista de Administração Contemporânea*, SciELO Brasil, v. 6, n. 3, p. 109–128, 2002. Citado na página 24.

MOURA, D.; NASR, M. S. el; SHAW, C. D. Visualizing and understanding players' behavior in video games: Discovering patterns and supporting aggregation and comparison. In: *Proceedings of the 2011 ACM SIGGRAPH Symposium on Video Games*. New York, NY, USA: ACM, 2011. (Sandbox '11), p. 11–15. ISBN 978-1-4503-0775-8. Disponível em: <http://doi.acm.org/10.1145/2018556.2018559>. Citado na página 18.

MUELLER, L. M. J. P. *RESORTING TO CROSS-VALIDATION IN MACHINE LEARNING*. 2016. <https://www.dummies.com/programming/big-data/data-science/resorting-cross-validation-machine-learning/>. Citado na página 26.

PAZETO, T. d. C. B. et al. Avaliação de funções executivas, linguagem oral e escrita em pré-escolares. Universidade Presbiteriana Mackenzie, 2012. Citado 2 vezes nas páginas 33 e 44.

RISH, I. *An empirical study of the naive Bayes classifier*. 2001. Citado na página 21.

SCIENTISTS, F. of A. *Summit on educational games: Harnessing the power of video games for learning*. [S.l.]: Author Washington, DC, 2006. Citado na página 16.

SILVA, L. A. da; PERES, S. M.; BOSCARIOLI, C. *Introdução à mineração de dados: com aplicações em R*. [S.l.]: Elsevier Brasil, 2017. Citado 4 vezes nas páginas 20, 23, 26 e 27.

SLIMANI, A. et al. Learning analytics through serious games: Data mining algorithms for performance measurement and improvement purposes. *International Journal of Emerging Technologies in Learning (iJET)*, International Association of Online Engineering, v. 13, n. 1, p. 46–64, 2018. Citado na página 29.

SMITH, S. P.; BLACKMORE, K.; NESBITT, K. A meta-analysis of data collection in serious games research. In: *Serious Games Analytics*. [S.l.]: Springer, 2015. p. 31–55. Citado na página 19.

SNOW, L. K. A. E. L.; MCNAMARA, D. S. *The Dynamical Analysis of Log Data Within Educational Games*. 2015. Citado 2 vezes nas páginas 18 e 19.

SUSI, T.; JOHANNESSON, M.; BACKLUND, P. *Serious games: An overview*. [S.l.]: Institutionen för kommunikation och information, 2007. Citado na página 15.

SWAMINATHAN, S. *Logistic Regression — Detailed Overview*. 2018. <https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc>. Citado na página 25.

UNICEF, E. do Rádio pela Infância da. *Rádio pela infância - Boletim de Setembro de 2009*. 2009. <https://www.unicef.org/brazil/pt/rpi_setembro2009.pdf>. Citado na página 12.

VARELLA, C. A. A. *Análise de Componentes Principais*. 2008. Citado na página 27.

VOGEL, J. J. et al. Computer gaming and interactive simulations for learning: A meta-analysis. *Journal of Educational Computing Research*, v. 34, n. 3, p. 229–243, 2006. Disponível em: <https://doi.org/10.2190/FLHV-K4WA-WPVQ-H0YM>. Citado na página 12.