UNIVERSIDADE FEDERAL DO RIO GRANDE
CENTRO DE CIÊNCIAS COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO
CURSO DE MESTRADO EM ENGENHARIA DE COMPUTAÇÃO

Dissertação de Mestrado

O Uso de Aprendizado de Máquina para Identificar Alunos em Risco de Evasão na Educação a Distância

Myke Morais de Oliveira

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal do Rio Grande, como requisito parcial para a obtenção do grau de Mestre em Engenharia de Computação

Orientadora: Prof. Dr. Danúbia Bueno Espíndola
Co-orientador: Prof. Dr. Marcelo Rita Pias

Rio Grande, 2020

Ficha Catalográfica

O48u Oliveira, Myke Morais de.
O uso de aprendizado de máquina para identificar alunos em risco de evasão na Educação a Distância / Myke Morais de Oliveira. – 2020.
69 f.

Dissertação (mestrado) – Universidade Federal do Rio Grande – FURG, Programa de Pós-Graduação em Computação, Rio Grande/RS, 2020.
Orientadora: Dra. Danúbia Bueno Espíndola.
Coorientador: Dr. Marcelo Rita Pias.

1. Modelo Preditivo 2. Aprendizado de Máquina 3. *Deep Learning* 4. Evasão na Educação a Distância 5. Dados do Moodle I. Espíndola, Danúbia Bueno II. Pias, Marcelo Rita III. Título.

CDU 37.018.43:004

Catalogação na Fonte: Bibliotecário José Paulo dos Santos CRB 10/2344

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE
CENTRO DE CIÊNCIAS COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO
CURSO DE MESTRADO EM ENGENHARIA DE COMPUTAÇÃO

DISSERTAÇÃO DE MESTRADO

O uso de aprendizado de máquina para identificar alunos em risco de evasão na educação à distância

Myke Morais de Oliveira

Banca examinadora:

Prof. Dr. Tiago Thompsem Primo

Prof. Dr. Eduardo Nunes Borges

Prof. Dr. Rodrigo Andrade de Bem

Profª. Drª. Danúbia Bueno Espíndola
Orientadora

Prof. Dr. Marcelo Rita Pias
Coorientador

AGRADECIMENTOS

Tenho muito a agradecer por estar concluindo mais uma etapa da minha vida. Agradeço aos meus pais Cledinei e Neuza, que sempre me proporcionaram tudo o que eu precisei, e não mediram esforços para me auxiliar em tudo o que eu precisava. Agradeço a minha orientadora Danúbia, que me auxiliou durante todo o meu Mestrado. Agradeço ao meu coorientador Marcelo por ter me ajudado a escolher o tema da minha dissertação, me coorientando junto com minha orientadora durante todo o desenvolvimento do projeto. Agradeço a Regina que foi uma orientadora para mim, sempre me incentivando a fazer pesquisas científicas, escrever artigos, apresentar trabalhos. Muito obrigado mesmo. Aos meus amigos e colegas de mestrado, especialmente Leo, que vem me motivando e colaborando em trabalhos desde minha graduação. À Universidade de Federal do Rio Grande, por ter me acolhido e dado todo o suporte necessário durante todo o meu mestrado no Centro de Ciências Computacionais. Agradeço a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES por ter financiado meu trabalho de mestrado, o que foi muito importante para a minha permanência na cidade de Rio Grande.

RESUMO

OLIVEIRA, Myke Morais de. **O Uso de Aprendizado de Máquina para Identificar Alunos em Risco de Evasão na Educação a Distância**. 2020. 70 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal do Rio Grande, Rio Grande.

Identificar alunos em risco de evasão tornou-se um importante objeto de pesquisa, visto que é um problema que ocasiona danos sociais, acadêmicos e financeiros. Diante desse cenário, surgiram diversas pesquisas na literatura que propõem soluções para ajudar na identificação prévia de estudantes em risco de evasão. Muitas delas utilizam algoritmos convencionais de aprendizado de máquina sobre dados educacionais, com o objetivo de detectar padrões que denunciem o perfil de um aluno que evade. No entanto, existem maneiras mais avançadas na atualidade, que poderiam explorar melhor, em termos de desempenho e qualidade, os dados educacionais para gerar um modelo preditivo mais robusto, como *Deep Learning*. Assim, nesta dissertação, apresentam-se duas abordagens para ajudar no processo de identificação prévia de alunos em risco de evasão. Na primeira abordagem, oito algoritmos convencionais de aprendizado de máquina foram utilizados para explorar o *dataset* que foi construído com dados da plataforma Moodle de dois cursos a distância, e avalia-lo no processo de modelagem preditiva. Essa abordagem resultou em dois experimentos que foram essenciais para a implementação da segunda abordagem, em que utilizou-se *Deep Learning* para a implementação de uma *Recurrent Neural Network* que, com células de LSTM em sua arquitetura, tem uma grande capacidade de aprendizagem. Com esta abordagem, realizou-se um terceiro experimento, em que pode ser observado o potencial de uma LSTM para lidar com a natureza dos dados dessa pesquisa.

Palavras-chave: Modelo Preditivo, Aprendizado de Máquina, *Deep Learning*, Evasão na Educação a Distância, Dados do Moodle.

ABSTRACT

OLIVEIRA, Myke Morais de. **Using Machine Learning to Identify Students at Dropout Risk in Distance Education**. 2020. 70 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal do Rio Grande, Rio Grande.

Identifying students at dropout risk has become an important research object since it is a problem that causes social, academic and financial damage. Given this scenario, several researches have been developed in the literature proposing solutions to support the early identification of students at dropout risk. Many of them use conventional machine learning algorithms on educational data to detect patterns that can reveal an at-risk student profile. However, there are more advanced mechanisms in the present moment that could better exploit, in terms of performance and quality, educational data to generate a more robust predictive model, like Deep Learning. Thus, in this dissertation, two approaches are presented to help in the process of early identification of students at dropout risk. In the first one, eight conventional machine learning algorithms were used to explore the dataset that was built with data from the Moodle platform of two distance postgraduate programs and to evaluate it in the predictive modeling process. This approach resulted in two experiments that were essential for the implementation of the second approach, in which Deep Learning was used to implement a Recurrent Neural Network that, with LSTM cells in its architecture, has a great capacity for learning. Therefore, a third experiment was carried out with the second approach, in which the potential of an LSTM can be observed to deal with the nature of the data in this research.

Keywords: Predictive Model, Machine Learning, Deep Learning, Dropout in Distance Education, Moodle Data.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AB	*Adaptive Boosting (AdaBoost)*
AVA	Ambiente Virtual de Aprendizagem
AUROC	*Area Under the ROC Curve*
EaD	Educação a Distância
ET	*Extremely Randomized Trees (Extra Trees)*
FN	Falso Negativo
FP	Falso Positivo
GB	*Gradient Boosting*
KNN	*k-Nearest Neighbors*
LR	*Logistic Regression*
LSTM	Rede de Memória de Longo Prazo (*Long-Short Term Memory*)
MOOC	*Massive Open Online Course*
NB	*Naive Bayes*
RF	*Random Forest*
RNN	Rede Neural Recorrente (*Recurrent Neural Network*)
ROC	*Receiver Operating Characteristic*
RSL	Revisão Sistemática da Literatura
SBIE	Simpósio Brasileiro de Informática na Educação
SVC	*C-Support Vector Classification*
TIC	Tecnologia de Informação e Comunicação
TICEDU	Tecnologias de Informação e Comunicação na Educação
TFP	Taxa de Falso Positivos
TVP	Taxa de Verdadeiro Positivos
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo

SUMÁRIO

1 INTRODUÇÃO

Neste Capítulo são apresentados o contexto e motivação para a elaboração da dissertação (Seção 1.1), o problema de pesquisa (Seção 1.2), as justificativas do ponto de vista científico, social e econômico (Seção 1.3), os objetivos geral e específicos (Seção 1.4) e, por fim, a organização do trabalho (Seção 1.5).

1.1 Contexto e Motivação

A educação a distância (EaD) é uma modalidade de ensino em que alunos e professores estão separados, física ou temporalmente e, à vista disso, torna-se fundamental o uso de meios e tecnologias de informação e comunicação (TIC) para intermediar a comunicação entre ambas as partes. Todavia, tal comunicação parece não ser suficiente, uma vez que há um grande problema de evasão nessa modalidade de ensino (CENSO, 2018).

A EaD enfrenta um grande problema de evasão, o que tem resultado em baixos índices de conclusão nos cursos desta modalidade (CENSO, 2018). Grande parte das universidades que fizeram parte da pesquisa do Censo relataram taxas de evasão entre 11% e 25%. À vista disso, as instituições de ensino que proporcionam a modalidade a distância precisam reconhecer e confrontar este problema de uma forma mais eficaz. É fundamental que o perfil do aluno que evade seja identificado, assim como compreender as principais causas que o levou a evadir.

Com esta perspectiva, diversos trabalhos na literatura propõem soluções com o propósito de ajudar na identificação de alunos em risco de evasão, tais como as descritas na Seção 2.3. Muitas delas utilizam algoritmos de aprendizagem de máquina em grandes quantidades de dados (*big data*) com o objetivo de descobrir padrões que possam caracterizar o perfil de um aluno que evade.

No entanto, o desempenho de um modelo clássico de aprendizado de máquina depende significativamente da construção manual das *features* sobre os *datasets* disponíveis. Isto é, o pesquisador determina quais *features* são utilizadas como entrada aos algoritmos de aprendizagem de máquina. Segundo GOODFELLOW; BENGIO; COURVILLE (2016), esses sistemas de aprendizagem enfrentam dificuldades por dependerem

de "conhecimento codificado", tornando-se necessária a capacidade de aquisição de seus próprios conhecimentos. A escolha da representação dos dados tem um impacto enorme sobre o desempenho dos algoritmos de aprendizagem de máquina, pois para muitas finalidades, é difícil saber quais *features* devem compor o modelo.

Uma solução para tal problema seria usar aprendizado de máquina para determinar a representação dos dados para o próprio mapeamento e classificação dos dados (*i.e., determinar as features e fazer a classificação*. De acordo com GOODFELLOW; BENGIO; COURVILLE (2016), utilizar *Deep Learning* geralmente resulta em um desempenho melhor, comparado com as representações realizadas manualmente pelo pesquisador. Permite, também, que os sistemas de Inteligência Artificial se adaptem a novas tarefas com mais facilidade, com a mínima intervenção humana.

Com esse entendimento, esta dissertação promoveu o uso de *Deep Learning* e métodos computacionais tradicionais para gerar modelos preditivos que pudessem ajudar no processo de identificação de alunos em risco de evasão na Educação a Distância. Também, foi possível avaliar o desempenho de algoritmos convencionais visualizados na literatura e compara-los com *Deep Learning*. Desta forma, foram elaboradas duas abordagens: a Abordagem 1 em que foram utilizados oito algoritmos de aprendizado de máquina tradicionais para gerar os modelos preditivos, e a Abordagem 2 que utilizou-se uma *Recurrent Neural Network*.

A *Recurrent Neural Network* foi selecionada porque o processo de modelagem seria realizado com um *dataset* disponibilizado na plataforma Moodle[1]. A plataforma registra, de forma cronológica, milhares de dados que são gerados por meio das interações dos estudantes com o curso. Com isso, esse histórico temporal de dados poderia ser utilizado para o treinamento da *Recurrent Neural Network*, que possui células de memória com a capacidade de armazenar grandes extensões de dados temporais.

1.2 Problema de Pesquisa

Diante do cenário apresentado na Seção anterior, pode-se entender que a evasão de um aluno é um evento que pode ocorrer a qualquer momento do curso, e torna-se pouco provável que o professor ou qualquer outra pessoa responsável perceba esse ato em tempo de tentar ajudar o estudante em risco. Treinar um modelo preditivo que consiga identificar alunos em risco de evasão torna-se bastante relevante, ainda mais se esse modelo tiver acesso a uma base de dados que lhe forneça informações em tempo real sobre o desempenho dos alunos.

As propostas de modelos baseadas em técnicas convencionais podem não explorar de forma adequada (em termos de desempenho e qualidade) as características das mudanças

[1]O Moodle é uma plataforma amplamente utilizada pelos cursos da EaD do sistema UAB (Universidade Aberta do Brasil), que é responsável por fomentar a EaD nas universidades públicas do Brasi. Mais informações disponíveis em: <https://www.capes.gov.br/uab/o-que-e-uab>

de contexto temporal. Isto é, não modelam o contexto temporal na forma de uma grande memória com o histórico de atividades do aluno desde o primeiro dia. Assim, o problema de pesquisa é dado pela seguinte questão: **É possível modelar todos os pontos de contato digital do aluno ao longo do período acadêmico para predizer seu risco eminente de evasão?**

Esta questão pode ser respondida ao explorar os registros temporais que o Moodle armazena. Todas as ações realizadas *online* pelos alunos na plataforma, geram rastros digitais. Tais rastros são registrados no exato período de tempo em que foram executados. Estas características descrevem uma série temporal (*times series*), que considera os valores anteriores e o valor atual para a estimativa de novos eventos (SORJAMAA et al., 2007).

1.3 Justificativas da Pesquisa

Do ponto de vista científico, a pesquisa justifica-se pelo ato de utilizar aprendizado de máquina para amparar o cenário acadêmico, que enfrenta problemas com a evasão de alunos. Assim, a dissertação vai contribuir com a frente de pesquisa que trabalha com o aprendizado de máquina para predição de alunos em risco. Também, utilizar registros de interações de alunos para predição temporal de risco é uma abordagem não explorada no cenário educacional.

Do ponto de vista social, a pesquisa justifica-se pelas repercussões negativas que a evasão reflete no desenvolvimento acadêmico dos alunos que optaram por abandonar um curso, e que poderiam ser evitadas. A credibilidade que o aluno dá aos programas educacionais das instituições de ensino é prejudicada, assim como a sua reputação acadêmica, visto que a evasão é um indicador de qualidade da experiência educacional do aluno, e, portanto, é uma falha da instituição no atendimento às suas necessidades (LAGUARDIA; PORTELA, 2009). Também, vale salientar que o aluno que não completa o curso pode ter enfrentado situações que, talvez, a instituição de ensino pudesse ajuda-lo, ou até mesmo o professor. À vista disso, torna-se importante para a instituição de ensino, e para o corpo docente, o conhecimento de uma possível evasão.

Do ponto de vista econômico, tem-se as implicações que a evasão gera em termos de custos financeiros estão relacionados aos recursos que foram investidos para a mensalidade dos alunos, a organização e manutenção do curso (LAGUARDIA; PORTELA, 2009). Salienta-se, também, que são recursos do governo, visto que a educação a distância é administrada pelo sistema UAB (Universidade Aberta do Brasil) em instituições públicas. Um estudo realizado na Universidade Federal do Rio Grande do Sul (UFRGS) referente a um curso do ensino superior a distância do sistema UAB, revelou que o custo médio por aluno era de R$ 3.651,79, totalizando R$ 10.955,37 pelos seis semestres ofertados em 2009 (SOARES, 2015).

1.4 Objetivos da Pesquisa

Diante da contextualização apresentada nas Seções anteriores, em linhas gerais, a presente dissertação teve por objetivo principal utilizar os rastros digitais de alunos da Educação a Distância para gerar um modelo preditivo usando *Deep Learning*. Para complementar a pesquisa, métodos computacionais convencionais também foram utilizados no processo de modelagem, o que resultou em duas abordagens. Desta forma, pôde-se fazer uma comparação de desempenho entre as duas abordagens. Para isso, os seguintes objetivos específicos foram atingidos:

- Realizar um estudo para fundamentar os conceitos que envolvem o desenvolvimento desta pesquisa, como trabalhos relacionados, o aprendizado de máquina e seus algoritmos, *deep learning* e redes neurais, a modelagem preditiva no contexto da educação a distância e métricas para avaliar o desempenho dos modelos preditivos;

- Analisar, de forma exploratória, os dados disponíveis na plataforma Moodle. Assim, pode-se ter um entendimento da natureza dos dados, especificar quais deles seriam utilizados para construir um *dataset*, realizar um pré-processamento para a execução de testes, entre outras tarefas;

- Determinar quais dos algoritmos de aprendizado de máquina convencionais presentes na literatura seriam utilizados para a execução da primeira abordagem, assim como as métricas avaliativas que foram utilizadas para validação dos resultados;

- O primeiro experimento será realizado para entender melhor sobre o processo de modelagem e preparação do *dataset* para o treinamento de algoritmos de aprendizado de máquina.

- Um segundo experimento será executado para determinar pontos de corte no *dataset*. Nesse experimento, serão utilizadas novas métricas de avaliação de desempenho com o objetivo de verificar se havia sobreajuste (*overfitting*) nos modelos, e se a quantidade de dados utilizada era suficiente para a generalização dos modelos;

- O uso de *deep learning* será executado no terceiro experimento. Uma rede neural será estudada para lidar com a natureza dos dados adquiridos (série temporal), e, para sua validação, utilizar a métrica presente no experimento anterior, que proporcionava a visualização gráfica sobre o desempenho da rede;

- A comparação entre as abordagens será realizada para concluir a pesquisa.

1.5 Organização do Trabalho

Neste primeiro Capítulo, apresentou-se a contextualização necessária para introduzir o trabalho que foi desenvolvido. A descrição do problema de pesquisa foi colocada na Seção 1.2 para um melhor entendimento, e as justificativas para a realização da dissertação foram expostas na Seção 1.3. O objetivo principal e específicos podem ser visualizados na Seção 1.4.

O Capıtulo 2 apresentada a fundamentação teórica utilizada para o desenvolvimento da dissertação. Tópicos como a evasão na Educação a Distância, análise de dados preditivos, capacidade de generalização dos modelos, trabalhos relacionados e problemas em aberto na literatura são dissertados.

A metodologia da pesquisa está descrita no Capıtulo 3, em que contém as etapas e procedimentos metodológicos executados para a implementação de ambas abordagens.

No Capıtulo 4 são apresentados os resultados observados pelas abordagens que foram propostas. Foram realizados três experimentos e discussões relevantes.

Por fim, no Capítulo 5 são apresentadas as conclusões obtidas pelo desenvolvimento do trabalho, contribuições e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capıtulo apresenta a fundamentação teórica acerca do tema principal de pesquisa desta dissertação. Com isso, são apresentados evidências sobre a evasão na Educação a Distância na seção 2.1, conceitos e definições de aprendizagem de máquina para predição na seção 2.2 e uma revisão da literatura para comparar os trabalhos relacionados (Seção 2.3).

2.1 Evasão na Educação a Distância

A Internet proporciona opções inimagináveis para flexibilizar cursos presenciais e implementar cursos EaD (MORAN, 2003). A EaD, no final dos anos 1990, já era considerada um componente valioso para muitos sistemas educacionais, provando sua importância para atender demandas em áreas que as universidades e escolas tradicionais tinham dificuldade (GOUGH, 1996).

No Brasil, esta modalidade é fomentada pelo Sistema UAB[1], que tem por objetivo expandir a oferta de cursos e programas educacionais no superior do País. Assim como apoia pesquisas em metodologias inovadoras de ensino superior apoiadas por TIC.

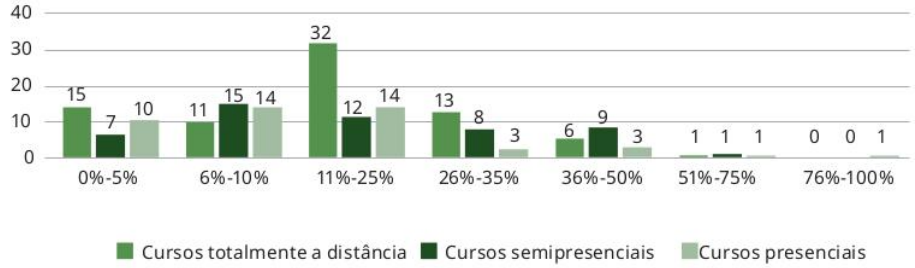
No entanto, a EaD tem enfrentado um grande problema de evasão de alunos (CENSO, 2018). A evasão de alunos é compreendida como sendo um fenômeno complexo, o qual pode ser influenciado por diversas variáveis (PRIM; FÁVERO, 2013).

Segundo EAD (2017), as questões que levam os alunos a evadirem estão relacionadas a situação financeira, falta de tempo e a pouca adaptação à modalidade a distância. A Figura 1 apresenta os índices de evasão nas modalidades totalmente a distância, semipresencial e presencial. Como pode ser observado, as taxas de evasão informadas pelas instituições de ensino participantes pertencem sobretudo na faixa entre 11% e 25%, seguido por 6%-10% e 0%-5%. Contudo, existe um número considerável de cursos nas faixas de 26%-35% e 36%-50%, sem contar que muitas universidades não participam da pesquisa. Salienta-se que o eixo y da Figura 1 representa o número de cursos. Con-

[1]Sistema da Universidade Aberta do Brasil. Para mais informações, acesso o link <http://www.capes.gov.br/acessoainformacao/informacoes-classificadas/93-conteudo-estatico/7836-o-que-e-uab>

cluindo, a maior parte dos cursos, considerando as três modalidades, relatam índices de evasão que chegam até 25%.

Figura 1: Índices de Evasão em Cursos Regulamentados



Fonte: (EAD, 2017)

2.2 Análise de Dados Preditivos

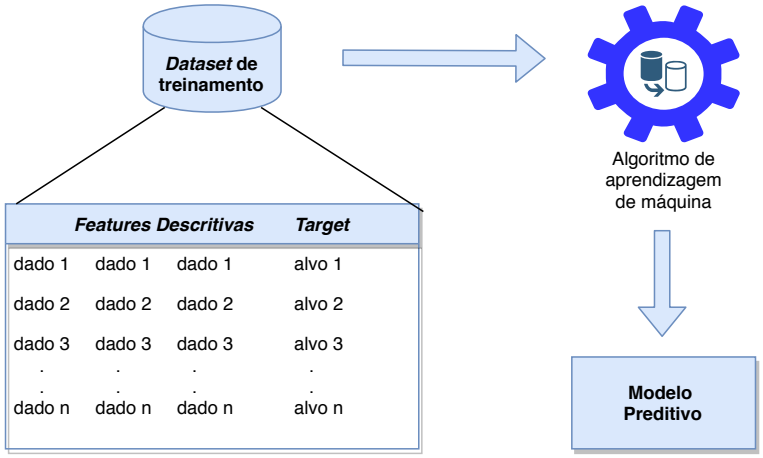
A análise de dados preditivos é a habilidade de utilizar e construir modelos capazes de fazer predições com base na extração de padrões em dados históricos (*e.g.*, predição de riscos, propensão, disgnóstico) (KELLEHER; MAC NAMEE; D'ARCY, 2015). Desta forma, um modelo pode ser muito útil para ajudar um usuário final a tomar decisões sobre um determinado problema. Para o treinamento do modelo preditivo utiliza-se aprendizado de máquina.

O aprendizado de máquina aborda a questão de como construir computadores que aperfeiçoam-se automaticamente através da experiência (JORDAN; MITCHELL, 2015). Para construir modelos utilizados em aplicações reais, (KELLEHER; MAC NAMEE; D'ARCY, 2015) destaca a utilização do aprendizado de máquina supervisionado. As técnicas de aprendizado de máquina supervisionado aprendem automaticamente um modelo de relacionamentos entre um conjunto de *features* de entrada (características) e um *target* (variável alvo) com base em um conjunto de instâncias (dados exemplos históricos) (KELLEHER; MAC NAMEE; D'ARCY, 2015). A Figura 2 ilustra a aprendizagem de um modelo sobre um conjunto de instâncias.

O desempenho dos algoritmos de aprendizado de máquina depende muito da representação dos dados que eles recebem (GOODFELLOW; BENGIO; COURVILLE, 2016). Por exemplo, um sistema de Inteligência Artificial para recomendação de parto cesário usando o algoritmo *Logistic Regression*, o sistema não examina o paciente diretamente, e sim o médico atribui várias informações descritivas ao sistema (*features*), como a presença ou não de uma cicatriz uterina. O *Logistic Regression* aprende como cada uma dessas *features* correlaciona-se com vários *targets*. No entanto, não pode influenciar como as *features* são definidas.

Segundo GOODFELLOW; BENGIO; COURVILLE (2016), uma solução para esse problema é usar o aprendizado de máquina para descobrir a representação dos dados.

Figura 2: Treinamento de um Modelo Preditivo

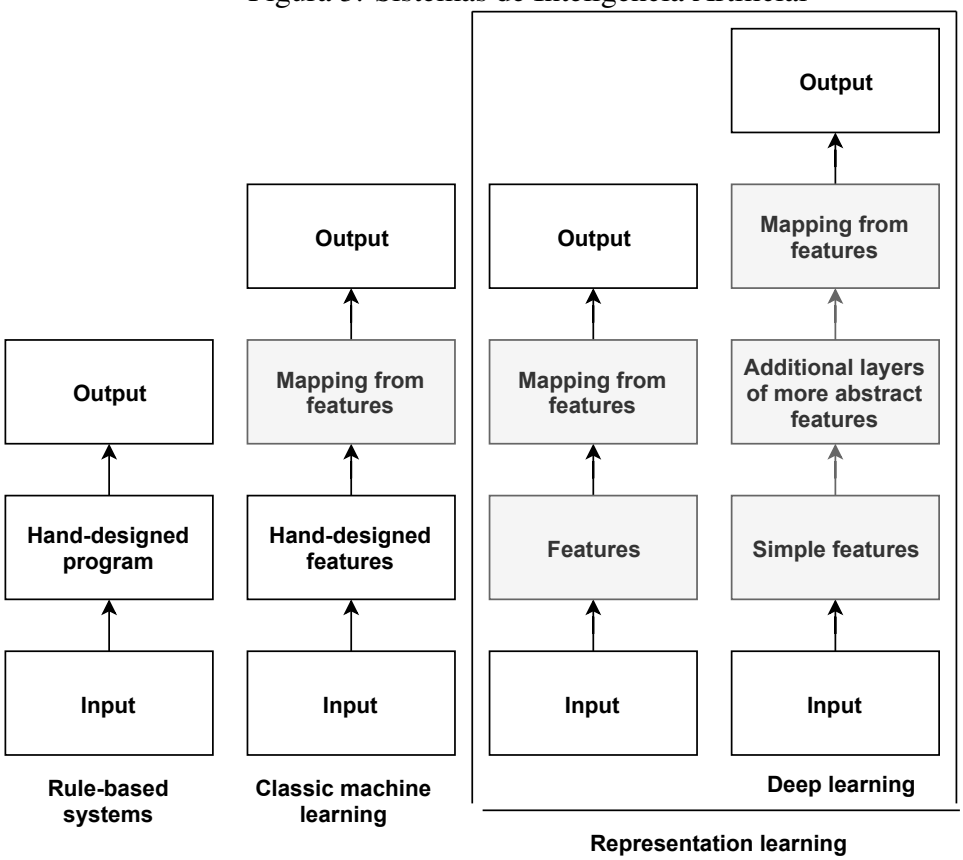


Fonte: Adaptado de (KELLEHER; MAC NAMEE; D'ARCY, 2015)

Essa abordagem é conhecida como Aprendizagem Representacional (*Representation Learning*).

Como pode ser observado na Figura 3, são apresentados quatro fluxogramas descrevendo como as diferentes partes de um sistema IA se relacionam umas com as outras em diferentes disciplinas.

Figura 3: Sistemas de Inteligência Artificial



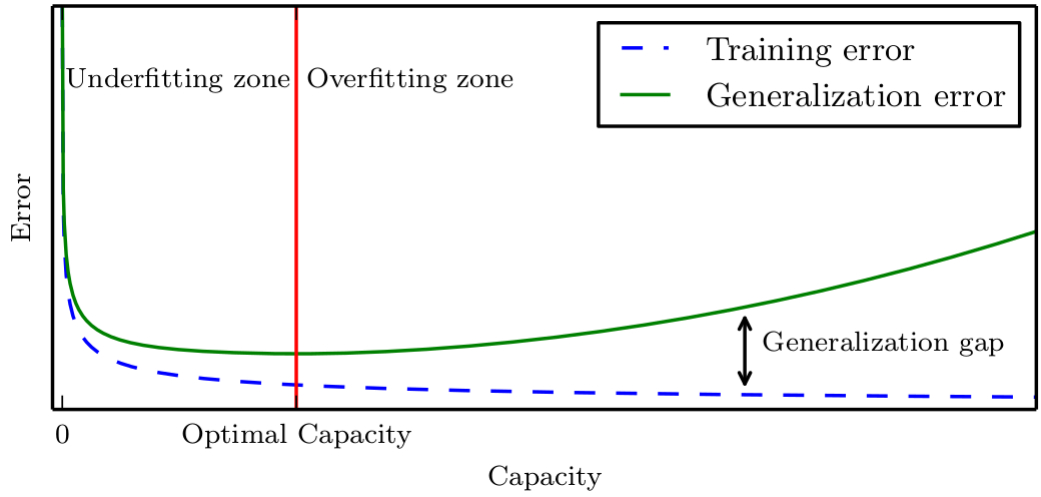
Fonte: (GOODFELLOW; BENGIO; COURVILLE, 2016)

O primeiro fluxograma refere-se aos *Rule-Based Systems*, que são sistemas que utilizam regras condicionais (*if-then*) para automatizar uma base de conhecimento inteligente. O segundo representa o aprendizado de máquina (*machine learning*), que possibilitou que o computador adquirisse seu próprio conhecimento por meio de dados (as *features*). O terceiro bloco da início a aprendizagem representacional, em que o computador consegue aprender sobre dados brutos. *Deep Learning* representa a evolução do terceiro bloco em termos de poder de processamento, redes neurais maiores e *big data*. Esta dissertação, na sua primeira fase, encontra-se no *Classic Machine Learning* (segundo bloco), em que os algoritmos de aprendizado de máquina são treinados com *features* geradas pelo pesquisador. Na sua segunda fase, então, esta dissertação apresentará a aplicação de *deep learning* no contexto deste trabalho, com o objetivo de melhorar o desempenho e a capacidade de generalização do modelo.

2.2.1 Capacidade de Generalização dos Modelos

O principal desafio em aprendizado de máquina é capacitar o algoritmo de aprendizado de máquina para que ele consiga ter um bom desempenho quando for executado sobre dados novos e não vistos anteriormente, o que é chamado de generalização (GOODFELLOW; BENGIO; COURVILLE, 2016). O desempenho de um algoritmo é determinado pela sua capacidade de reduzir o erro de treinamento, e diminuir a diferença entre o erro de treinamento e o de teste. Esses dois fatores correspondem a *underfitting* e *overfitting*. *Underfitting* ocorre quando o modelo não é capaz de obter um valor de erro baixo no conjunto de treinamento, enquanto que o *overfitting* ocorre quando a diferença entre o erro de treinamento e o erro de teste é muito grande. Esta concepção pode ser visualizada na Figura 4.

Figura 4: Capacidade de Generalização



Fonte: (GOODFELLOW; BENGIO; COURVILLE, 2016)

Na Figura 4, pode-se visualizar as áreas de *underfitting* e *overfitting*, a curva que

representa o erro de treinamento, e a curva que reflete o erro de generalização. No eixo x, tem-se a capacidade que, segundo (GOODFELLOW; BENGIO; COURVILLE, 2016), representa a habilidade de um algoritmo se ajustar a uma variedade de funções. Na Figura, observa-se um ponto em que a capacidade está em seu melhor estado, em que a diferença (*gap*) entre as curvas não está tão grande, e ambas as curvas estão com um valor de erro baixo. Na medida em que o erro de treinamento vai reduzindo, o erro de generalização aumenta.

Nesta dissertação, foram plotadas as curvas de aprendizagem dos algoritmos para avaliar seu desempenho de treinamento. O objetivo é verificar o quanto os algoritmos podem ser beneficiados com a adição de mais exemplos de treinamento, e se o estimador sofre mais com erro de variância ou *bias*. As curvas de aprendizagem mostram a pontuação de validação e treinamento de um algoritmo, de acordo com o número de exemplos de treinamento (SCIKIT-LEARN, 2007). Como foi utilizado *stratified k-fold cross-validation*, em que k é 10, essa pontuação de validação e treinamento está distribuída em 10 pontos que formam as curvas.

A variância refere-se a sensibilidade de um estimador quanto a amostragem de treinamento (FRIEDMAN, 1997). Quando essa sensibilidade é baixa, o estimador estará mais estável contra mudanças (variações de amostragem) nos dados, e, logo, sua variação será menor sobre repetições de amostragem. O *bias* de um estimador é seu erro médio para diferentes amostragens de treinamento. Ele nos diz o quão perto o estimador está de chegar no resultado esperado. Para ambos *bias* e variância, deseja-se ter um valor baixo, pois ambos contribuem para o erro do estimador (FRIEDMAN, 1997).

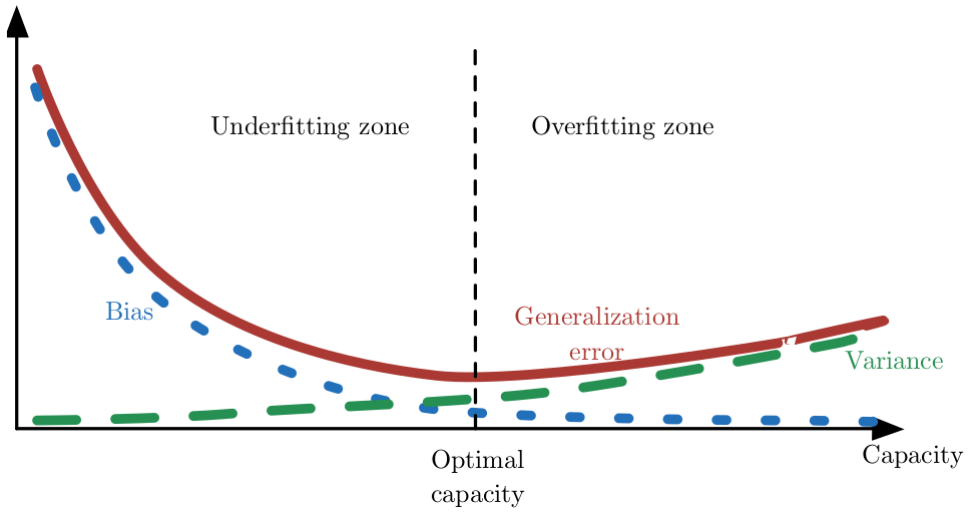
O objetivo do treinamento é obter informações sobre o *target* (Evadiu). O *bias* e a variância devem ser encontrados em seu valor médio, pois quando se tem um valor ótimo de um, isto é, um valor baixo, o outro, consequentemente, terá um valor alto. Portanto, a sensibilidade quanto aos dados de treinamento é essencial, e quanto maior for essa sensibilidade, menor será o *bias*. No entanto, isso resulta no aumento da variância e, consequentemente, há um "desvio variância-*bias*"(do inglês *bias-variance trade-off*). Essa idealização está ilustrada na Figura 5.

Para um determinado *bias*, a variância geralmente diminui com o aumento do tamanho da amostra de treinamento. Todavia, para problemas com grandes amostras de treinamento, o *bias* pode contribuir para o erro do estimador.

2.3 Trabalhos Relacionados

Com o objetivo de aprofundar-se no contexto desta pesquisa, foi realizada uma revisão da literatura para identificar, compreender e analisar as pesquisas publicadas no âmbito desta dissertação. Também, esta revisão tem o intuito de obter um panorama referente aos estudos que foram e estão sendo realizados no contexto da EaD.

Figura 5: *Bias* e Variância



Fonte: (GOODFELLOW; BENGIO; COURVILLE, 2016)

A busca de trabalhos para revisão foi realizada em repositórios que apresentam um certo padrão de qualidade. Assim, selecionaram-se cinco repositórios, um nacional e quatro internacionais, que estão presentes na Tabela 1.

Tabela 1: Repositórios de Pesquisa

	Repositórios
Nacional	Simpósio Brasileiro de informática na Educação (SBIE)[2]
Internacional	ACM Digital Library[3]
	IEEE Xplore Digital Library[4]
	Scopus[5]
	Taylor and Francis[6]

O SBIE ocorre dentro do Congresso Brasileiro de Informática na Educação, e é um evento importante em informática na educação no território nacional. Os repositórios internacionais selecionados são os mais abrangentes no tema da revisão, e integram diversas conferências e periódicos. Desta forma, acredita-se que com esses repositórios seja possível obter uma boa compreensão sobre o estado da arte.

Na Tabela 2 são apresentados os critérios de inclusão que foram utilizados para selecionar os artigos para esta revisão. Em MORENO-MARCOS et al. (2018), salienta-se que primeiro a pesquisa (busca) é realizada de acordo com os critérios de inclusão. Posteriormente, os resultados são filtrados por meio dos critérios de exclusão.

Tabela 2: Critérios de Inclusão

Palavras-Chave	**Keywords**
Predição, Preditivo,	Prediction, Predictive, Predict,
Predizer, Predizendo	Predicting, Forecast, Forcasting
Evasão	Dropout

Como pode ser visualizado na Tabela 2, utilizou-se como *string* de busca a palavra "predição"e seus sinônimos, juntamente com a palavra "evasão". O termo "*Dropout*"é utilizado na literatura estrangeira para referir-se a evasão (pelo menos na maioria dos casos). As palavras "*forecast*"e "*forecasting*"também foram encontradas em alguns trabalhos e, por essa razão, foram incluídas. As palavras "modelo"ou "modelagem"não foram utilizadas, porque quando inseridas, os resultados de busca eram muito reduzidos. Então, utilizar apenas os sinônimos de predição e evasão pareceu suficiente.

As palavras-chave foram utilizadas para buscas em títulos, resumos (*abstracts*) e palavras-chave, uma vez que essas três partes devem conter os termos mais representativos do trabalho (MORENO-MARCOS et al., 2018). À vista disso, as palavras-chave formaram as seguintes *strings* de busca:

- *String* em inglês: ((prediction OR predictive OR predict OR predicting OR forecast OR forecasting) AND (dropout OR evasion).
- *String* em português: ((predição OR preditivo OR predizer OR predizendo) AND evasão).

As buscas restringiram-se até o final do ano de 2018 e não tiveram uma restrição na data inicial. A EaD é uma modalidade de ensino que está presente na instituições de ensino há muito tempo, então podem haver pesquisas sendo realizadas até mesmo antes dos anos 2000.

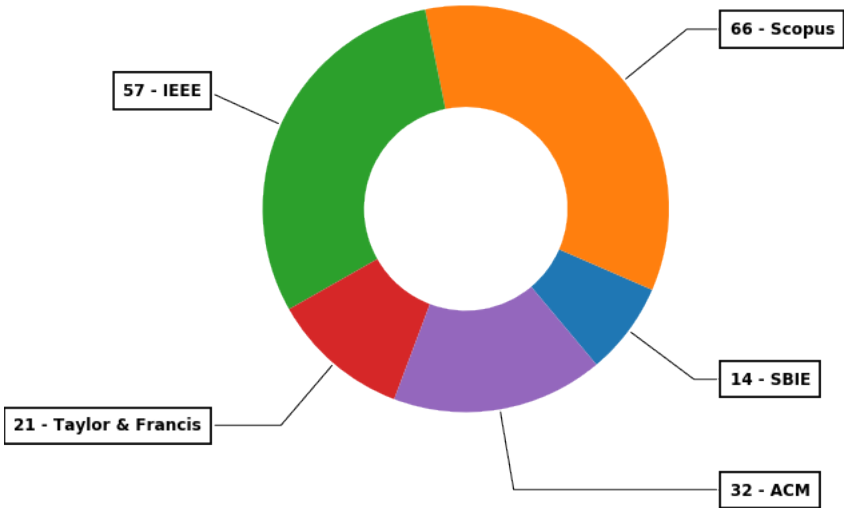
As *strings* de busca resultaram em um total de 193 artigos. Deste total, 14 artigos pertencem ao SBIE, 69 ao Scopus, 57 do IEEE, 32 do ACM e 21 do Taylor Francis. Salienta-se que trabalhos em que o título revelava explicitamente outro ponto de vista (trabalho fora do escopo), estes não eram nem coletados para análise (*e.g.*, Predição em MOOCs). Na Figura 6, pode-se visualizar a distribuição de artigos por periódico.

Após a definição dos critérios de inclusão, definiram-se os seguintes critérios de exclusão:

1. Artigos duplicados;
2. Estudos fora do escopo;
3. Estudos que não apresentam um modelo preditivo;
4. Estudos secundários.

Devido a grande variedade de revistas e conferências em comum nos periódicos utilizados, é possível que alguns trabalhos estejam duplicados e, devido a isso, definiu-se o critério 1. Estudos que não estejam relacionados a predição de evasão em cursos da EaD também não serão considerados (2), assim como aqueles que não apresentarem um

Figura 6: Artigos por periódicos



modelo de predição (3). E por fim, o critério 4 exclui estudos que compreendem revisões da literatura ou *Surveys*, visto que a busca é por estudos primários.

A Tabela 3 apresenta os trabalhos selecionados com base nos critérios de exclusão. Haviam trabalhos duplicados entre os periódicos Scopus, ACM e IEEE, que foram eliminados. Em alguns casos, foi possível identificar a inclusão/exclusão do trabalho a partir da descrição do título, como "Predição em MOOCs", por exemplo. No entanto, analisava-se o *abstract* para confirmação da inclusão/exclusão e, em alguns casos, outras partes do texto necessitavam ser verificadas, como a conclusão e o desenvolvimento, por exemplo.

As conferências e revistas internacionais nas quais os artigos foram publicados podem ser conferidas na Tabela 4.

2.3.1 Descrição dos Trabalhos Relacionados

O estudo de KOTSIANTIS; PIERRAKEAS; PINTELAS (2003) foi um dos primeiros publicados no tema desta dissertação, em que foi validado o uso de técnicas de aprendizado de máquina para lidar com o problema da evasão de alunos na EaD. Nesse trabalho, realizou-se uma série de experimentos para avaliar qual algoritmo seria mais apropriado para a predição de evasão. Os autores chegaram a conclusão de que o algoritmo *Naive Bayes* seria o mais apropriado para ser aplicado ao problema, e com isso, foi desenvolvido um protótipo de uma ferramenta que seria utilizada para identificar alunos em risco de evasão de forma automática, utilizando este algoritmo.

Em KOTSIANTIS; PINTELAS (2004), foi apresentado uma arquitetura de alto nível e um estudo de caso para o protótipo de uma ferramenta capaz de identificar alunos propensos a evadir no ensino EaD. Os autores KOTSIANTIS; PINTELAS (2004), a ferramenta pode ser utilizada para predição de notas acadêmicas, previsão de quais alunos enviarão tarefas por escrito, entre outras utilizadas. A ferramenta, também, possui com-

Tabela 3: Trabalhos Selecionados

Autoria	**Título do Trabalho**
Kotsiantis et al. (2003)	Preventing Student Dropout in Distance Learning Using Machine Learning Techniques
Kotsiantis e Pintelas (2004)	A Decision Support Prototype Tool for Predicting Student Performance in an ODL Environment
Xenos (2004)	Prediction and Assessment of Student Behaviour in Open and Distance Education in Computers Using Bayesian Networks
Kotsiantis (2009)	Educational Data Mining: A Case Study for Predicting Dropout-Prone Students
Dewan et al. (2015)	Predicting Dropout-Prone Students in E-Learning Education System
Kostopoulos et al. (2015)	Estimating Student Dropout in Distance Higher Education Using Semi-Supervised Techniques
Kostopoulos et al. (2015)	Predicting Student Performance in Distance Higher Education Using Semi-supervised Techniques
Cambruzzi et al. (2015)	Dropout Prediction and Reduction in Distance Education Courses with the Learning Analytics Multitrail Approach
Santana et al. (2015)	A Predictive Model for Identifying Students with Dropout Profiles in Online Courses
Silva et al. (2015)	Um Modelo Preditivo para Diagnóstico de Evasão Baseado nas Interações de Alunos em Fóruns de Discussão
Kostopoulos et al. (2017)	Early Dropout Prediction in Distance Higher Education Using Active Learning
Kostopoulos et al. (2017)	Predicting Student Performance in Distance Higher Education Using Active Learning
Peña et al. (2017)	Mining Activity Grades to Model Students' Performance
Burgos et al. (2017)	Data Mining for Modeling Students' Performance: A Tutoring Action Plan to Prevent Academic Dropout
Queiroga et al. (2017)	Predição de Estudantes com Risco de Evasão em Cursos Técnicos a Distância
Rabelo et al. (2017)	Utilização de Técnicas de Mineração de Dados Educacionais para a Predição de Desempenho de Alunos de EaD em Ambientes Virtuais de Aprendizagem
Ramos et al. (2017)	Um Modelo Preditivo da Evasão dos Alunos na EAD a partir dos Construtos da Teoria da Distância Transacional
Galafassi et al. (2017)	Predictive Teaching and Learning
Kang e Wang (2018)	Analyze and Predict Student Dropout from Online Programs
Ramos et al. (2018)	Um Estudo Comparativo de Classificadores na Previsão da Evasão de Alunos em EAD
Kostopoulos et al. (2018)	Forecasting students' success in an open university

ponentes como seleção de *features* e de balanceamento de dados, e será bastante útil para auxiliar e facilitar o papel dos tutores no processo de acompanhamento dos estudantes.

Na pesquisa de XENOS (2004), uma abordagem metodológica baseada em Redes Bayesianas é apresentada para modelar o comportamento dos alunos para predições. O

Tabela 4: Revistas e Conferências Internacionais

Autoria	**Conferência/Revista**
Kotsiantis et al. (2003)	International Conference on Knowledge-Based and Intelligent Information and Engineering Systems
Kotsiantis e Pintelas (2004)	Interactive Technology and Smart Education
Xenos (2004)	Computers & Education
Kotsiantis (2009)	International Journal of Knowledge Engineering and Soft Data Paradigms
Dewan et al. (2015)	UIC-ATC-ScalCom-CBDCom-IoP - IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and IEEE 12th Intl Conf on Autonomic and Trusted Computing and IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops
Kostopoulos et al. (2015)	Panhellenic Conference on Informatics
Kostopoulos et al. (2015)	Model and Data Engineering
Cambruzzi et al. (2015)	Journal of Universal Computer Science
Santana et al. (2015)	International Conference on Educational Data Mining
Kostopoulos et al. (2017)	8th International Conference on Information, Intelligence, Systems & Applications (IISA)
Kostopoulos et al. (2017)	International Conference on Engineering Applications of Neural Networks
Peña et al. (2017)	International Conference on Engineering & MIS – ICEMIS2017
Burgos et al. (2017)	Computers and Electrical Engineering
Galafassi et al. (2017)	EPIA Conference on Artificial Intelligence
Kang e Wang (2018)	International Conference on Compute and Data Analysis
Kostopoulos et al. (2018)	International Journal of Learning Technology

estudo foi realizado com dados de alunos de um curso de informática da Universidade Aberta Hellenic na Grécia, que implanta métodos EaD. O método apresentado oferece uma maneira eficaz de modelar a experiência passada para tomar decisões em relação ao procedimento educacional, e também, identificar razões que levaram os alunos a um determinado estado (*e.g*, abandonar o curso).

No trabalho de KOTSIANTIS (2009), é apresentado um estudo experimental que tem por objetivo mostrar o desempenho do algoritmo *Naive Bayes*, utilizando uma técnica chamada de *local cost-sensitive* para lidar com o problema de desbalanceamento de dados no *dataset*. Segundo KOTSIANTIS (2009), o problema está no desbalanceamento entre as classes do *dataset*, pois a classe de interesse que representa os alunos que evadiram possui menos observações que a classe majoritária, e isso resulta em um erro maior na classe de interesse.

Na pesquisa de DEWAN et al. (2015), foi proposta uma técnica para prever alunos propensos a desistência que usa as informações dos alunos extraídas do AVA de cursos *online*. Essa técnica utiliza uma combinação de múltiplos classificadores com o objetivo de reduzir as imprecisões dos classificadores individuais, sendo eles o *k-Nearest Neighbor*

Classifier, *Radial Basis Function Network* e *Support Vector Machines*.

No artigo de KOSTOPOULOS; KOTSIANTIS; PINTELAS (2015a), estudou-se o uso de técnicas semi-supervisionadas para predição de evasão em um curso a distância de Ciência da Computação da Universidade Aberta Hellenic na Grécia. O aprendizado semi-supervisionado consiste em gerar predições confiáveis usando poucos dados rotulados e muitos dados não rotulados (*i.e.*, quando não existe a classificação prévia de estudante evadido ou não). Os resultados revelam que uma boa acurácia preditiva pode ser obtida usando poucos dados rotulados em comparação com algoritmos de aprendizado supervisionado. Em KOSTOPOULOS; KOTSIANTIS; PINTELAS (2015b) é apresentada uma versão mais completa do estudo proposto em (KOSTOPOULOS; KOTSIANTIS; PINTELAS, 2015a), no qual utilizam-se mais atributos, e com isso, mais etapas são acrescentadas no processo de modelagem. Em cada etapa apresenta-se a acurácia dos algoritmos sobre a utilização de técnicas semi-supervisionadas. Para ambos os estudos, um algoritmo multi-classificador *Tri-Training* com um classificador base C4.5 (árvore de decisão) obteve um desempenho melhor que as demais técnicas semi-supervisionadas, com uma acurácia entre 53.26% e 75.29% antes da metade do ano acadêmico. Destaca-se que em ambos os estudos apresenta-se uma ferramenta *web* desenvolvida sobre o modelo preditivo gerado para predição de risco de evasão.

Em CAMBRUZZI; RIGO; BARBOSA (2015) foi apresentado um sistema de *Learning Analytics* desenvolvido para lidar com o problema de evasão em cursos da EaD. Este sistema possui ferramentas complementares que permite a visualização dos dados, predições, análise textual, apoio a ações pedagógicas, entre outros. Para a implementação do sistema foi adotada uma abordagem chamada *Multitrail*, que possibilita a representação e manipulação de dados de diversas fontes e formatos. Os resultados revelam que a predição de evasão possui uma acurácia de 87%. Com isso, foi implementado um conjunto de ações pedagógicas referentes aos alunos entre as maiores probabilidades de evasão, e observou-se uma redução média de 11% nas taxas de evasão.

No trabalho de SANTANA et al. (2015), foram utilizados quatro algoritmos de classificação a fim de encontrar o modelo com maior acurácia na predição do perfil dos alunos desistentes em cursos a distância. Os dados para geração de modelos foram obtidos de duas fontes de dados disponíveis na Universidade Federal do Alagoas no Brasil. Os resultados mostraram que o modelo gerado pelo algoritmo *Support Vector Machines* foi o mais acurado dentre os selecionados, com 92,03% de acurácia.

No estudo de SILVA et al. (2015) foi comparado o desempenho de cinco algoritmos de classificação para a geração de um modelo preditivo para evasão de alunos da Ead. O modelo é proposto para realizar o diagnóstico em AVA a partir de interações de alunos em fóruns de discussão. Os resultados obtidos apontaram que as técnicas baseadas em árvore de decisão tiveram os melhores desempenhos e, em alguns casos, alcançaram taxa de precisão acima de 73%.

Na pesquisa apresentada por KOSTOPOULOS et al. (2017), foi investigada a eficiência de técnicas de aprendizagem ativa para a predição de taxas de evasão de alunos do curso a distância de Ciência da Computação da Universidade Aberta Hellenic. A aprendizagem ativa é um método típico entre técnicas que utilizam dados não rotulados com uma pequena quantidade de dados rotulados (KOSTOPOULOS et al., 2017). Para isso, foi utilizado *Pool-Based Sampling* com *Margin Sampling Query strategy*. Os modelos foram validados pelas métricas de acurácia e AUC. Os resultados apontam que uma boa qualidade preditiva pode ser alcançada utilizando esta abordagem. O aprendiz ativo que usa o NB como classificador obteve o melhor desempenho, com uma acurácia de 66,26% com base apenas em dados pré-universitários apenas, ou seja, dados referentes ao aluno antes deste ingressar no curso. No meio do ano letivo, o algoritmo já atingia 84,56% de acurácia.

Em (KOSTOPOULOS et al., 2017), também utilizaram-se técnicas de aprendizagem ativa na tarefa de classificação, com o objetivo de prever o desempenho de alunos nos exames finais. Neste caso, os autores obtiveram um melhor resultado utilizando o algoritmo *Sequential Minimal Optimization*, que é uma versão otimizada do SVM. O *Sequential Minimal Optimization* apresentou uma acurácia de 64,61% apenas utilizando dados pré-universitários no início do ano letivo. À medida em que novas variáveis educacionais são adicionadas, a acurácia é continuamente aumentada, atingindo 75,54% no final do primeiro semestre. Além disso, antes dos exames finais a acurácia excede os 80%.

Na pesquisa de PEÑA et al. (2017) foram utilizados dados da plataforma Moodle que contêm o histórico de atividades dos usuários para gerar um modelo preditivo que identificasse o risco de um aluno evadir evadir em tempo real. Dentre os algoritmos utilizados na pesquisa para validação, o algoritmo *Logistic Regression*, que obteve o melhor desempenho, foi utilizado para a implementação de um sistema que prevê em tempo real se um aluno vai evadir ou não. Em BURGOS et al. (2018) apresenta-se uma versão estendida desta pesquisa, em que foi, também, a aplicação dos modelos gerados em um plano de intervenção, assim como uma discussão sobre a aplicabilidade do modelo proposto.

No trabalho de QUEIROGA; CECHINEL; ARAÚJO (2017) apresenta-se uma abordagem para a identificação de alunos em risco de evasão em cursos técnicos a distância. Aqui, foi utilizada apenas a contagem de interações dos estudantes dentro do AVA Moodle, assim como atributos derivados dessas contagens. Os resultados parecem satisfatórios desde as primeiras semanas dos cursos com uma acurácia próxima a 75%. Os autores destacam que o desempenho dos algoritmos melhora ao aumentar a granularidade, isto é, utilizar a contagem de interações diárias.

O estudo de RABELO et al. (2017) relata a aplicação de técnicas de mineração de dados educacionais sobre dados do Moodle para predição de desempenho de alunos da EaD. Predizer o desempenho final dos alunos contribui para propor reajustes na conduta durante o processo de ensino e aprendizagem, assim como para diminuir os índices de

evasão (RABELO et al., 2017). O melhor resultado obtido foi através do algoritmo J48, com 96,5% de acertos (496 instâncias) e 3,5% de erros de classificação (18 instâncias).

Em RAMOS et al. (2017) focou-se na utilização dos construtos da Teoria da Distância Transacional como preditores para a geração de um modelo que prevê a evasão de alunos na EaD. Esta teoria sugere que os cursos da EaD podem ser avaliados e planejados conforme uma medida proveniente de três construtos, sendo eles: autonomia, diálogo e estrutura (RAMOS et al., 2017). Como resultados, pôde-se observar que os construtos da Teoria da Distância Transacional podem estar vinculados com a evasão de alunos. A partir desta teoria, os autores obtiveram uma acurácia 89,42% com o algoritmo *Logistic Regression*.

Na pesquisa de GALAFASSI; GALAFASSI; VICARI (2017), foi apresentada uma análise comportamental acadêmica utilizando registros de atividades do Moodle com a finalidade de poder utiliza-los para predição de desempenho. Entre as características principais que caracterizam o comportamento do aluno, estão o tempo *online* no curso, tarefas submetidas e visualização de recursos. Assim, essas características foram relacionadas com o desempenho do aluno (*i.e.*, sucesso, reprovação, evasão), para então enriquecer um modelo preditivo. Nesse seguimento, seriam gerados dois tipos de modelos, um apenas com os dados de desempenho, e outro com os dados enriquecidos, de acordo com as características analisadas. Os resultados mostram que o modelo de dados enriquecido é mais acurado e pode ajudar o professor a identificar alunos em risco.

Uma estrutura de mineração de dados educacionais foi desenvolvida em (KANG; WANG, 2018) para analisar os dados institucionais e prever possíveis alunos em risco de evasão em programas *online* antes do início do novo período. Os modelos de predição de regressão linear, quando utilizados para dados não balanceados e dados balanceados, produzem taxas de acurácia relativamente altas, assim como de *recall* (KANG; WANG, 2018). Os autores apresentaram duas listas de alunos com possíveis desistências para os administradores, para que pudessem intervir e tentar trabalhar com o caso para que não desistam. O resultado do projeto foi adotado pelo centro de educação a distância da instituição em que o estudo foi realizado nos Estados Unidos.

O estudo de RAMOS et al. (2018) apresenta uma análise comparativa de cinco algoritmos classificadores para a tarefa de predição de alunos com risco de evasão em cursos de graduação EaD. Com base na análise dos resultados, o algoritmo *Logistic Regression* foi selecionado para prosseguir nas próximas etapas da pesquisa, sendo utilizado para uma futura implementação em um ambiente experimental. A validação foi feita por meio das métricas acurácia, *precision*, *recall* e AUC.

Por fim, em (KOSTOPOULOS et al., 2018) apresenta-se um conjunto de algoritmos de classificação e regressão para predição de desempenho de alunos em um EaD. A metodologia proposta combina regras de classificação e regressão, e foi comparada com outros classificadores e algoritmos de regressão, revelando excelentes resultados. Também, um

protótipo de ferramenta de suporte seria projetado com base na metodologia apresentada.

Foram verificadas algumas pesquisas que encontram-se no contexto de *representation learning*, mas em contextos diferentes. No estudo de OKUBO et al. (2017), foi proposto um método que utiliza Redes Neurais Recorrentes para predição de notas finais de alunos utilizando dados de sistemas educacionais. O método, que foi aplicado sobre dados de 108 alunos, foi comparado com uma técnica de análise de regressão múltipla, e revelou ser mais efetivo. Em CORRIGAN; SMEATON (2017) foi proposto um método para a predição do sucesso do aluno, que consiste na implementação de uma Rede Neural Recorrente em uma arquitetura de *Long Short Term Memory* (LSTM). Os dados utilizado neste estudo foram extraídos do Moodle (*i.e.*, os logs).

2.3.2 Problemas em Aberto Identificados na Literatura

Em resumo, vários estudos investigam a aplicação de técnicas de aprendizado de máquina para prever e identificar estudantes que estão em risco de evasão nos cursos da EaD. Esses trabalhos compartilham algumas similaridades, como identificar e comparar o desempenho de algoritmos classificadores para avaliar qual seria o melhor utilizado naquele contexto.

Após uma análise do estado da arte, encontrou-se a seguinte lista de limitações das técnicas propostas para a problemática de evasão na EaD:

- Limitação de amplitude e escopo dos *datasets*. Em alguns trabalhos, não são utilizados os rastros digitais ao longo da vida acadêmica do aluno para a modelagem preditiva. Há modelos baseados apenas em perfis socio-demográficos. Isto é, considerando os dados do aluno antes do período do curso, como faixa etária, renda familiar, entre outros.

- Nos trabalhos selecionados para a análise final, encontraram-se modelos preditivos baseados em técnicas convencionais de aprendizado de máquina apenas. Assim, o desempenho e qualidade do modelo podem estar limitadas às escolha de *features* personalizadas. Um exemplo seria o algoritmo *Logistic Regression*.

- As métricas utilizadas para avaliar os modelos, muitas vezes, trazem um cenário de bom desempenho, porém, em alguns casos, essa não é uma informação confiável. Como averiguar se aquele alto desempenho não é um sobreajuste (*overfitting*)?

- A natureza da geração dos dados na plataforma Moodle é de séries temporais continuas. As propostas de modelos baseadas em técnicas convencionais não exploram de forma adequada (desempenho e qualidade) as características das mudanças de contexto temporal que são importantes para modelar o contexto do aluno na forma de uma grande memória com o histórico de atividades do estudante desde o pri-

meiro dia. É possível modelar todos os pontos de contato digital do aluno ao longo do peŕíodo acadêmico para predizer seu risco eminente de evasão?

Contribuições desta dissertação perante o cenário observado na literatura:

- Avaliar o uso dos rastros digitais do AVA Moodle para modelagem preditiva de risco de evasão, avaliando os métodos tradicionais de aprendizado de máquina, com o desempenho de uma rede neural recorrente.
- Utilizar métricas de validação que consigam identificar a eficácia do modelo resultante como ferramenta de identificação de alunos em risco de evasão na EaD.

3 PROCEDIMENTOS METODOLÓGICOS

Esta pesquisa, conforme sua natureza, classifica-se como aplicada. A pesquisa aplicada tem por objetivo gerar conhecimentos para aplicações práticas e conduzidos à resolução de problemas específicos (SILVA; MENEZES, 2001) (PRODANOV; FREITAS, 2013). Quanto à abordagem do problema, a pesquisa classifica-se como quantitativa. Tal classificação é determinada devido aos métodos estatísticos que foram aplicados para analisar e avaliar os resultados do estudo. A pesquisa quantitativa considera que tudo pode ser quantificável, de modo a interpretar informações e opiniões no formato numérico com técnicas estatísticas (SILVA; MENEZES, 2001). Do ponto de vista de seus objetivos, a pesquisa é classificada como exploratória. Este tipo de pesquisa busca proporcionar maior entendimento sobre o problema com a intenção de torná-lo claro (SILVA; MENEZES, 2001). Assim, foi realizado um levantamento bibliográfico para obter maior compreensão sobre a aplicação de técnicas de aprendizado de máquina para predição na educação a distância.

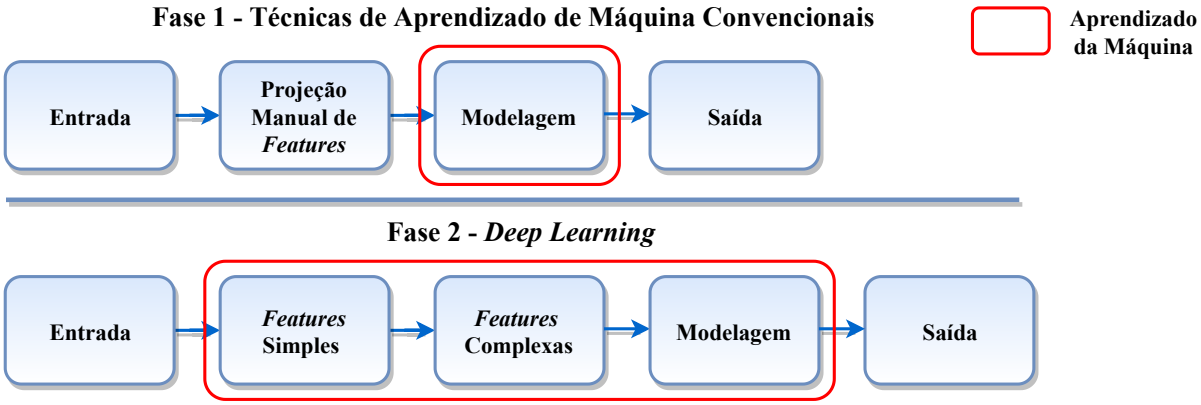
Estudos preliminares foram conduzidos para introdução de conhecimentos referente ao tema de pesquisa. Realizou-se uma pesquisa bibliográfica, e também, houve uma familiarização com a ferramenta utilizada para o desenvolvimento do trabalho, o Jupyter Notebook[1]. O Jupyter é uma plataforma *web* de código aberto que permite ao usuário construir e compartilhar documentos (*notebooks*) com representações gráficas, textos, codificações, fórmulas, entre outros recursos. Tais características constituem uma plataforma eficiente para o desenvolvimento de projetos. As bibliotecas de *data science* do Python (pandas, numpy, matplotlib, scikit-learn e outras) foram usadas para implementar os *pipelines* de processamento de dados (ou simplesmente *pipeline* de dados) no Jupyter.

Como foi proposto nos objetivos dessa dissertação, foi realizado o planejamento para o desenvolvimento de duas abordagens. Na primeira abordagem, oito algoritmos de aprendizado de máquina foram utilizados para serem validados e analisados, e, para a segunda abordagem, foi utilizado *deep learning* para a implementação de três redes neurais que seriam validadas e comparadas com os resultados obtidos pela primeira abordagem. A Figura 7 apresenta estas duas abordagens, que teve como embasamento a Figura 3 da

[1]Link de acesso: <http://jupyter.org/>

Seção 2.2.

Figura 7: *Pipeline* de Dados



Fonte: Adaptado de (GOODFELLOW; BENGIO; COURVILLE, 2016)

A Figura 7 possui dois *pipelines* de dados, um caracterizando o processo de geração do modelo preditivo utilizando aprendizado de máquina convencional, e outro utilizando *deep learning*. No canto superior direito, pode-se observar uma legenda que representa o aprendizado da máquina, o que caracteriza as etapas em que o algoritmo aprende. No caso da abordagem 1, esse processo de aprendizagem ocorre na etapa de modelagem. Na abordagem 2, esse processo ocorre em todas as etapas entre a entrada e saída. A abordagem 1, ilustrada no diagrama superior, compreende as seguintes etapas:

- Entrada: os dados que vão caracterizar as *features*.
- Projeção Manual de *Features*: especificação do formato dos dados e geração de novas *features*. Por exemplo, média de acessos semanais na plataforma Moodle. Esta etapa pode garantir bons resultados na tarefa de aprendizagem.
- Modelagem: processo de treinamento dos modelos preditivos.
- Saída: modelo preditivo.

Salienta-se que, quando o modelo preditivo é gerado, ele precisa ser validado. Para isso, utilizam-se diversas métricas de validação que estão presentes na literatura (as métricas de validação desta dissertação serão discutidas posteriormente nesta Seção). Na abordagem 2, presente no diagrama inferior da. Figura 7, encontram-se as seguintes etapas:

- Entrada: os dados que vão caracterizar as *features*.
- *Features* Simples: em redes neurais, normalmente os dados são utilizados em seu formato bruto, sem a necessidade de um pré-processamento. Em alguns casos, é importante estabelecer uma padronização no *dataset*, para que os dados sejam

processados de maneira correta. Então, as *features* simples caracterizam a vantagem de não precisar gastar muito tempo com a preparação dos dados.

- *Features* Complexas: adicionar mais camadas na rede para obter *features* mais abstratas sobre os dados de entrada.

- Modelagem: processo de treinamento do modelo preditivo.

- Saída: modelo preditivo.

Com base no entendimento das etapas ilustradas na Figura 7, este Capítulo irá abordar os procedimentos metodológicos para a implementação de ambas as abordagens. Primeiro, foi realizada uma Seção para descrever o *dataset* utilizado (Seção 3.1), depois serão retratadas as etapas e ferramentas utilizadas para o desenvolvimento da primeira abordagem na Seção 3.2, e, na Seção 3.3, serão apresentadas as etapas e ferramentas utilizadas para a segunda abordagem.

3.1 *Dataset*

O *dataset* foi construído com dados extraídos de dois programas de Pós-Graduação EaD da Universidade Federal do Rio Grande (FURG). Os cursos nomeados de Aplicações para a Web e Tecnologias da Informação e Comunicação na Educação (TIC-EDU), apresentaram altos índices de evasão até o momento da pesquisa, sendo 38,88% e 59,21%, como ilustrado na Tabela 5. Nesse trabalho, os cursos são tratados como Curso 1 (TIC-EDU) e Curso 2 (Aplicações para a Web). 264151,

Tabela 5: Índices de Evasão por Curso e Módulo

	Curso 1 - 90 Alunos	**Curso 2 - 76 Alunos**
Módulo 1	20	32
Módulo 2	8	8
Módulo 3 e 4	7	5
Somatório	35 (38,88%)	45 (59,21%)

Como pode-se observar, a Tabela 5 mostra o número de alunos matriculados nas colunas e o número de abandonos a cada módulo. Os cursos são da Pós-Graduação EaD, fomentados pelo Sistema UAB, e são compostos por quatro módulos. O curso teve início em março de 2018, e, pode-se notar que, logo no primeiro módulo, a quantidade de alunos que abandonaram o curso é muito grande. Logo no início do Curso 1, cerca de 22,22% (20) dos estudantes desistiram do curso. No Curso 2, houve uma taxa de evasão de 42,10% (32) logo no início do curso. Em sua totalidade, os dois cursos apresentaram um índice de 48,19% (80), sendo que 38,88% destes alunos eram do Curso 1, e 59,21%

eram do Curso 2. Nota-se que o índice de evasão no Curso 2 é maior, e trata-se de um curso com menos alunos matriculados, e, por isso, o impacto acaba sendo maior.

O *dataset* que foi utilizado para o treinamento dos algoritmos foi construído com os dados coletados de ambos os cursos, que dispõe de 470.401 registros de interações de alunos com a plataforma Moodle. Esses registros estão disponíveis nos relatórios dos cursos, e são chamados de logs. Na Tabela 6 é possível visualizar a estrutura desses relatórios, e quais são os campos de dados que eles possuem.

Tabela 6: *Logs* de Dados

Atributos	**Descrição**
Curso	Nome do curso (*e.g.*, Curso 1).
Hora	Data e horário em que a ação ocorreu.
Endereço IP	O endereço IP do computador que realizou a ação.
Nome Completo	O nome completo do usuário que realizou a ação.
Ação	A ação executada pelo usuário (*e.g.*, Submissão de tarefa).
Informação	Descrição da ação ocorrida (*e.g.*, Tarefa 1 - Relatório Final).

Os atributos da Tabela 6 são: o Curso, que contém o nome e, também, a denominação da disciplina; o Nome Completo, que corresponde a identificação do aluno, tutor, e qualquer outro usuário que tenha acessado o curso; a Ação, que categoriza a interação efetuada; por fim, a Informação carrega detalhes sobre esta ação. Salienta-se que o atributo Nome Completo foi anonimizado por razões de proteção de privacidade, e assim, transformado em um pseudo de identificação (ID).

Selecionaram-se dez ações diferentes de um total de quarenta e três ações observadas no *dataset*. As ações selecionadas foram usadas como *features*, e podem ser visualizadas na Tabela 7.

Tabela 7: *Features* Selecionadas

Atributos	**Descrição**
course_view	Visualização do curso.
forum_view	Visualização do fórum.
forum_view_discussion	Visualização de discussão no fórum.
resource_view	Visualização de um recurso.
forum_add_post	Adição de uma publicação ao fórum.
forum_add_disc	Adição de discussão no fórum.
assign_view	Visualização da atividade.
assign_sub	Submissão de atividade.
user_view	Visualização de usuário.
url_view	Visualização de um endereço de um recurso.

Para fazer essa seleção, a covariância foi calculada como uma medida de interdependência entre as *features*. A covariância próxima de zero foi usada como valor limite para

a remoção de *features*. Em alguns casos, as *features* que exibiram uma ocorrência extremamente baixa, ou muitos dados nulos, foram removidos. Na Tabela 7 pode-se visualizar o conjunto selecionado de *features*, e suas respectivas descrições. Evadiu é a variável escolhida como alvo e, também, acrescentaram-se outras duas *features*: polo (localização) e curso (Curso 1 e Curso 2).

Utilizar apenas os logs de dados para treinar algoritmos de aprendizado de máquina pode parecer insuficiente, e cogitou-se o plano de utilizar, também, dados demográficos e de desempenho acadêmico (*e.g.*, notas de trabalhos e provas). Porém, para utilizar o modelo preditivo em um sistema de alerta de risco que funcione em tempo real, seria necessário o uso constante dos dados. Isto é, enquanto os estudantes utilizam o Moodle, o sistema faria uma leitura destes registros para manter o *status* do aluno (*e.g.*, em risco, sem risco).

Para determinar quais alunos foram classificados nas classes de evadido e não evadido (regular), uma investigação mais aprofundada foi realizada. Os registros de acesso ao Moodle gravam a data e hora da última visita que o usuário fizeram ao AVA. A regra aplicada considera um aluno evadido nos casos em que este fica sem acessar o AVA Moodle por mais de 30 dias. Outro ponto de corte foi definido pela não visualização de um aluno no módulo posterior ao que ele se encontrava. Quando um aluno matriculado em um módulo não acessou em 30 dias o próximo módulo do programa, este caso também foi considerado um abandono. Esse critério foi validado com os coordenadores dos cursos baseando-se em experiências anteriores.

3.2 Abordagem 1 - Aprendizado de Máquina Convencional

Nesta Seção, será retratado como foi elaborada a preparação dos dados para a execução da primeira abordagem na Seção 3.2.1, quais foram os algoritmos utilizados na etapa de modelagem na Seção 3.2.2, e quais foram as métricas avaliativas utilizadas na Seção 3.2.3.

3.2.1 Preparação dos Dados Para a Abordagem 1

A preparação dos dados é uma etapa importante para se obter bons resultados com o modelo preditivo. Para isso, um *pipeline* de processamento de dados foi definido com as seguintes operações: limpeza e estruturação do *dataset*, anonimização e transformação de dados (operações de *wrangler*[2]).

Para a limpeza do *dataset*, foi realizada a eliminação de duplicatas, inconsistências em nomes de *features* (*e.g.*, espaço entre palavras), dados nulos e recursos dispensáveis foram removidos. Os dados de formato bruto de texto foram transformados em dados numéricos. Assim, seria possível calcular frequências de acesso e interações no Moodle

[2]Técnicas utilizadas para conversão dos dados brutos para um formato executável

por parte dos estudantes. Salienta-se que os nomes dos estudantes foram anonimizados para proteção de identificação.

3.2.2 Modelagem Para a Abordagem 1

Para o treinamento dos modelos preditivos, foram utilizados oito algoritmos de aprendizado de máquina, que foram explorados em trabalhos relacionados (KOSTOPOULOS et al., 2018) (DEWAN et al., 2015) (RAMOS et al., 2017). Os algoritmos selecionados são: *k-Nearest Neighbors* (KNN), *Gaussian Naive Bayes* (NB), *C-Support Vector Classification* (SVC), *Logistic Regression* (LR), *Random Forest* (RF), *Adaptive Boosting* (*AdaBoost*), *Gradient Boosting* e *Extremely Randomized Trees* (*Extra Trees*).

- *k-Nearest Neighbors*: de acordo com (SHALEV-SHWARTZ; BEN-DAVID, 2014), o conceito central que idealiza o KNN é assumir que elementos semelhantes devem ser iguais. Para um melhor entendimento, quando o algoritmo processa a entrada de uma nova instância, a similaridade dessa nova instância com seus vizinhos mais próximos na base de treinamento é verificada. Desta forma, a classificação é dada com base na similaridade do vizinho mais próximo.

- *Gaussian Naive Bayes*: o NB é um classificador probabilístico baseado no Teorema de *Bayes*. Conforme SHALEV-SHWARTZ; BEN-DAVID (2014), o algoritmo NB é uma demonstração clássica de como pressupostos generativos e estimativas de parâmetros simplificam o processo de aprendizagem. Neste estudo, o NB Gaussiano foi utilizado.

- *C-Support Vector Classification*: SVC é uma subclasse dos métodos *Support Vector Machines* (SVM) que aborda os desafios complexos da amostra que procura por grandes separadores de margem (SHALEV-SHWARTZ; BEN-DAVID, 2014). Ou seja, os dados de treinamento são separados por meio de uma linha de separação (hiperplano), que busca maximizar a distância entre os pontos mais próximos e classes distintas. A margem significativa é a distância entre o hiperplano e o primeiro ponto de cada classe.

- *Logistic Regression*: o algoritmo LR pertence à família de métodos lineares que predizem o resultado da variável de destino com base em uma ou mais variáveis preditoras (JAYAPRAKASH et al., 2014). O LR possui uma função logística integrada que captura a combinação linear das variáveis preditoras, modelando a probabilidade de ocorrência de um valor particular da variável alvo.

- *Random Forest*: RF é um classificador composto por uma coleção de árvores de decisão, em que cada uma dessas árvores é construída a partir de um algoritmo A aplicado a um conjunto de treinamento S e em um vetor aleatório, onde é amostrado aleatoriamente a partir de alguma distribuição. A predição do *random forest* é

obtida através de uma votação majoritária sobre as previsões de árvores individuais. Os autores (SHALEV-SHWARTZ; BEN-DAVID, 2014) apontaram que o RF ajuda a evitar *overfitting* porque explora um conjunto de árvores de decisão e não apenas uma única.

- *AdaBoost*: este algoritmo pertence à família de algoritmos *Boosting*, que usa uma generalização de preditores lineares para tratar de duas questões principais, *bias-variance* e a complexidade computacional do processo de aprendizagem (SHALEV-SHWARTZ; BEN-DAVID, 2014). O *AdaBoost* produz uma hipótese definida por uma combinação linear de hipóteses simples. Em outras palavras, conta com a família de classes de hipóteses obtidas pela composição de um preditor linear baseado nas classes simples.

- *Gradient Boosting*: Segundo RIDGEWAY (1999), esse algoritmo originou-se da conexão entre Otimização e *Boosting*. O *Gradient Boosting* determina a direção, o texto, em cada interação realizada pelo algoritmo. Dessa forma, o algoritmo precisa melhorar o ajuste de dados e selecionar um determinado modelo da classe de funções permitida que esteja mais de acordo com a direção (RIDGEWAY, 1999).

- *Extra Trees*: este algoritmo foi introduzido por (GEURTS; ERNST; WEHENKEL, 2006), que relata que o algoritmo constrói um conjunto de árvores de decisão, ou regressão, que é isento com base no procedimento padrão de cima para baixo. Suas duas principais diferenças, comparadas a outros métodos de agrupamento baseados em árvore, são que o *Extra Trees* divide os nós escolhendo pontos de corte muito aleatórios e usando toda a amostra de aprendizado para cultivar árvores.

Nesta abordagem, o treinamento dos modelos foi realizado utilizando *Stratified 10-fold cross-validation*, que consiste em dividir o *dataset* em dez partições menores, usando parte dos dados para treinamento, e o restante para validação, até que todas as partições menores sejam utilizadas para treinar e validar o modelo. Neste caso, foi definido um tamanho de 20% para o *dataset* de validação. Ao final do processo de treinamento, a acurácia é calculada a partir da média de acertos entre todas as partições.

3.2.3 Validação dos Modelos Para a Abordagem 1

Para validar o desempenho dos modelos, foram medidas a Acurácia, AUROC (*Area Under the ROC curve*), *Precision*, *Recall*, F1 *score* e as Curvas de Aprendizagem para a visualização gráfica dos resultados, que foram definidas na Seção 2.2.1. A utilização de várias métricas de avaliação pode garantir uma confiabilidade maior dos resultados, e verificar se existe sobreajuste nos modelos (*overfitting*).

Essas métricas fazem uso de quatro medidas geradas pelos algoritmos de classificação. Estas medidas são os Verdadeiros Positivos (VP), Verdadeiros Negativos (VN), Falsos

Positivos (FP) e Falsos Negativos (FN). Os Verdadeiros Positivos e Verdadeiros Negativos representam o índice de classificações corretas. Isto é, um positivo (aquele que evadiu) ser classificado como evadido, e um negativo (o aluno que não evadiu) ser classificado como tal. Os Falsos Positivos e Falsos Negativos representam o índice de classificações incorretas, ou seja, quando um positivo é classificado como negativo (e vice-versa).

A Acurácia é a métrica de desempenho mais usada para classificação (PROVOST et al., 1998). Ela é estimada dividindo o total de positivos e negativos classificados corretamente, pelo total de observações no conjunto de validação (OLSON; DELEN, 2008), ou seja, as classificações corretas e incorretas, como pode ser observado na fórmula (1).

$$\frac{VP + VN}{VP + VN + FP + FN} \tag{1}$$

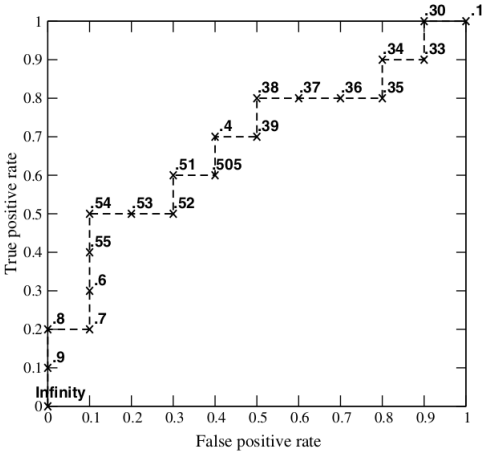
A curva ROC (*Receiver Operating Characteristic*) apresenta um sistema de coordenadas para a visualização do desempenho do modelo. Nesse espaço métrico, a taxa de VP (TVP) é plotada no eixo Y e a taxa de FP (TFP) é representada no eixo X. Como pode-se observar na Figura 16, um ponto no espaço ROC (TFP, TVP) corresponde a um desempenho de classificador de modelo. A curva resultante ilustra o erro balanceado para um determinado modelo, em que o comportamento preditivo capturado não está vinculado a nenhum custo de distribuição ou erro de classe. A AUROC (*Area Under the ROC Curve*) calcula a relação geral entre a taxa de VP, também conhecida como *sensitivity* ou *recall*, e a taxa de FP, também conhecida como *Fall-out*. A última medida também é calculada como 1-*specificity*.

Figura 8: Curva ROC

[1]

Inst#	Class	Score	Inst#	Class	Score
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	n	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	20	n	.1

[2]



Fonte: (FAWCETT, 2006)

Precision: representa a taxa de predição correta dentro da classe positiva esperada. *Precision* é calculado através da fórmula ilustrada em (2) (SHALEV-SHWARTZ; BEN-DAVID, 2014).

$$\frac{VP}{VP + FP} \tag{2}$$

Recall: também conhecido como *sensitivity*, essa métrica representa a TVP, conforme previsto pelo modelo de classificação (SHALEV-SHWARTZ; BEN-DAVID, 2014). O *recall* é dado pela fórmula (3).

$$\frac{VP}{VP + FN} \tag{3}$$

F1 score: também conhecido como *F-score* ou *F-measure*, avalia a precisão obtida através do conjunto de validação. O *F1 score* representa a média harmônica entre *precision* e *recall*, e a fórmula a fórmula é representada por (4) (SHALEV-SHWARTZ; BEN-DAVID, 2014). O valor máximo de *F1 score* de 1 é obtido quando *precision* e *recall* são iguais a 1, e seu valor mínimo de 0 é atingido sempre que uma das duas métricas é 0.

$$2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

O Jupyter Notebook facilitou o desenvolvimento do trabalho devido suas ferramentas, bibliotecas e a facilidade de criar relatórios e gráficos em meio a cada operação realizada. Isso foi bastante útil na etapa de preparação dos dados, pois pode-se visualizar as funções executadas no *dataset* graças a capacidade de execução de partes do código, independentemente do restante do notebook. As principais bibliotecas do Python utilizadas foram o Pandas, Matplotlib e Scikit Learn. O Pandas possibilita a manipulação dos dados, e é utilizado praticamente em todos os trechos de código. O Matplotlib foi usado para a geração das visualizações gráficas, e o Scikit Learn foi a biblioteca utilizada para o aprendizado de máquina. Com o Scikit Learn, foram importados os algoritmos mencionados na Seção de modelagem e as as métricas de validação dos modelos.

3.3 Abordagem 2 - *Deep Learning*

Nesta Seção, apresentam-se as etapas e procedimentos metodológicos utilizados para a execução da Abordagem 2. Na Seção 3.3.1, apresentam-se os procedimentos utilizados para a preparação do *dataset*. Na Seção 3.3.2 está definida a rede neural utilizada para a execução desta abordagem, e, na Seção 3.3.3, disserta-se sobre as métricas utilizadas para validar os modelos preditivos.

3.3.1 Preparação dos Dados Para a Abordagem 2

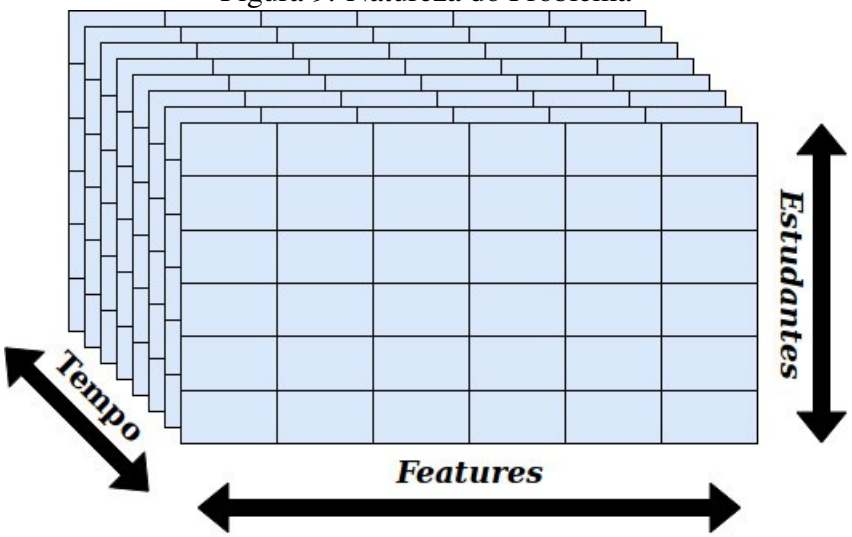
Os dados extraídos do ambiente Moodle caracterizam os rastros digitais dos estudantes ao longo do curso. Estes registros são armazenados em uma série temporal, carregando informações referentes às ações executadas pelos usuários no seu exato período de tempo. Assim, o problema de interesse é encontrar padrões nestes dados de cunho temporal, de forma a caracterizar o comportamento de evasão do aluno ao longo de sua experiência acadêmica. Com isto, faz-se necessário utilizar um *dataset* representativo, com registros

que representam o contato dos estudantes com objetos de aprendizagem no Moodle na dimensão tempo (*e.g.*, do primeiro dia de aula, até a data de encerramento do curso).

A cada passo de tempo (*time steps*), as medições registradas serão usadas como entrada, e uma probabilidade de predição de evasão será gerada. É importante notar que isso permite um monitoramento em tempo real da probabilidade de evasão do aluno, assim como um entendimento de sua trajetória.

Nesta abordagem, será utilizado o atributo "Hora"descrito na Tabela 6, para modelar o comportamento dos alunos ao longo do curso. O tempo entre medições pode variar de minutos a horas. Um diagrama simplificado dos dados pode ser visto na Figura 9.

Figura 9: Natureza do Problema



A Figura 9 representa uma série temporal em que, para cada estudante, existe um número X de observações. No eixo *x*, estão as *features* do *dataset*, que correspondem as 10 ações selecionadas + o curso e polo, totalizando 12. No eixo *z*, tem-se os 166 estudantes com suas respectivas classificações (1 para evadiu e 0 para não evadiu). Por fim, o eixo *y* equivale a profundidade que, neste caso, é o tempo (*time steps*). Salienta-se que os estudantes possuem diferentes quantidades de observações, então o valor de X varia constantemente. Na Tabela 8 pode-se ter uma percepção de como é a estrutura deste *dataset*.

O *dataset* presente na Tabela 8 possui multi-índices, sendo o ID do estudante um, e os *time steps* outro. As *features* caracterizam uma determinada ação, a qual possui valor de 1 quando esta determinada ação é efetuada, e 0 quando não há registro. Ao final, localiza-se a situação do estudante, sendo 1 para evadido e 0 para regular (quando não evadido).

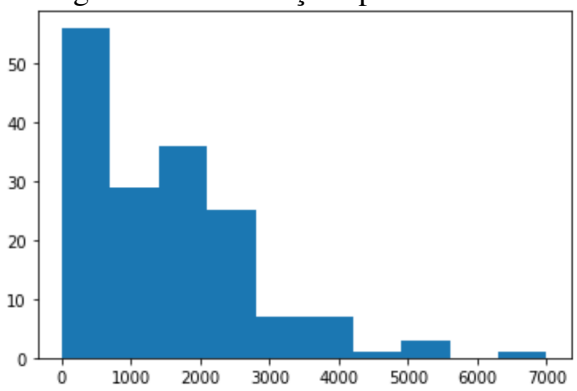
Utilizou-se a função pad_sequences do Keras para padronizar o número de observações por estudantes. Keras é uma biblioteca de redes neurais escrita em Python. O Keras é capaz de rodar sobre os *frameworks* TensorFlow ou Theano. O Tensorflow é o *framework* utilizado nesta abordagem.

Tabela 8: *Dataset* como Série Temporal

id	tempo	acao_1	acao_2	acao_3	...	evadiu
33	10-04-2018 22:22	1	0	0	...	1
	10-04-2018 22:23	0	1	0	...	1
	10-04-2018 22:25	0	0	1	...	1
	...	...	...	...	...	1
34	10-05-2018 12:34	0	0	1	...	0
	10-05-2018 12:37	1	0	0	...	0
	...	...	...	...	...	0

Na Figura 10, pode-se observar que o número de *time steps* para alguns alunos chega até cerca de sete mil. A média calculada de observações foi de 1473, a qual foi utilizada como valor máximo na função de sequenciamento.

Figura 10: Observações por Estudante



Padronizar o número de *time steps* é necessário, porque a rede descrita na Seção 3.3.2 espera um formato de três dimensões devido a alguns componentes presentes nas camadas. Com a função de sequenciamento, o *dataset* ficou com o formato de (166, 1473, 12).

3.3.2 Modelagem Para a Abordagem 2

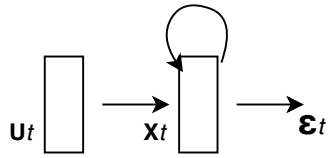
Nesta Seção, será definida a rede neural utilizada no processo de modelagem, que foi implementada com o suporte da biblioteca Keras de *deep learning*. Assim, será abordado a *Recurrent Neural Network* (RNN) com a arquitetura de uma *Long Short-Term Memory* (LSTM) na Seção 3.3.2.1.

3.3.2.1 Recurrent Neural Network

A Recurrent Neural Network (RNN), segundo LUKOšEVIčIUS; JAEGER (2009), representa uma classe de modelos computacionais projetados para serem semelhantes à módulos cerebrais biológicos. Nestes modelos computacionais, existe uma série de neurônios interconectados por conexões sinápticas, que possibilita a propagação de sinais através de ativações na rede. A principal característica de uma RNN é a presença de

pelo menos uma conexão de *feedback* para que as ativações possam fluir em um *loop*, o que permite que a rede aprenda sequências por meio de um processamento temporal (BULLINARIA, 2013), como ilustrado na Figura 11.

Figura 11: Esquema de uma *Recurrent Neural Network*



Fonte: PASCANU; MIKOLOV; BENGIO (2012)

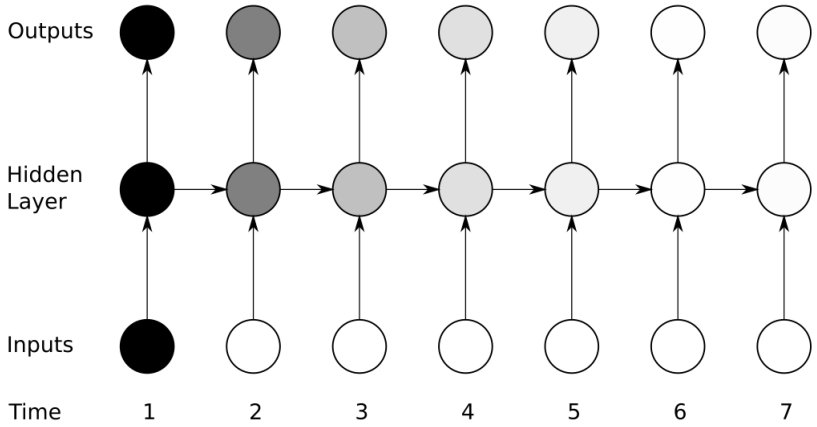
Na Figura 11, é ilustrado um simples diagrama para demonstrar o funcionamento de uma RNN. As flechas representam o fluxo dos dados na rede. As conexões recorrentes na camada oculta permitem que as informações persistam de uma entrada para outra. De acordo com LUKOšEVIčIUS; JAEGER (2009), a presença destes *loops* tem um papel fundamental:

- Mesmo na ausência de uma entrada, uma RNN pode desenvolver um processo de ativação temporal auto-sustentado ao longo de suas trilhas de conexão recorrentes, o que a torna um sistema dinâmico, diferente do tipo de rede *feedforward*, que representam funções.
- Quando acionada por um sinal de entrada, a RNN preserva em seu estado interno uma transformação não linear do histórico de entrada. A rede possui uma memória dinâmica capaz de processar informações de natureza temporal.

Com esse entendimento, as RNN são utilizadas nesta abordagem devido sua capacidade de armazenar entradas passadas, para produzir a saída desejada (HOCHREITER, 1998), o que é importante para predição em séries temporais. De acordo com HOCHREITER (1998), aplicações envolvendo dependências temporais abrangem muitas etapas entre a entrada de dados e o resultado desejado, e isso toma muito tempo. As RNN devem aprender quais entradas anteriores devem ser armazenadas para produzir a saída desejada atual. Em métodos de aprendizado baseados em gradiente, o sinal de erro atual tem que voltar no tempo através das conexões de *feedback*, para entradas passadas e, então, conseguir construir um armazenamento de entradas adequado (HOCHREITER, 1998). O problema é que esses sinais de erro tendem a desaparecer ao fluir de volta no tempo, o que é conhecido como *Vanishing Gradient Problem*, como ilustrado na Figura 12.

Como pode ser observado na Figura 12, os nodos que representam a primeira entrada da rede estão coloridos em preto, enquanto que o sombreamento presente nos outros nodos indica a sensibilidade referente a essa entrada. Na medida em que a rede processa novas entradas, essa sensibilidade passa a diminuir, e a rede vai esquecendo informações referente as primeiras entradas. Essa sensibilidade que vem diminuindo com o tempo

Figura 12: *Vanishing Gradient Problem*



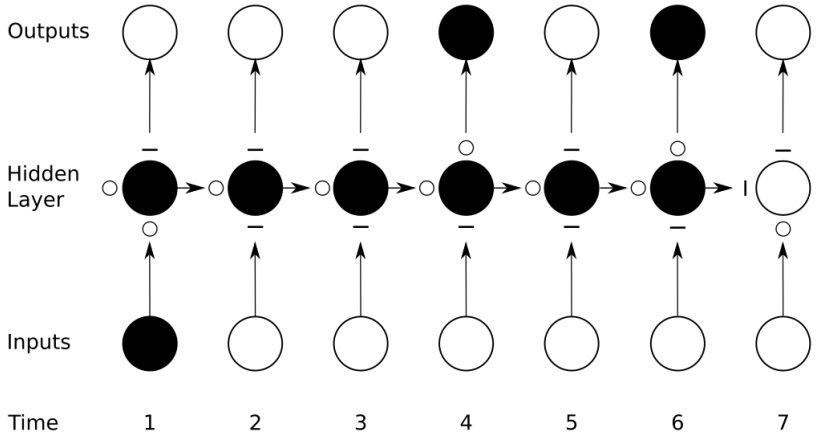
Fonte: (GRAVES, 2012)

representa o gradiente. Grande parte dos algoritmos de treinamento são baseados em gradiente, como o *Backprogation Through Time* (PASCANU; MIKOLOV; BENGIO, 2012). Segundo PASCANU; MIKOLOV; BENGIO (2012), a maioria desses métodos baseados em gradiente se comporta basicamente da mesma forma, revelando pouco sucesso ao lidar adequadamente com tarefas mais complexas. Trabalhar com dependências de longo prazo são tarefas difíceis devido sua mudanças de pesos (HOCHREITER, 1998). Nesse sentido, surgiu a LSTM, como um modelo eficaz para vários problemas de aprendizagem relacionados a dados sequenciais (GREFF et al., 2016).

As LSTM RNN demonstraram-se úteis no processo de aprendizagem de padrões em sequências de dados de comprimento indeterminado, devido a sua capacidade de armazenar informações (MALHOTRA et al., 2015). Isso torna-se possível devido as unidades especiais na camada oculta recorrente da rede, conhecidas como blocos de memória (SAK; SENIOR; BEAUFAYS, 2014). Estes blocos de memória contêm células de memória com auto-conexões que armazenam o estado temporal da rede, assim como unidades multiplicativas especiais chamadas de *gates*, para controlar o fluxo de informações. Assim, por meio dos *gates* é possível armazenar e acessar informações nas células de memória de uma LSTM em escalas temporais mais longas, mitigando o problema de *vanishing gradient* (GRAVES, 2012). Esta concepção pode ser visualizada na Figura 13.

No cenário apresentado pela Figura 13, nota-se que a sensibilidade permanece alta ao longo do tempo, ou seja, a informação do gradiente permanece disponível até que a mesma utilizada. Também, é possível visualizar os *gates* de controle de fluxo nos seus dois estados: aberto 'o' e fechado '-'. Desta forma, a célula de memória ira armazenar a informação referente a primeira entrada, desde que o *gate* de esquecimento esteja aberto, e o *gate* de entrada esteja fechado (GRAVES, 2012). Cada bloco de memória contém um *gate* de entrada e um *gate* de saída, sendo que o *gate* de entrada controla o fluxo de ativações de entrada na célula de memória, e o *gate* de saída controla o fluxo de saída das ativações de células para o restante da rede (SAK; SENIOR; BEAUFAYS, 2014). Nas

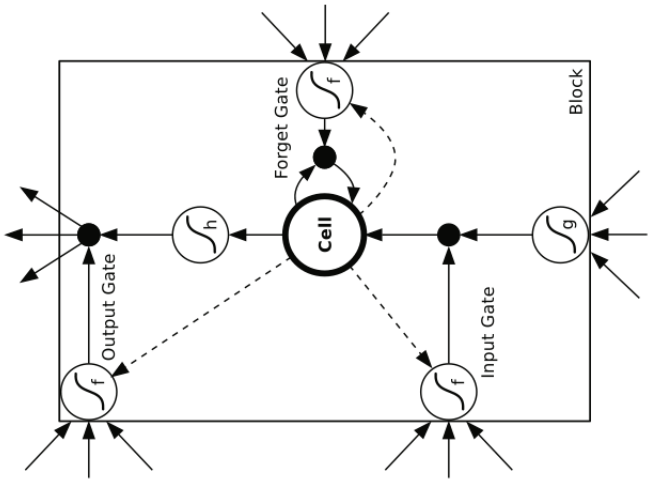
Figura 13: Preservação do Gradiente com a LSTM



Fonte: (GRAVES, 2012)

arquiteturas de LSTM mais recentes, acrescentou-se um *gate* de esquecimento no bloco de memória, para impossibilitar que fluxos de entrada contínuos, que não são segmentados em subsequências, fossem processados (SAK; SENIOR; BEAUFAYS, 2014). O bloco de memória pode ser visualizado na Figura 14.

Figura 14: Bloco de Memória de um LSTM



Fonte: (GRAVES, 2012)

O bloco de memória ilustrado na Figura 14 contém uma célula de memória, os *gates* de entrada, saída e esquecimento. Os *gates*, segundo GRAVES (2012), são unidades de soma não lineares, que coletam ativações de dentro e de fora do bloco, e controlam a ativação da célula por meio de multiplicações, caracterizados pelos pequenos círculos pretos. A letra f nos *gates* de entrada, saída e esquecimento, representam a função de ativação do fluxo de dados (geralmente a *logistic sigmoid*), que vai de 0 (fechado) a 1 (aberto). As letras h e g representam as funções de ativação do fluxo de dados dentro da célula de memória, que normalmente são a *tanh* ou *logistic sigmoid* (GRAVES, 2012).

3.3.3 Validação dos Modelos Para a Abordagem 2

Para validar o desempenho da LSTM, foi utilizada a métrica de acurácia e *loss* (também conhecida como custo ou perda). Essa função aponta o quão longe o estimador está da predição correta (PONTI; COSTA, 2018). Assim, quanto menor for o custo, melhor vai ser o desempenho do modelo.

Ambas as métricas tiveram seus valores ilustrados em um gráfico para visualização das curvas de treinamento e validação. Assim, pode-se analisar se a rede apresentaria um desempenho melhor com a adição de mais exemplos de treinamento, e se o estimador sofre mais com erro de variância ou *bias*.

4 RESULTADOS

Esse capítulo apresenta os resultados de execução das duas abordagens propostas. Foi realizado um primeiro experimento (Seção 4.1 que caracteriza o entendimento do problema, dos dados e do funcionamento dos algoritmos, sem estabelecer um ponto de corte para o treinamento dos modelos (*i.e.*, definir o momento em que a predição deve ocorrer). Houve o Experimento B (Seção 4.2) em que foram utilizados mais dados, e estabeleceram-se pontos de corte semanais. Por fim, o Experimento C (Seção 4.3), que marca a execução da Abordagem 2 do trabalho.

4.1 Experimento A - Entendimento do Problema

O objetivo deste experimento foi avaliar o desempenho dos algoritmos de aprendizado de máquina descritos na Seção 3.2.2 quando aplicados sobre o *dataset* disponível no Moodle. Também, poder entender os dados, a sua natureza, e aprender a lidar com eles de tal forma que possam ser utilizados para o aprendizado de máquina.

Salienta-se que este experimento foi elaborado em um período anterior ao do experimento B, e, por esta razão, não havia a disponibilização de todos os dados presentes na Seção 3.1 (*Datasets*). A Tabela 9 fornece uma visão geral dos registros de dados que foram extraídos da plataforma Moodle referentes ao primeiro módulo dos cursos mencionados em 3.1, e que foram utilizados neste experimento.

Tabela 9: Registros Coletados no Experimento A

	Curso 1	**Curso 2**	**Total**
Registros	115.407	84.762	200.166
Estudantes	90	76	166
Estudantes Evadidos	20	31	51
Estudantes Não Evadidos	70	45	115

O *dataset* dispunha de mais de 200 mil dados que caracterizavam o perfil de cada um dos 166 alunos em uma escala temporal. Esses dados foram então transformados em um conjunto de *features* que pode ser visualizado na Tabela 10.

Cada linha na Tabela representa um aluno e sua média de interações com o curso. Por

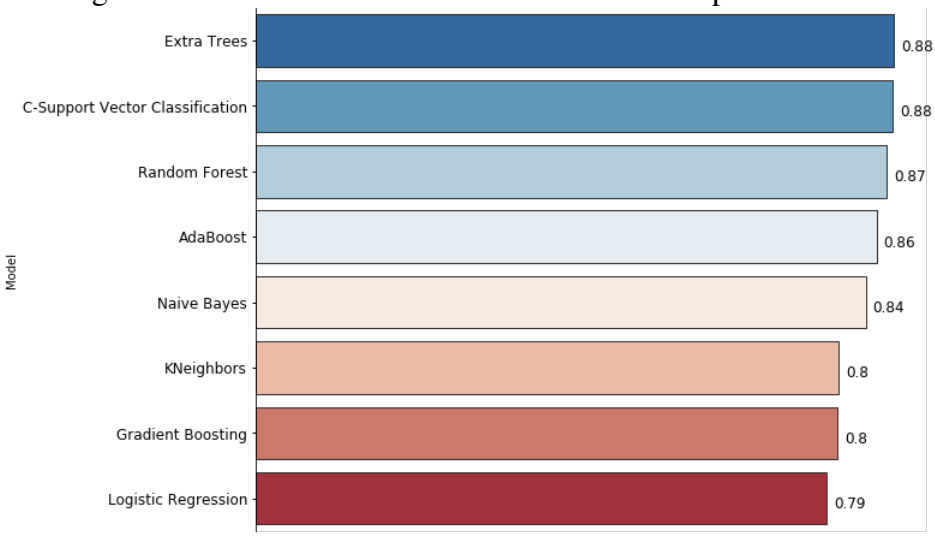
Tabela 10: Preparação dos Dados no Experimento A

id	course_view	forum_view	resource_view	...	dropout
0	0.393056	0.151389	0.001389	...	0
1	0.334171	0.120101	0.041709	...	0
3	0.363395	0.151194	0.010610	...	0

exemplo, o aluno representado pelo ID 3 na Tabela 10, teve uma média de interação de 0.36% com *course_view* (visualização do curso), 0.15% de visitas ao fórum, e assim por diante. No final da Tabela, tem-se a classificação do estado daquele aluno: 1 se ele evadiu, ou 0 se ele não evadiu.

Após concluir o pré-processamento dos dados, realizou-se o treinamento dos algoritmos KNN, NB, SVC, LR, RF, AdaBoost, *Gradient Boosting* e *Extra Trees*. A acurácia dos modelos gerados é apresentada em ordem decrescente na Figura 15. Todos os algoritmos usados no treinamento do modelo alcançaram uma alta acurácia, com uma média de 84%. A melhor acurácia foi de 88%, e é fornecida pelo classificador *Extra Trees*. O LR teve o menor desempenho, com uma acurácia de 79%.

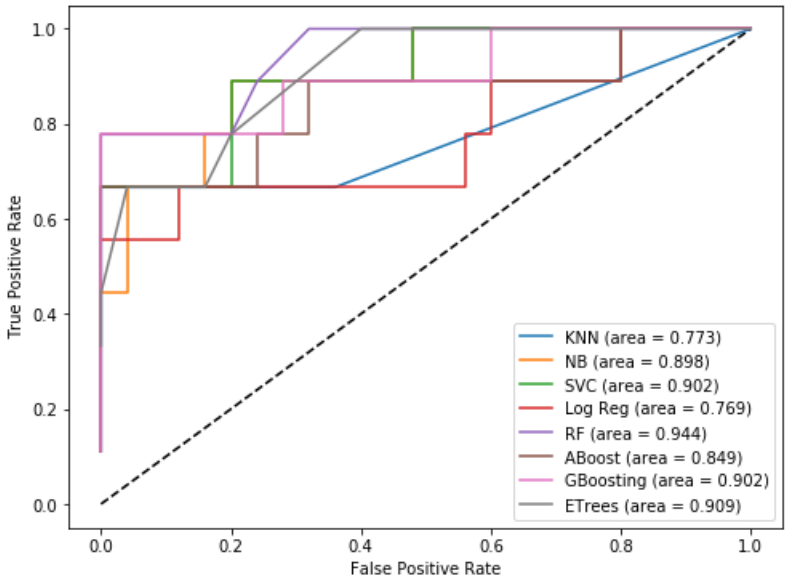
Figura 15: Acurácia dos Modelos Treinados no Experimento A



A AUROC foi usada para quantificar a capacidade do classificador de distinguir entre as duas classes de interesse. A figura 16 apresenta a TFP no eixo x, e a TVP no eixo y. Metade dos modelos mostrados na Figura 16 exibem um AUROC maior que 90%. O modelo RF atingiu um AUROC de 94%. O menor grau de separação de rendimento foi no modelo LR, que também obteve a menor acurácia entre os modelos validados.

Os resultados obtidos pelas métricas *Precision*, *Recall* e F1 *score* podem ser visualizados na Tabela 11. Os resultados observados para cada modelo apresentam pequenas variações entre essas métricas. O algoritmo Extra Trees alcançou a maior média de *recall* com o valor de 0,94. A média mínima de 0,79 foi obtida através do algoritmo AdaBoost. Os algoritmos KNN, NB, SVC e principalmente LR sofreram grandes variações entre os

Figura 16: Curvas ROCS



valores de 0 e 1. Nesses casos, a classe 1 (positiva) exibiu valores de *recall* rasos, em comparação com a classe 0 (negativa). Isso sugere que esses algoritmos não foram eficientes para classificar a classe positiva (1 - Evadiu).

Tabela 11: Recall, Precision e F1 score

Algorithms	Recall	Precision	F1 score
k-Nearest Neighbors	0.91	0.92	0.91
Gaussian Naive Bayes	0.85	0.85	0.85
C-Support Vector Classification	0.91	0.92	0.91
Logistic Regression	0.88	0.90	0.87
Random Forest	0.91	0.91	0.91
AdaBoost	0.79	0.80	0.80
Gradient Boosting	0.91	0.91	0.91
Extra Trees	0.94	0.95	0.94

Em relação ao *precision*, não encontrou-se muita variação entre os valores das classes 0 e 1. Em alguns classificadores, a classe positiva teve um desempenho melhor que a classe negativa e vice-versa. A maioria dos algoritmos obteve uma média de *precision* razoável na classificação correta de ambas as classes. Uma exceção é o classificador AdaBoost, que alcançou um valor de 0,60 na classe positiva (não informado na Tabela), o que indica um número de FP alto.

O F1 *score* também apresentou um desequilíbrio entre entre as classes 0 e 1, em que a classe 1 possui as menores porcentagens. Esta informação é alarmante porque a classe 1 é a mais importante nesta pesquisa, pois representa aqueles que abandonaram o curso. Os melhores resultados foram alcançados através do algoritmo *Extra Trees*.

Comparando os resultados com trabalhos relacionados que utilizaram as mesmas mé-

tricas para validar os modelos preditivos gerados, em (RAMOS et al., 2018) foram utilizados os algoritmos LR, KNN e o SVM, os valores de *recall* para esses algoritmos foi de 0.61% para o LR, 0.56% para o KNN e 0.36% para o SVM. Em relação ao *precision*, os valores são 0.71% para o LR, 0.75% para o KNN e 0.86% para o SVM, o qual obteve o melhor valor. Os autores também utilizaram a métrica AUROC, que obteve um percentual de 0.85% para o LR, 0.79% para o KNN e 0.79% para o SVM. O SVM foi um dos algoritmos utilizados no trabalho de (PEÑA et al., 2017) e (BURGOS et al., 2018), nos quais apresentou-se um *recall* de 0.69% e *precision* de 0.36% (os autores estão presentes nos dois trabalhos, o que explica a duplicação dos resultados). Os resultados apresentados por essas pesquisas são inferiores em algumas métricas e algoritmos aos presentes nesta dissertação. No entanto, vale salientar que a complexidade das *features* nesses trabalhos é completamente diferente. Essa questão também se remete a etapa de Projeção Manual de *Features*, em que o desempenho dos algoritmos depende da maneira como as *features* estão representadas.

4.2 Experimento B - Ponto de Corte Semanal

Neste experimento, foram calculadas as médias de interações semanais dos alunos com a plataforma Moodle para construir um *dataset* de treinamento. Assim, a cada semana um modelo preditivo seria gerado, e, com isso, pode-se avaliar o desempenho de um modelo preditivo logo nas primeiras semanas de curso. Foram gerados modelos de predição para cada uma das 42 semanas de curso, o que compreende os módulos 1 e 2. Salienta-se que, no momento deste experimento, não havia a disponibilidade de todos os dados, e, por isso, apenas os módulos 1 e 2 foram utilizados.

O *dataset* construído pode ser visualizado na Tabela 12. Como pode ser observado, existem dois alunos com suas respectivas frequências de acesso, e cada uma das *features* representa uma semana (*e.g.*, W1 para semana 1). Ao final é atribuído um valor binário representando o estado deste aluno, 1 para evadiu e 0 para regular. O objetivo é acrescentar novas *features* na medida em que o algoritmo é treinado.

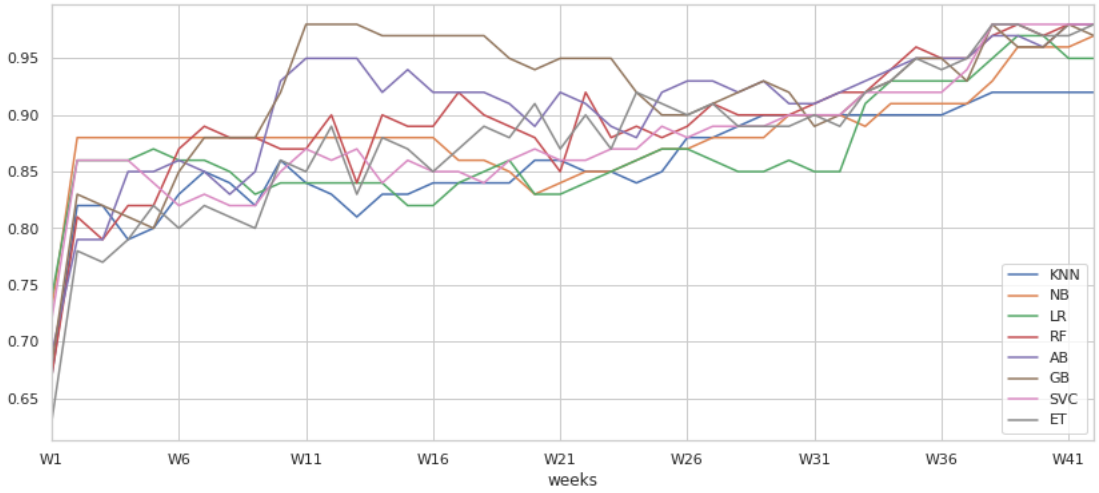
Tabela 12: Dataset Após o Pré-processamento

id	**W1**	**W2**	**W3**	**...**	**evadiu**
33	0.40625	0.370	0.857	...	0
34	0.370	0.181	0.01	...	1

Os resultados obtidos para cada modelo preditivo são ilustrados na Figura 17. Nota-se que a partir da primeira semana de curso a acurácia dos modelos gira em torno de 80% a 85%. Os algoritmos *Gradient Boosting* e *AdaBoost* tiveram seu desempenho elevado a partir da semana 10 e, após a semana 36, todos os algoritmos já apresentavam uma acurácia superior a 90%. Na semana 44, todos os algoritmos já apresentavam acurácia de

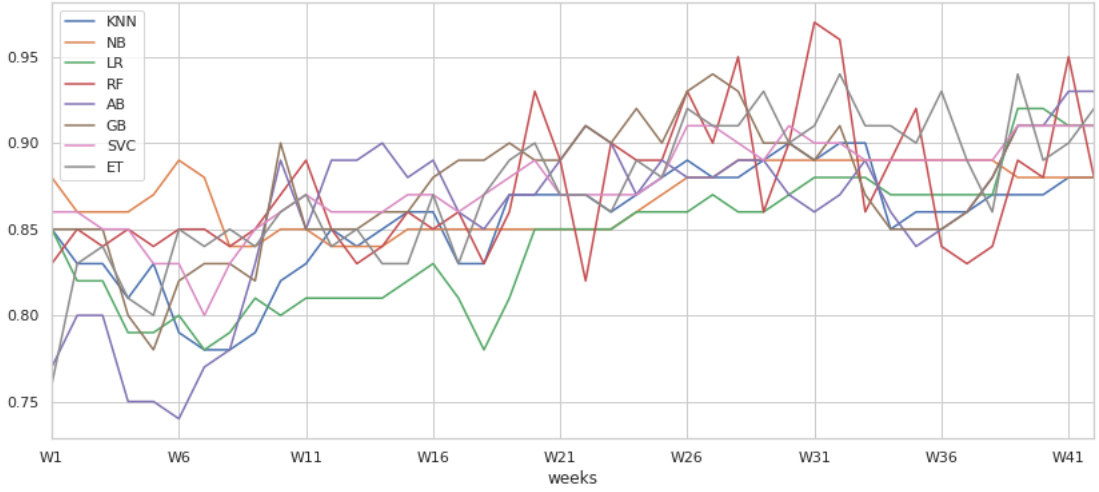
95% ou superior a este valor, com exceção do algoritmo KNN.

Figura 17: Acurácia dos Algoritmos de Aprendizado de Máquina



Para avaliar a capacidade dos modelos em distinguirem entre as duas classes (*i.e.*, evadido e não evadido), foi calculada a AUROC para cada modelo. Na Figura 18 pode-se visualizar os valores gerados por esta métrica para cada semana de curso. Nas primeiras semanas os valores variaram bastante entre 75% e 85%, tendo leves elevações ao longo das semanas. Em alguns casos, como para o *Random Forest*, por exemplo, houve constantes variações. Essas variações são justificadas devido ao grande número de *thresholds*, que são os pontos que formam as curvas, tanto na Figura 18, quanto na Figura 17. Existem 44 *thresholds*, um para cada modelo (*i.e.*, para cada semana).

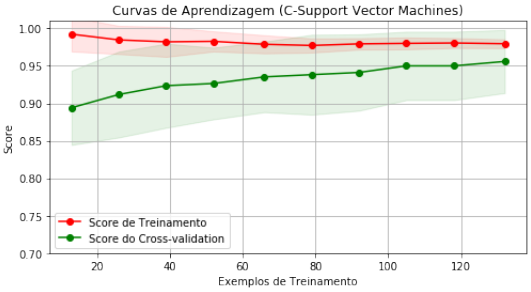
Figura 18: AUROC dos Algoritmos de Aprendizado de Máquina



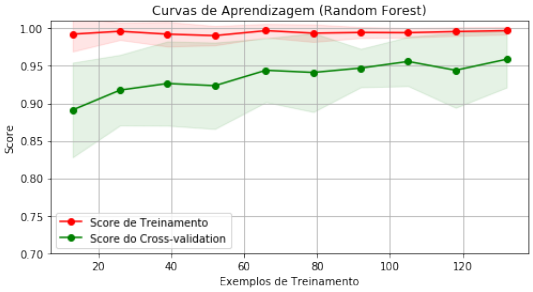
Como pode ser observado na Figura 18, o desempenho final dos algoritmos em relação a AUROC ficou em torno dos 90%. As curvas ROC não foram plotadas para este experimento devido ao grande número de modelos, totalizando mais de 300 gráficos. No entanto, as curvas de aprendizagem dos algoritmos foram plotadas em um gráfico, para

que seja possível avaliar o desempenho de cada um dos modelos, verificar se o estimador sofre mais com erro de variância ou *bias* e se foram sobreajustados (*overfitted*). As curvas de aprendizagem dos algoritmos *C-Support Vector Machines* e *Random Forest* podem ser visualizadas na Figura 19.

Figura 19: Curvas de Aprendizagem: *C-Support Vector Machines* e *Random Forest*



[1]

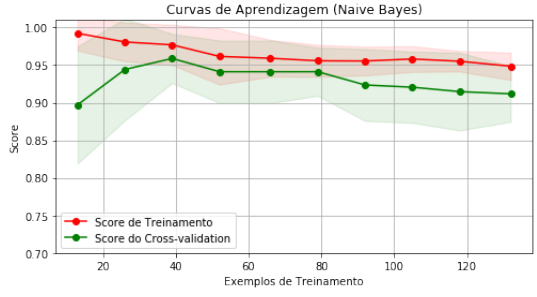


[2]

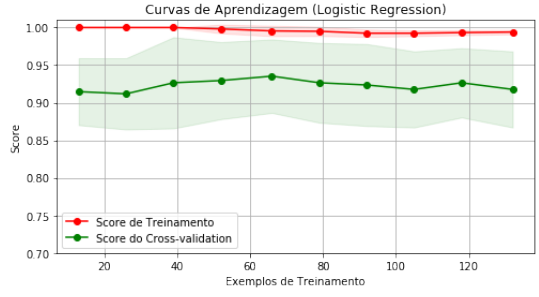
Na Figura 19 são ilustradas duas Subfiguras, sendo que a de número 1 refere-se as curvas de aprendizagem do algoritmo SVC, e a 2 está relacionada ao RF. Analisando o algoritmo SVC, pode-se observar que as curvas de treinamento e de validação estão convergindo na medida em que a amostragem de treinamento aumenta. A distância entre as duas curvas mostra que existe uma certa variância, pois as curvas não convergiram, e também, o *bias* não está alto. Este modelo parece bastante promissor, pois não parece estar sobreajustado, e adicionar mais amostras de treinamento provavelmente aumentará a generalização. Em relação ao desempenho do algoritmo RF ilustrado na Subfigura 2, pode-se observar que a curva de treinamento está quase no seu *score* máximo, e isso pode indicar *overfitting*. Também, a distância entre as curvas é ainda maior que a encontrada no SVC.

O desempenho do algoritmo LR mostrou-se um pouco semelhante ao do RF, como pode-se ver na Subfigura 2 da Figura 20. Aqui, pode-se verificar um *overfitting* na curva de treinamento, e a curva de validação não parece aumentar com o aumento de dados. Neste caso, seria melhor aumentar a complexidade do modelo, adicionando mais *features* e reajustando os hiperparâmetros do algoritmo.

Figura 20: Curvas de Aprendizagem: *Naive Bayes* e *Logistic Regression*



[1]

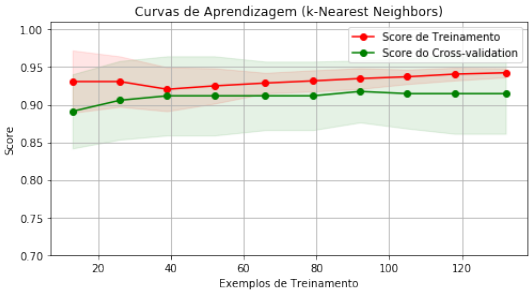


[2]

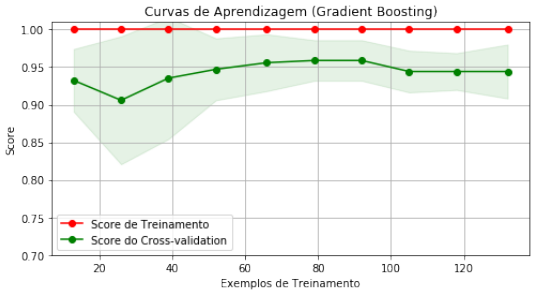
Quanto ao algoritmo NB, o qual tem suas curvas de aprendizagem ilustradas na Subfigura 1 da Figura 20, observa-se que há uma grande aproximação das curvas em alguns pontos, mas a partir de um certo momento, em que a amostragem aumenta, as curvas se distanciam um pouco, assim como seu *score*. A seguir, apresenta-se as curvas de aprendizagem do KNN e Gradient Boosting na Figura 21.

Figura 21: Curvas de Aprendizagem: *k-Nearest Neighbors* e *Gradient Boosting*

[1]



[2]

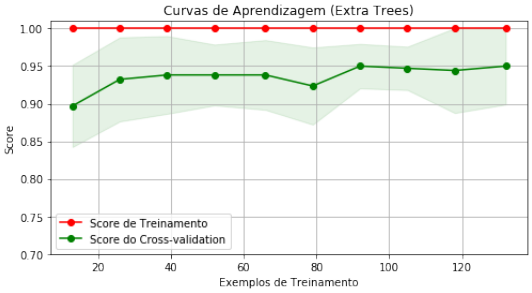


Na Subfigura 1 da Figura 21, observa-se que o algoritmo KNN apresenta ter um bom desempenho e um *score* médio entre 89% e 95%. Tanto o *bias* quanto a variância parecem estar numa medida apropriada. Na Subfigura 2 da Figura 21 é possível visualizar o desempenho do algoritmo GB, que tem sua curva de treinamento no *score* máximo (*overfitting*), e apresenta uma grande variância. Aqui, o modelo não é muito beneficiado de mais dados de treinamento, pois a curva de validação não parece estar subindo.

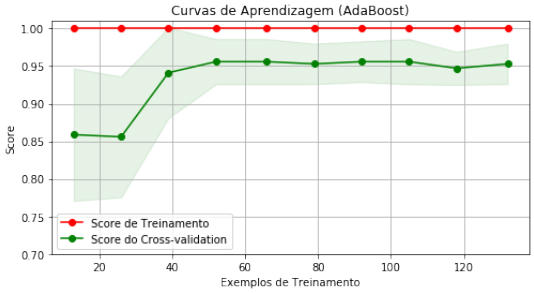
Por fim, na Figura 22 pode-se visualizar as curvas de aprendizagem dos algoritmos ET (Subfigura 1) e *AdaBoost* (Subfigura 2). Ambos os modelos apresentam *overfitting* em seu treinamento e possuem suas curvas de validação planas, sem uma expectativa de que possam aumentar.

Figura 22: Curvas de Aprendizagem: *Extra Trees* e *AdaBoost*

[1]



[2]



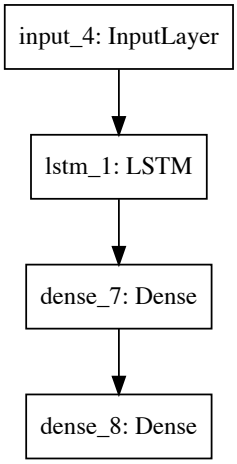
Neste caso, também, seria necessário aumentar a complexidade do modelo com mais *features* descritivas, assim como ajudas os hiperparâmetros do algoritmo. O Acréscimo de mais exemplos de treinamento parece ser benéfico para todos os modelos. No entanto, em alguns casos, seria necessário fazer mais reajustes nos modelos. Os algoritmos que

aparentam ter um melhor desempenho são o SVC, NB e KNN. Os demais modelos parecem apresentar *overfitting*, e seria necessário fazer grandes reajustes para melhorar suas performances.

4.3 Experimento C - *Deep Learning*

Neste experimento, o objetivo é avaliar o uso de *deep learning* na tarefa de predição de alunos em risco de evasão, utilizando o *dataset* disponível no Moodle para mapear o comportamento dos estudantes ao longo do curso. Assim, foi implementada uma *recurrent neural network*, com a arquitetura de uma LSTM. Para isso, foi utilizado o Keras e Tensorflow. Na Figura 23, pode-se visualizar o gráfico gerado gerado pela rede com o modelo da rede.

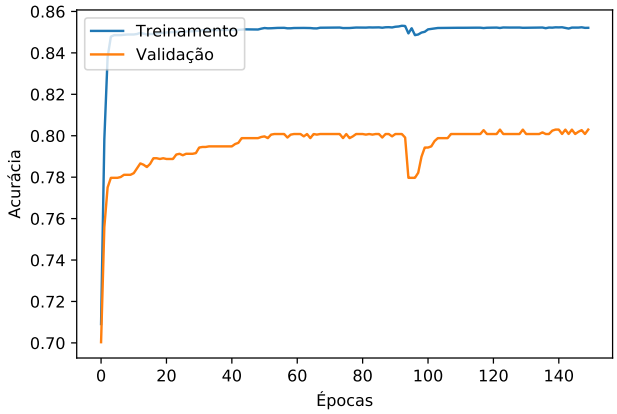
Figura 23: Modelo da RNN LSTM



Como pode-se observar, camadas totalmente conectadas foram definidas usando a classe Dense, que recebe o número de neurônios e uma função de ativação que vai definir a saída do neurônio. Na primeira camada, foram definidas 10 células de memória LSTM. Para a segunda camada, definiu-se o número de neurônios igual a 10, e usa ReLU (*Rectified Linear Unit*) como função de ativação. Por fim, tem-se a saída da rede, que usa a função de ativação Sigmoid.

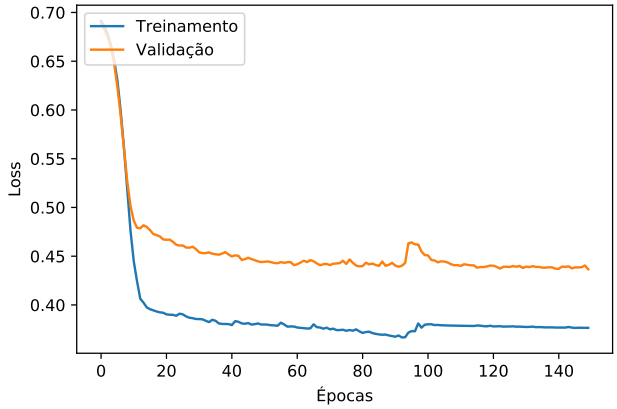
A acurácia da LSTM pode ser observada na Figura 24, em que pode-se visualizar duas curvas, uma mostrando o desempenho de treinamento, e outra mostrando o desempenho de validação. Observa-se que ambas as curvas mantêm-se planas durante todo processo de treinamento, com leves alterações em sua pontuação. A média de acertos calculada é de 85% para o conjunto de treinamento, e 79% para validação. Houve uma diferença de 6% entre as duas médias, o que na Figura 24 parece maior.

Figura 24: *Recurrent Neural Network* - LSTM: Acurácia



Na Figura 25, tem-se um gráfico similar. No entanto, a *loss* tende a diminuir na medida em que o modelo é treinado. Observando a Figura 25, pode-se notar que ambas as curvas estão indo na mesma direção, mas, também, apresentam um *gap* entre elas. A média calculada pela *loss* foi de 0.39 para o *dataset* de treinamento, e 0.45 para o de validação, deixando um *gap* similar ao visualizado na Figura 24.

Figura 25: *Recurrent Neural Network* - LSTM: *Loss*



Com a observação dos gráficos ilustrados nas Figuras 24 e 25, conclui-se que, apesar de não apresentar uma acurácia muito alta quando comparada com os algoritmos da Abordagem 1, a LSTM apresentou um bom desempenho. Os gráficos indicam que o modelo não sofreu *overfitting*, pois a acurácia não está tão alta, e a curva de *loss* sobre os dados de validação está indo na direção da curva de treinamento. No entanto, seria necessário que o *dataset* tivesse mais dados para caracterizar as classes positiva e negativa. Ao que parece, a rede não conseguiu identificar muitas características que representassem os perfis dos alunos.

5 CONSIDERAÇÕES FINAIS

O objetivo principal dessa dissertação foi utilizar os registros educacionais do Moodle para predizer o risco eminente de evasão de alunos da Educação a Distância. Para a execução desta meta, utilizaram-se métodos computacionais convencionais, como os algoritmos *Logistic Regression* e *Random Forest*, para integrarem a Abordagem 1 da pesquisa, junto com outros seis algoritmos presentes na Seção 3.2.2. Ademais, tem-se a Abordagem 2, em que implementou-se uma rede *Long Short Term Memory*, que tem a capacidade de lidar com grandes dependências de dados, o que seria essencial para utilizar sobre o *dataset* construído com rastros digitais do Moodle.

O *dataset* utilizado nessa pesquisa para a execução de ambas abordagens foi construído com dados educacionais extraídos da plataforma Moodle. Esses dados caracterizam todo o histórico de interações dos alunos com seus respectivos cursos EaD: Aplicações para Web e Tecnologias da Informação e Comunicação na Educação.

Dois experimentos foram implementados pela Abordagem 1: Experimento A e Experimento B. No primeiro experimento, observou-se que o algoritmo *Extra Trees* apresentou um melhor desempenho na tarefa de classificação perante os demais algoritmos. Esta conclusão deu-se por meio das métricas acurácia, AUROC, *precision*, *recall* e F1 *score* que, para o algoritmo *Extra Trees*, apresentaram os *scores* mais altos. Este experimento foi publicado na conferência *Frontiers in Education*, e encontra-se no anexo A - . Para o experimento B, em que foram especificados pontos de corte semanais, os resultados revelam uma acurácia alta logo nas primeiras semanas de curso. Para verificar se os modelos apresentavam *overfitting*, foram plotadas suas curvas de aprendizagem. A análise dessas curvas indicam que houve *overfitting* nos algoritmos *Random Forest*, *Logistic Regression*, *Gradient Boosting*, *Extra Trees* e *AdaBoost*.

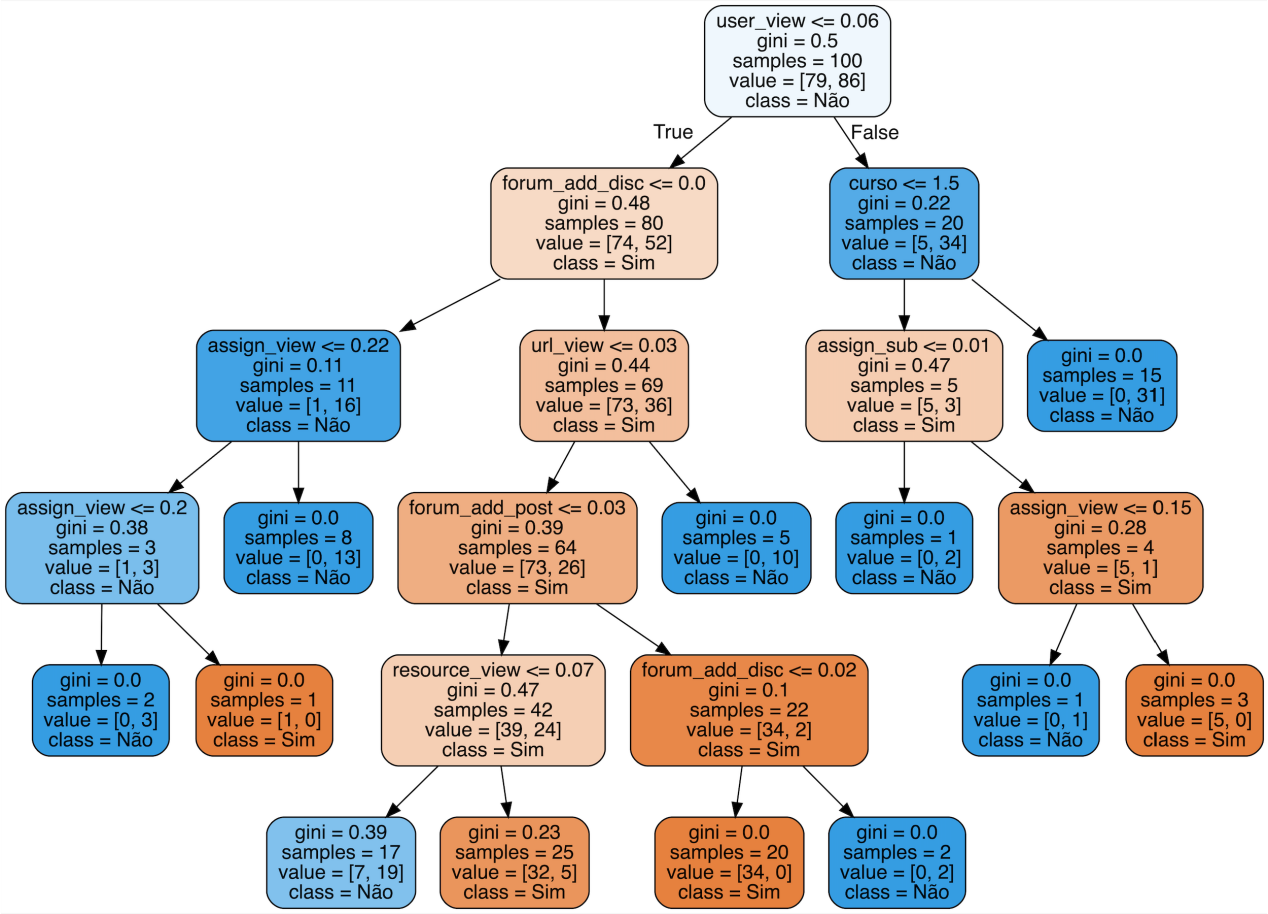
O terceiro experimento (Experimento C), em que utilizou-se *deep learning*, apresentou bons resultados. No entanto, o modelo gerado pela LSTM está longe de ser generalizado. Essa conclusão deu-se ao analisar as curvas de aprendizagem implementadas para esse experimento, que utilizam duas métricas de avaliação: acurácia e *loss*. O *gap* que existe entre as curvas de treinamento e validação para ambas as métricas indica que o modelo necessita de mais dados para caracterizar melhor as duas classes do *dataset* (evadiu:

1 - sim, 0 - não). No entanto, o desempenho dessa arquitetura de rede parece promissor para lidar com o problema abordado nessa pesquisa.

Os algoritmos *C-Support Vector Machines*, *Naive Bayes* e *k-Nearest Neighbors*, que foram utilizados na Abordagem 1, também parecem apresentar um desempenho promissor. Os valores de variância e *bias* não são altos para estes três algoritmos, e eles não parecem apresentar *overfitting*.

Uma das grandes limitações apresentada pelo uso de aprendizado de máquina e que foram reconhecidas nessa pesquisa, é a falta de transparência por trás de suas decisões (DU; LIU; HU, 2019). Como pode-se interpretar as decisões tomadas pelos algoritmos utilizados nessa pesquisa? É de grande importância poder explicar as decisões tomadas por um algoritmo quando aplicado na sociedade. Existem pesquisas que discutem sobre o uso de técnicas computacionais para interpretar predições realizadas por modelos como RNN (DU; LIU; HU, 2019). Nessa dissertação, foi utilizado um algoritmo que pode ser facilmente interpretado, o *Random Forest*. O *Random Forest* possui um conjunto de árvores de decisões, e é possível selecionar uma dessas árvores para poder interpreta-la. Assim, com a ajuda de bibliotecas como o Graphviz, foi gerado com gráfico ilustrando as decisões tomadas por uma das árvores do *Random Forest* (Figura 26).

Figura 26: Uma Árvore de Decisão do Algoritmo *Random Forest*



A árvore de decisão basicamente representa um fluxograma de árvore binária, em que cada nó divide um grupo de observações de acordo com alguma *feature*. O objetivo da árvore é dividir os dados em grupos, de modo que todos os elementos de um grupo pertençam à mesma categoria. O nó é composto por: uma condição; o *gini*, que mede a probabilidade de uma observação ser classificada incorretamente quando escolhido aleatoriamente; *samples* que refere-se ao número total de observações presentes naquele nó; *value* que indica o número de observações por classe; e *class* informa a classe que aquele nó representa (Sim - evadiu e Não - não evadiu). Estabeleceu-se uma profundidade de tamanho cinco para a árvore, visto que ela ficaria muito grande e não seria possível visualizar seus nós.

No primeiro nó, pode-se observar que a *feature user_view* foi utilizada como condição para dividir os dados daquele nó. Se o valor de *user_view* for menor ou igual a 0.06, as observações passam para o nó seguinte, que possui como condição *forum_add_disc* menor ou igual a 0.0. Nota-se que os nós azuis representam a classe Não, e os nós laranjas referem-se a classe Sim. Quanto mais forte for a cor do nó, mais observações daquela classe estão presentes nele. Este é um problema discutido na atualidade, como visto em (DU; LIU; HU, 2019), que apresenta direções futuras e contribuições para demais pesquisas.

REFERÊNCIAS

BULLINARIA, J. A. Recurrent neural networks. **Neural Computation: Lecture**, [S.l.], v.12, 2013.

BURGOS, C.; CAMPANARIO, M. L.; PENA, D. de la; LARA, J. A.; LIZCANO, D.; MARTÍNEZ, M. A. Data mining for modeling students' performance: a tutoring action plan to prevent academic dropout. **Computers & Electrical Engineering**, [S.l.], v.66, p.541–556, 2018.

CAMBRUZZI, W. L.; RIGO, S. J.; BARBOSA, J. L. Dropout Prediction and Reduction in Distance Education Courses with the Learning Analytics Multitrail Approach. **J. UCS**, [S.l.], v.21, n.1, p.23–47, 2015.

CENSO, E. Relatório analítico da aprendizagem a distância no Brasil 2017 [livro eletrônico]/[organização] ABED–Associação Brasileira de Educação a Distância. **Curitiba: Editora InterSaberes**, [S.l.], 2018.

CORRIGAN, O.; SMEATON, A. F. A course agnostic approach to predicting student success from VLE log data using recurrent neural networks. In: EUROPEAN CONFERENCE ON TECHNOLOGY ENHANCED LEARNING, 2017. **Anais...** [S.l.: s.n.], 2017. p.545–548.

DEWAN, M. A. A.; LIN, F.; WEN, D. et al. Predicting dropout-prone students in e-learning education system. In: IEEE 12TH INTL CONF ON UBIQUITOUS INTELLIGENCE AND COMPUTING AND 2015 IEEE 12TH INTL CONF ON AUTONOMIC AND TRUSTED COMPUTING AND 2015 IEEE 15TH INTL CONF ON SCALABLE COMPUTING AND COMMUNICATIONS AND ITS ASSOCIATED WORKSHOPS (UIC-ATC-SCALCOM), 2015., 2015. **Anais...** [S.l.: s.n.], 2015. p.1735–1740.

DU, M.; LIU, N.; HU, X. Techniques for interpretable machine learning. **Communications of the ACM**, [S.l.], v.63, n.1, p.68–77, 2019.

EAD, A. C. BR: relatório analítico da aprendizagem a distância no brasil 2016= censo ead. **BR: Curitiba: InterSaberes**, [S.l.], 2017.

FAWCETT, T. An introduction to ROC analysis. **Pattern recognition letters**, [S.l.], v.27, n.8, p.861–874, 2006.

FRIEDMAN, J. H. On bias, variance, 0/1—loss, and the curse-of-dimensionality. **Data mining and knowledge discovery**, [S.l.], v.1, n.1, p.55–77, 1997.

GALAFASSI, C.; GALAFASSI, F. F. P.; VICARI, R. M. Predictive Teaching and Learning. In: EPIA CONFERENCE ON ARTIFICIAL INTELLIGENCE, 2017. **Anais...** [S.l.: s.n.], 2017. p.549–560.

GEURTS, P.; ERNST, D.; WEHENKEL, L. Extremely randomized trees. **Machine learning**, [S.l.], v.63, n.1, p.3–42, 2006.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>.

GOUGH, E. The Study of Distance Education. In: KEEGAN, D. (Ed.). **Foundations of Distance Education**. 11 New Fetter Lane, London EC4P 4EE: Routledge, 1996. p.3–19.

GRAVES, A. Long Short-Term Memory. In: **Supervised Sequence Labelling with Recurrent Neural Networks**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p.37–45.

GREFF, K.; SRIVASTAVA, R. K.; KOUTNÍK, J.; STEUNEBRINK, B. R.; SCHMIDHUBER, J. LSTM: a search space odyssey. **IEEE transactions on neural networks and learning systems**, [S.l.], v.28, n.10, p.2222–2232, 2016.

HOCHREITER, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. **International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems**, [S.l.], v.6, n.02, p.107–116, 1998.

JAYAPRAKASH, S. M.; MOODY, E. W.; LAURÍA, E. J.; REGAN, J. R.; BARON, J. D. Early alert of academically at-risk students: an open source analytics initiative. **Journal of Learning Analytics**, [S.l.], v.1, n.1, p.6–47, 2014.

JORDAN, M. I.; MITCHELL, T. M. Machine learning: trends, perspectives, and prospects. **Science**, [S.l.], v.349, n.6245, p.255–260, 2015.

KANG, K.; WANG, S. Analyze and Predict Student Dropout from Online Programs. In: INTERNATIONAL CONFERENCE ON COMPUTE AND DATA ANALYSIS, 2., 2018. **Proceedings...** [S.l.: s.n.], 2018. p.6–12.

KELLEHER, J. D.; MAC NAMEE, B.; D'ARCY, A. **Fundamentals of machine learning for predictive data analytics**: algorithms, worked examples, and case studies. [S.l.]: MIT Press, 2015.

KOSTOPOULOS, G.; KOTSIANTIS, S.; PIERRAKEAS, C.; KOUTSONIKOS, G.; GRAVVANIS, G. A. Forecasting students' success in an open university. **International Journal of Learning Technology**, [S.l.], v.13, n.1, p.26–43, 2018.

KOSTOPOULOS, G.; KOTSIANTIS, S.; PINTELAS, P. Estimating student dropout in distance higher education using semi-supervised techniques. In: PANHELLENIC CONFERENCE ON INFORMATICS, 19., 2015. **Proceedings...** [S.l.: s.n.], 2015. p.38–43.

KOSTOPOULOS, G.; KOTSIANTIS, S.; PINTELAS, P. Predicting student performance in distance higher education using semi-supervised techniques. In: **Model and data engineering**. [S.l.]: Springer, 2015. p.259–270.

KOSTOPOULOS, G.; KOTSIANTIS, S.; RAGOS, O.; GRAPSA, T. N. Early dropout prediction in distance higher education using active learning. In: INTERNATIONAL CONFERENCE ON INFORMATION, INTELLIGENCE, SYSTEMS & APPLICATIONS (IISA), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p.1–6.

KOSTOPOULOS, G.; LIPITAKIS, A.-D.; KOTSIANTIS, S.; GRAVVANIS, G. Predicting Student Performance in Distance Higher Education Using Active Learning. In: INTERNATIONAL CONFERENCE ON ENGINEERING APPLICATIONS OF NEURAL NETWORKS, 2017. **Anais...** [S.l.: s.n.], 2017. p.75–86.

KOTSIANTIS, S. Educational data mining: a case study for predicting dropout-prone students. **International Journal of Knowledge Engineering and Soft Data Paradigms**, [S.l.], v.1, n.2, p.101–111, 2009.

KOTSIANTIS, S. B.; PIERRAKEAS, C.; PINTELAS, P. E. Preventing student dropout in distance learning using machine learning techniques. In: INTERNATIONAL CONFERENCE ON KNOWLEDGE-BASED AND INTELLIGENT INFORMATION AND ENGINEERING SYSTEMS, 2003. **Anais...** [S.l.: s.n.], 2003. p.267–274.

KOTSIANTIS, S. B.; PINTELAS, P. E. A decision support prototype tool for predicting student performance in an ODL environment. **Interactive Technology and Smart Education**, [S.l.], v.1, n.4, p.253–264, 2004.

LAGUARDIA, J.; PORTELA, M. Evasão na educação a distância. **ETD-Educação Temática Digital**, [S.l.], v.11, n.1, p.349–379, 2009.

LUKOšEVIčIUS, M.; JAEGER, H. Reservoir computing approaches to recurrent neural network training. **Computer Science Review**, [S.l.], v.3, n.3, p.127 – 149, 2009.

MALHOTRA, P.; VIG, L.; SHROFF, G.; AGARWAL, P. Long short term memory networks for anomaly detection in time series. In: 2015. **Proceedings...** [S.l.: s.n.], 2015. p.89.

MORAN, J. M. Educação inovadora presencial e a distância. **São Paulo, SP: CA**, [S.l.], 2003.

MORENO-MARCOS, P. M.; ALARIO-HOYOS, C.; MUÑOZ-MERINO, P. J.; KLOOS, C. D. Prediction in MOOCs: a review and future research directions. **IEEE Transactions on Learning Technologies**, [S.l.], 2018.

OKUBO, F.; YAMASHITA, T.; SHIMADA, A.; OGATA, H. A neural network approach for students' performance prediction. In: LAK, 2017. **Anais...** [S.l.: s.n.], 2017. p.598–599.

OLSON, D. L.; DELEN, D. **Advanced data mining techniques**. [S.l.]: Springer Science & Business Media, 2008.

PASCANU, R.; MIKOLOV, T.; BENGIO, Y. Understanding the exploding gradient problem. **CoRR, abs/1211.5063**, [S.l.], v.2, 2012.

PEÑA, D. de la; LARA, J. A.; LIZCANO, D.; MARTÍNEZ, M. A.; BURGOS, C.; CAMPANARIO, M. L. Mining activity grades to model students' performance. In: INTERNATIONAL CONFERENCE ON ENGINEERING & MIS (ICEMIS), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p.1–6.

PONTI, M. A.; COSTA, G. B. P. da. Como funciona o deep learning. **arXiv preprint arXiv:1806.07908**, [S.l.], 2018.

PRIM, A. L.; FÁVERO, J. D. Motivos da evasão escolar nos cursos de ensino superior de uma faculdade na cidade de Blumenau. **Revista E-Tech: Tecnologias para Competitividade Industrial-ISSN-1983-1838**, [S.l.], p.53–72, 2013.

PRODANOV, C. C.; FREITAS, E. C. de. **Metodologia do trabalho científico**: métodos e técnicas da pesquisa e do trabalho acadêmico-2ª edição. [S.l.]: Editora Feevale, 2013.

PROVOST, F. J.; FAWCETT, T.; KOHAVI, R. et al. The case against accuracy estimation for comparing induction algorithms. In: ICML, 1998. **Anais...** [S.l.: s.n.], 1998. v.98, p.445–453.

QUEIROGA, E.; CECHINEL, C.; ARAÚJO, R. Predição de estudantes com risco de evasão em cursos técnicos a distância. In: BRAZILIAN SYMPOSIUM ON COMPUTERS IN EDUCATION (SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO-SBIE), 2017. **Anais...** [S.l.: s.n.], 2017. v.28, n.1, p.1547.

RABELO, H.; BURLAMAQUI, A.; VALENTIM, R.; SOUZA RABELO, D. S. de; MEDEIROS, S. Utilização de técnicas de mineração de dados educacionais para predição de desempenho de alunos de EaD em ambientes virtuais de aprendizagem. In: BRAZILIAN SYMPOSIUM ON COMPUTERS IN EDUCATION (SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO-SBIE), 2017. **Anais...** [S.l.: s.n.], 2017. v.28, n.1, p.1527.

RAMOS, J. L. C.; GOMES, A. S.; RODRIGUES, R.; SILVA, J.; SOUZA, F. d. F. de; GOUVEIA ZAMBOM, E. de; PRADO, L. Um Modelo Preditivo da Evasão dos Alunos na EAD a Partir dos Construtos da Teoria da Distância Transacional. In: BRAZILIAN SYMPOSIUM ON COMPUTERS IN EDUCATION (SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO-SBIE), 2017. **Anais...** [S.l.: s.n.], 2017. v.28, n.1, p.1227.

RAMOS, J. L. C.; SILVA, J.; PRADO, L.; GOMES, A.; RODRIGUES, R. Um estudo comparativo de classificadores na previsão da evasão de alunos em EAD. In: BRAZILIAN SYMPOSIUM ON COMPUTERS IN EDUCATION (SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO-SBIE), 2018. **Anais...** [S.l.: s.n.], 2018. v.29, n.1, p.1463.

RIDGEWAY, G. The state of boosting. **Computing Science and Statistics**, [S.l.], p.172–181, 1999.

SAK, H.; SENIOR, A.; BEAUFAYS, F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In: FIFTEENTH ANNUAL CONFERENCE OF THE INTERNATIONAL SPEECH COMMUNICATION ASSOCIATION, 2014. **Anais...** [S.l.: s.n.], 2014.

SANTANA, M. A.; BARROS COSTA, E. de; SANTOS NETO, B. F. dos; SILVA, I. C. L.; REGO, J. B. A predictive model for identifying students with dropout profiles in online courses. In: EDM (WORKSHOPS), 2015. **Anais...** [S.l.: s.n.], 2015.

SCIKIT-LEARN. **Validation curves**: plotting scores to evaluate models. Data de Acesso: 12 de julho de 2019.

SHALEV-SHWARTZ, S.; BEN-DAVID, S. **Understanding machine learning**: from theory to algorithms. [S.l.]: Cambridge university press, 2014.

SILVA, E. L. d.; MENEZES, E. M. Metodologia da pesquisa e elaboração de dissertação. , [S.l.], 2001.

SILVA, F.; SILVA, J. da; SILVA, R.; FONSECA, L. C. Um modelo preditivo para diagnóstico de evasão baseado nas interações de alunos em fóruns de discussão. In: BRAZILIAN SYMPOSIUM ON COMPUTERS IN EDUCATION (SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO-SBIE), 2015. **Anais...** [S.l.: s.n.], 2015. v.26, n.1, p.1187.

SOARES, F. B. O Custo aluno UAB no Ensino Superior a distância na UFRGS: estudo de caso referente ao curso de graduação tecnológica em planejamento e gestão para o desenvolvimento rural. , [S.l.], 2015.

SORJAMAA, A.; HAO, J.; REYHANI, N.; JI, Y.; LENDASSE, A. Methodology for long-term prediction of time series. **Neurocomputing**, [S.l.], v.70, n.16-18, p.2861–2869, 2007.

XENOS, M. Prediction and assessment of student behaviour in open and distance education in computers using Bayesian networks. **Computers & Education**, [S.l.], v.43, n.4, p.345–359, 2004.

APÊNDICE A - ARTIGO PUBLICADO NO *FRONTIERS IN EDUCATION* (FIE) 2019

Understanding the Student Dropout in Distance Learning

Myke Morais de Oliveira, Regina Barwaldt,
Marcelo Rita Pias, Danúbia Bueno Espíndola
Center for Computational Sciences (C3)
Federal University of Rio Grande (FURG)
Rio Grande - RS, Brazil
mykemoliveira@gmail.com, reginabarwaldt@furg.br,
pias.marcelo@gmail.com, danubiaespindola@furg.br

***Abstract*—This Research to Practice Full Paper introduces an approach for the early identification of students at risk of dropping out of their distance learning courses. Students dropout in university courses have long been a major issue leading to social, academic and financial impacts. Predictive modelling analysis has been used as a tool to identify at an early stage probable cases of dropout. In distance learning, this type of analysis is made possible mostly because of the increasing adoption rate of Virtual Learning Environments (VLEs) where data on the student academic activities can be continuously recorded. This paper discusses the design and validation of a Learning Analytics system for early identification of students at risk of dropping out. The case study presented relies on data collected from two postgraduate courses as part of Brazils Open University. The methodology for the system design and validation comprises the steps to build an intelligent data pipeline. Jupyter Notebooks have been d used as the data science analysis environment in order to create data pipelines and have their performance evaluated. Results obtained from the validation of models built out of eight machine learning techniques show an average accuracy of 84% among the set of ML techniques tested. The highest accuracy is delivered by the Extra Trees classifier with 88%. Logistic Regression performed the worst performance with an accuracy of 79%.**

***Index Terms*—Predictive Modelling, Dropout, Distance Education, Machine Learning**

I. INTRODUCTION

Distance learning means that students engage with learning materials at home or workplace. Such an interaction usually takes place through a virtual learning environment - VLE (e.g., Moodle, Canvas, Blackboard) where course activities and student progression are recorded in a log-based database. Such data creates a form of student digital learning footprint that can be captured throughout the academic course duration.

In the last few years, distance learning has experienced significantly high dropout rates worldwide whenever students decide to quit a course of study, often giving no formal justification and just not turning up in the VLE system (e.g., logging-in) [1]. The burden of a student dropout to the relevant parties cannot be neglected. Socio-economical impacts on the university, educators, and students arise in this context [2]. Students can suffer physiological illness thus requiring professional advice.

To shed some light on the dropout problem space, studies in the research field of Learning Analytics have discussed interesting yet early results from the validation of predictive models tailored to the dropout behaviour identification of students at risk [3] [4]. Datasets on the student personal demographic profile and academic performance are crucial enablers to these new analytics approaches.

This work addresses the following research question:

- Is it possible to build an accurate yet interpretable machine learning model that identifies students at risk of dropping out of their distance academic courses?

This paper introduces the design and validation of predictive models that can be further integrated into an early identification system capable of flagging students at risk of leaving their academic courses (drop-out). Early identification is a design issue which has the potential to provide a useful time window for intervention schemes. The system is centred on the analysis of the student digital footprint to unveil data patterns and traits of early dropout signs (e.g., lack of login in the VLE over the last week). The case study presented relies on the student data generated out of two blended learning courses offered at a public university in Brazil. The methodology for the system design and validation comprises the required steps to build an intelligent data pipeline.

II. BACKGROUND

Technology-mediated learning has significantly changed the way lecturers teach, and students learn. The Internet brings the possibility of transforming face-to-face lectures into distance learning (EaD) [4]. In Brazil, EaD courses exhibited dropout rates in the range of 11% to 25%. The leading causes for this include financial issues, lack of time and non-adaptation to the EaD modality, as discussed in Brazil Census on Distance Learning [5].

Most universities that offer EaD courses employ an online platform where students and instructors can interact, access and even manage the content. In this context, Virtual Learning Environments (VLE), web-based systems, allow instructors and students to carry out some actions, such as content sharing, tasks submissions and some form of online communication [6]. Such collaborative environments record large amounts

of information such as the actions and interactions students perform during the academic course [7].

Within the Moodle VLE, for example, a feature called log records activity reports for each action taken in the VLE [8]. It is possible to access data on the volume and duration of online sessions, what resources the student accessed, the type of teaching material that has been visualised, and active (or not) participation in the forum threads discussion to cite a few [7].

Such a large volume of data is characterised by the properties that define big data, which represents not only massive data, but it also describes the capacity for data mining and manipulation to obtain insights and patterns [9]. An insight characterises the discovery of new information that was previously not entirely visible. Researchers can use such insights the processing of generating predictive models to anticipate future scenarios.

Such a large dataset can be useful for the analysis and mining of relevant information. Learning Analytics (LA) is presented as a contributor to the educational setting with studies that exploit big data [10]. Recently, universities have begun employing big data analytics in an attempt to address pressing issues related to student retention [10]. Learning analytics core objective is to identify hidden patterns in the educational data and, based on these standards, to gain a better understanding of the educational context, assessing the students learning, and generating predictions on their academic performance [3]. There is a wide range of issues in which learning analytics play a role such as improving fundraising programs, helping teachers to have a better understanding of students skills and competencies, and then optimising student performance. The implementation of interventions is also one of the issues the learning analytics addresses, and this is an essential and timely activity [4]. When a student receives a notification (e.g., an email) with a suggestion that their performance in the classroom needs improvement, this intervention creates a scenario in which the student is more likely to engage with the instructor to overcome any learning difficulty. These interactions promote better student engagement which is often associated with better retention rates [3].

III. METHODOLOGY

Jupyter Notebook has been used as the platform to carry out the methodological procedures. A notebook is an open source web platform that allows the user to build and share documents (notebooks) with graphical representations, narrative texts, lines of code, mathematical formulas, among other resources. Such features make an efficient platform for the development of projects. The Python data science libraries (pandas, numpy, matplotlib, scikit-learn, and others) were used to implement the data processing pipelines, or simply called the data pipelines. Figure 1 shows the data pipeline created for this work.

The stages of the pipelie include the generation of datasets, data preprocessing, data modelling and validation. The last two stages can iterate a couple of times so that the target model performance is satisfactory.

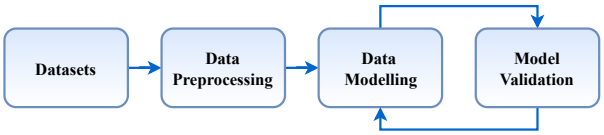


Fig. 1. Data Pipeline

A. Datasets

To develop the present study, two postgraduate programs of the Brazilian Open University (UAB) were used as points of data collection. Moodle has been selected as the VLE platform thus offering access to event logs that track the user activities throughout a distance learning course. This raw data is available both in batch and real-time mode.

The Moodle raw data have been used in batch-mode as the basis for the generation of the dataset with features. The dataset comprises 200,166 records split into 115,407 for Course 1 and 84,762 in Course 2. A total of 166 students were enrolled in both courses.

The structure of the logs is shown in Table I.

TABLE I
DATA LOGS

Attributes	Description
Course	Course name (e.g., Course 1).
Time	Date and time at which the action occurred.
IP Address	The IP address of the computer that performed the action.
Full Name	The full name of the user who performed the action.
Action	The action performed by the user (e.g., Task Submission).
Information	Description of action taken (e.g., Task 1 - Final Report).

The datafields Course and Full Name have been anonymised for privacy protection reasons. For instance, the course a specific student has interacted is identified as course 1 and course 2. The Action datafield categorises the action performed, while the Information datafield carries details about this specific action. Table I presents an example of these two datafields.

To determine which students can be classified in the dropout and not-dropout classes, further investigation was carried out by sifting through the Moodle access logs. Access logs record the timestamp of the last visit the students made to the VLE. The rule applied considers a drop-out in those cases where the student has not accessed the Moodle VLE for more than 30 days. The local university administration considers the 30 days a reasonable cut-off point given the postgraduate courses are part-time with the majority of the student interactions taking place inside the Moodle VLE. Further specialisation of the drop-out rule has been applied to rule out some exceptional cases. When a student enrolled in a module did not access within 30 days the next module of the programme, this case has also been considered a drop-out. It should be pointed out that the postgraduate programme of the UAB program

comprises four modules with an average duration time of three months each. As a result, logs referring to module 1 have been only collected because module 2 was in progress still. Given this scenario, it was possible to classify with more certainty which students were dropout cases.

B. Data Preprocessing

This step organises and establishes which data should be used as the input to further computational techniques and in what format they ought to be represented (e.g., binary values). To achieve this a data processing pipeline has been defined with the following operations: dataset cleaning and structuring, anonymisation, and data transformation (wrangler operations). Cleaning the data has been a significant labour work. Duplicates, inconsistencies in feature names (e.g., space between words), null data, and the removal of dispensable (not relevant) features have to be treated appropriately.

Structuring the dataset establishes the final data structure with the generation of features. Data representation is another issue that was addressed. During the first tests, online student interactions with the course materials were used as recorded values in the generated dataset. No cut-off point was used for the calculation of the frequencies in this first step of the work. The Action datafield was exceptionally valuable in this study since it provided the means to understand the types of student interactions. Also, this datafield made it possible to compute the frequency of interaction occurrences (e.g., access) through the Time datafield. Given this context, the Action datafield has been restructured as new features that characterise each of the actions. This concept is explored in Section III-A, where each datafield corresponds to an action (e.g., Task submission), and the table instances convey the classification of that action (i.e., 0 for not taking any action and 1 for yes, the action was performed). In this way, the dataset comprises features, which are the actions, and for each instance (student) a value of 0 or 1 has been assigned to assist the classification of actions. This process is also called binarisation of categorical variable.

TABLE II
DATA TRANSFORMATION

id	action_1	action _2	action _3	...	action _n
33	0	1	0	...	0
34	1	0	0	...	0

The anonymisation was used to protect the fundamental rights of freedom and privacy of the users, as well as of the courses in question. In this way, any information that denounced the identity of users and the course has been hidden.

C. Data Modelling

When the data processing finishes, the next step in the data pipeline is the predictive model generation through supervised learning. The machine learning task in-hand is classification where a model learns a mapping function f (x) from x to y; whereas x is the input data (feature list) and y is the classification label. For instance, if the classification task is to output whether a given image represents a dog or cat the created model learns from a few examples which can be a set of images from dogs and cats. The model adjusts its internal parameters so that the predicted label is the actual label in the examples (training dataset). This process of learning from labelled/annotated examples of dog/cat images is named supervised learning. In this case, the model (also called classifier) is interested in two classes (dogs and cats). In this work, a similar binary classification task is used where the target variable DROPOUT is yes or no.

The data pipeline as structured in Fig 1 allowed for experimentation and validation of a number of machine learning techniques including: k-Nearest Neighbors (KNN), Gaussian Naive Bayes (NB), C-Support Vector Classification (SVC), Logistic Regression (LR), Random Forest (RF), Adaptive Boosting (AdaBoost), Gradient Boosting and Extremely Randomized Trees (Extra Trees).

k-Nearest Neighbors: according to [11], the central concept that idealises k-NN is to assume that similar things must be equal. For a better understanding, when the algorithm processes the input of a new instance, the similarity of this new instance with its closest neighbours in the training base is verified. In this way, the classification is given based on the similarity of the nearest neighbour.

Gaussian Naive Bayes: the NB is a probabilistic classifier based on the Bayes Theorem. According to [11], the NB algorithm is a classic demonstration of how generative assumptions and parameter estimates simplify the learning process. In this study, Gaussian NB has been used.

Support Vector Classification: SVC is subclass of the Support Vector Machines (SVM) methods addresses the complex challenges of the sample looking for big margin separators [11]. That is, the training data is separated through a separation line (hyperplane), which seeks to maximise the distance between the closest points and distinct classes. The significant margin is the distance between the hyperplane and the first point of each class.

Logistic Regression: the LR algorithm belongs to the family of linear methods that predict the outcome of the target variable based on a single or more predictor variable [3]. The LR has a built-in logistic function that captures the linear combination of the predictor variables. This model the occurrence probability of a particular value of the target variable.

Random Forest: RF is a classifier composed by a collection of decision trees, in which each of those trees is constructed out of an algorithm A applied to a training set S and in a random vector, where is randomly sampled from some distribution. The random forest prediction is obtained through a majority vote on the predictions of individual trees. The authors [11] pointed out that RF helps to prevent overfitting because it explores a set of decision trees and not just a single one.

AdaBoost: this algorithm belongs to the Boosting algorithm family, which uses a generalisation of linear predictors

to overcome two main issues, bias-variance tradeoff and computational complexity of the learning process [11]. The AdaBoost produces a hypothesis defined by a linear combination of simple hypotheses. In other words, it counts on the family of classes of hypotheses obtained by the composition of a linear predictor based on the simple classes.

Gradient Boosting: according to [12], this algorithm originated from the connection between Optimization and Boosting. Gradient Boosting determines the direction, the text, in each interaction performed by the algorithm. In this way, the algorithm needs to improve the data fitting and select a certain model from the allowed class of functions that are in more agreement with the direction [12].

Extra Trees: this algorithm was introduced by [13], which reports that the algorithm constructs a set of decision trees, or regression, that it is exempt based on the standard procedure from top to bottom. It is two main differences, compared to other tree-based clustering methods are that Extra Trees divides nodes by choosing very random cut points, and by using the entire learning sample to grow trees.

These algorithms were chosen based on the analysis of previous research work that has shown acceptable performance [14] [15] [1] [16] [17] [18]. However, the performance of a classic machine learning model significantly depends on the hand designed features for the available datasets. Alternative modelling approaches such as representational data learning (deep learning) [19] offer new possibilities for improved performance at the expense of transparency and process visibility. At this stage of this work, it is important to reach a compromise between the performance and visibility of decisions made to reach predictions. In this case, this work focuses on techniques of the so-called traditional machine learning so that the model is not treated as a **black box** (as in the deep learning). A new research field named Explainable AI (XAI) [20] has been investigating methods to make deep neural networks more visible and interpretable.

D. Model Validation

For the models' validation, the following metrics have been used: Accuracy, Area Under the ROC curve (AUROC), Precision, Recall, and F1 score. The use of these evaluative metrics guarantees greater reliability on the efficiency of the models to deal with the real data problem (i.e., never seen by the classifier). This approach also ensures greater security so that the models do not suffer overfitting.

These metrics make use of four measures that are generated by the classification algorithms. These measures are the True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). TP and TN represent the correct classifications rate in the positive class. TP as the positives (DROPOUT = yes), and TN as the negatives (DROPOUT = no). The FP and FN represent the incorrect classifications rate (e.g., when a positive is classified as a negative, and vice versa). The FP represents the negatives that have been labeled as positives, and the FN are incorrectly classified positives.

Accuracy: this is undoubtedly the most widely used perfor mance metric for classification [21]. This is estimated by dividing the total of positives and negatives sorted correctly, by the total of observations in the test set [22], that is, both the correct and the incorrect classifications (the accuracy is given by the formula present in (1)). The test set refers to a small portion of the database that is intended to evaluate the algorithm performance. The test set refers to a small portion of the dataset that is destinated to evaluate the performance of the algorithm, which in this task is represented by 10% of the entire dataset. In this work, the validation is done by the Stratified 10-fold cross-validation, which consists of dividing the dataset into ten folds, using all of them for training and test (e.g., 1 for test and the other ones for training and so on).

$$\frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

AUROC: the Receiver Operating Characteristic (ROC) presents a coordinate system for the visualisation of the model performance. In this metric space, the TP rate (TPR) is plotted on the Y-axis, and FP rate (FPR) is represented on the X-axis. A point in the ROC space (FPR, TPR) corresponds to a model classifier performance. The resulting curve illustrates the balanced error for a given model where the captured predictive behaviour is not tied to any class distribution or error costs. The Area Under the ROC curve (AUROC) computes the overall relationship between the TP rate, also known as Sensitivity or Recall, and the FP rate, also known as Fall-out. The latter measure is also computed as 1-specificity.

Precision: the correct prediction rate within the expected positive class. Precision is calculated through the formula in (2) [11].

$$\frac{TP}{TP + FP} \tag{2}$$

Recall: also known as sensitivity, this metric represents the TP rate (TPR) as predicted by the classification model [11]. The recall is givin by the formula (3).

$$\frac{TP}{TP + FN} \tag{3}$$

F1 score: also known as F-score or F-measure, evaluates the accuracy obtained through the test set. The F1 score represents the harmonic mean between precision and recall, and the formula gives it in (4) [11].

$$2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

The maximum F1-score value of 1 is obtained when precision and recall equal to 1, and its minimum value of 0 is attained whenever one of the two metrics is 0.

IV. Modeling Results

Table III gives an overview of the data records that have been extracted from the Moodle VLE platform. This table refers to the total number of records considered for the enrolled

students. Records have been labelled to reflect the outcome of either a dropout or not-dropout which is the target variable defined. The number of records considers the entire digital footprint of the student when developing the course activities proposed in the VLE.

TABLE III
COLLECTED RECORDS

	Course 1	Course 2	Total
Records	115,407 (57.65%)	84762 (42.34%)	200,166
Students	90 (54.21%)	76 (45.78%)	166
Dropout students	20 (39.21%)	31 (60.78%)	51
Not dropout students	70 (60.86%)	45 (39.13%)	115

Data records related to the actions that tutors performed during the course (e.g., setup of assessments, upload of videos) have been removed to keep consistency (a total of 91,658 removed records). The actions in the final dataset related to the student VLE interactions were then transformed into a set of data features. A sample of the final dataset generated out of the preprocessing step is shown in Table IV. A single row represents the summarization of a given student actions.

Student ID information was anonymised through a two-layer randomisation procedure, and unimportant datafields for the modelling step were eliminated (IP Address and Information). Data inconsistencies such as feature names have been corrected.

TABLE IV
FINAL CHARACTERIZATION OF THE DATASET

id	course_view	forum_view	resource_view	...	dropout
0	0.393056	0.151389	0.001389	...	0
1	0.334171	0.120101	0.041709	...	0
3	0.363395	0.151194	0.010610	...	0

Fifteen features out of forty-three in the dataset were selected as the final feature data. To make this selection, the covariance was computed as an interdependence measure among features. Covariance close to zero has been used as the threshold value for the feature removal. In some cases, features that exhibited an extremely low occurrence or too many null data have been removed. The selected group of features (V) include: course_view, forum_view, forum_view_discussion, resource_view, forum_add_post, forum_add_disc, assign_view, assign_sub, user_view, url_view, page_view, forum_search, polo (location), course. Dropout is the chosen target variable.

Once the data was pre-processed, the first experiments using the k-NN, NB, SVC, LR, RF, AdaBoost, Gradient Boosting and Extra Trees algorithms were carried out. The accuracy of the generated models is presented in decreasing order of accuracy in Figure 2. All the algorithms used in the model training achieved a high accuracy with an average of 84%. The best accuracy of 88% is delivered by the Extra Trees classifier. LR performed the worst with an accuracy of 79%.

The AUROC metric has been used to quantify the classifier ability to distinguish between the two classes of interest.

TABLE V
DATASET FINAL FEATURES

Features	Description
course_view	Course visualization.
forum_view	Forum visualization.
forum_view_discussion	Discussion visualization on the forum.
resource_view	Visualization of a resource.
forum_add_post	Adding a forum post.
forum_add_disc	Adding a forum discussion.
assign_view	Activity visualization.
assign_sub	Activity submission.
user_view	User visualization.
url_view	Visualizing a resource address.
page_view	Visualizing a page.
forum_search	Forum search.

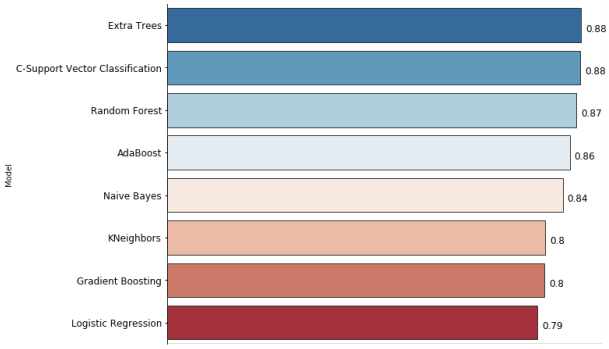


Fig. 2. Accuracy of the trained models

Figure 3 presents the FP rate (FPR) on the x-axis, also called specificity, which is a performance measure of the negative class. The TP rate (TPR) represented on the y-axis measures the performance of the positive class. TPR is also calculated as 1-specificity.

In this sense, a model that exhibits optimal performance has a ROC curve attached at the upper end of the left corner of the graph, that is, when the FP rate is 0% and TP rate is 100%.

A value of AUROC close to 100% of degree separation means the models can distinguish well between the two classes of interest (1: DROPOUT YES and 0: DROPOUT NO). In contrast, AUROC value close to zero represents a model that inverts the classes, e.g., classify DROPOUT YES as NO and the other way round. When the AUROC is 50%, the model cannot separate between classes (might as well toss a coin). Half of the models shown in Figure 3 exhibit an AUROC greater than 90%; the RF based model exceeded this level to reach an AUROC of 94%. The lowest separation degree yield in the LR model which also obtained the lowest accuracy among the models validated.

Precision, Recall and F1 score were measured using the scikit-learn library. The results obtained for these three metrics can be visualised in Table VI. The results observed for each model have small variation among these metrics. The values are calculated as the average for the two clases (1 and 0 respectively).

The algorithm Extra Trees achieved the highest recall average ith value of 0.94. The minimum average of 0.79

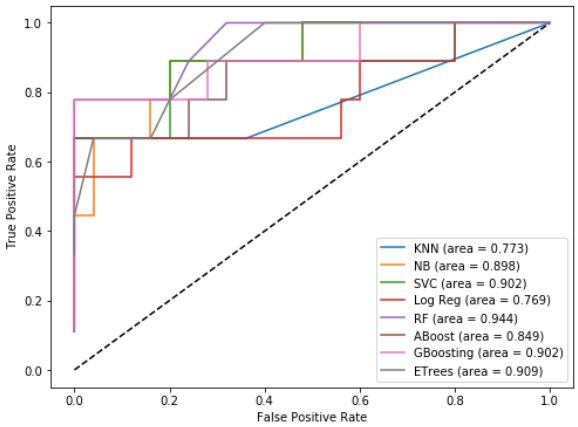


Fig. 3. ROC curves

was obtained through the algorithm AdaBoost. The KNN, NB, SVC and mainly LR algorithms suffered large variations between values of 0 and 1. In such cases, class 1 (positive) exhibited shallow recall values, compared to class 0 (negative). This suggests that such algorithms were not efficient for correctly classifying the dropout students.

TABLE VI
RECALL, PRECISION AND F1 SCORE

Algorithms	Recall	Precision	F1 score
k-Nearest Neighbors	0.91	0.92	0.91
Gaussian Naive Bayes	0.85	0.85	0.85
C-Support Vector Classification	0.91	0.92	0.91
Logistic Regression	0.88	0.90	0.87
Random Forest	0.91	0.91	0.91
AdaBoost	0.79	0.80	0.80
Gradient Boosting	0.91	0.91	0.91
Extra Trees	0.94	0.95	0.94

As for precision, not much variation has been found between the values of classes 0 and 1. In some classifiers, the positive class performed better than the negative class, and vice versa. Most of the algorithms obtained reasonable precision in the correct classification of both classes. An exception is the AdaBoost classifier, which achieved an average value of 0.60 in the positive class, which indicates a high FP number.

The F1 score combines precision with a recall to generate a mean, to result in a single value number that indicates the overall quality of the model. As it can be seen in VI, exist an imbalance between F1 scores in classes 0 and 1, where class 1 has the smallest percentages. The best results have been achieved through the algorithm Extra Trees.

This analysis suggests that the model based on the Extra Trees algorithm outperforms the other models used in the validation. Extra Trees exhibited a satisfactory accuracy and proved to be efficient in classifying the classes correctly. The essential performance measures are (a) Recall of 1.0 for the negative class (0: Dropout NO) and 0.78 for the positive class (1: Dropout YES); (b) Resulting mean of 0.94 that can be seen in Table VI. The mean of 0.94 reveals that considering both classes; this is the overall performance for the Extra Trees model. The Precision of the Extra Trees has an average of 0.95 which means that of all the positive examples, 0.95 of them are positive. Based on the analysis of the results, it was concluded that the Extra Trees algorithm obtained the best performance in the classification task compared to the other algorithms used. Extra Trees presented the best accuracy and proved to be efficient to classify the classes correctly. He presented a Recall of 1.0 for the negative class (0) and 0.78 for the positive class (1), resulting in a mean of 0.94 that can be seen in IV. The mean of 0.94 reveals that considering both classes; this is the overall performance for the Extra Trees model. The Precision of the Extra Trees has an average of 0.95 which means that of all the positive examples, 0.95 of them are positive.

Figure 4 presents the learning curve for the Extra Trees model during the training procedure. The graph comprises two curves, that is, training (red) and Cross Validation (green), where the score is located on the y-axis, and the number of training instances is on the x-axis.

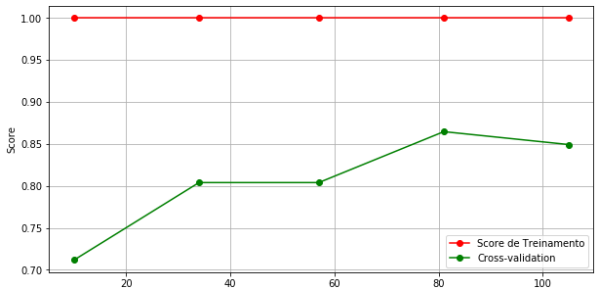


Fig. 4. Extra Trees learning curves

Based on the interpretation of the graph, it can be suggested that the training score is located at its maximum end (equivalent to 1). This indicates over-fitting from the provided training set; The Cross Validation curve, although it gives the idea of convergence, it is far from achieving an optimal result. The considerable gap between cross-validation score and training score indicates a scenario of high variance. This can be treated by gathering more data or reducing the complexity of the model (e.g., reducing the number of features).

V. FINAL CONSIDERATIONS

This paper introduces the design and validation of predictive models to detect students at risk of dropping out. Such an early identification system has the potential to provide a useful time window for student interventions. A possible intervention is to bring risk awareness straight to the students through a smartphone app. This paper presented performance results for eight machine learning models trained to classify students as a case of either dropout or non-dropout. Students enrolled in the postgraduate EaD (distance learning courses) extensively used the Moodle Moodle platform as their means of interaction with course content and communication between students and teachers.

As future work, the dataset will be augmented in an attempt to reduce the variance observed in the Extra Trees learning

curves (4). Besides, the selection of hyperparameters and features will be used to refine the models. The unbalance characteristic of the dataset might have influenced the results as well, since the negative class accounts for 69.27% of the instances in the dataset. Reaching a balanced dataset through up or downsampling is considered the next logical step. Once all these optimisation and model interpretation tasks are accomplished, a direction to explore is to enhance the performance which makes deep neural networks a suitable approach [19].

ACKNOWLEDGMENT

The authors thank the Coordination for the Improvement of Higher Education Personnel (CAPES), the Federal University of Rio Grande (FURG), and the Post-Graduate Program in Computation (PPGComp).

REFERENCES

[1] J. L. C. Ramos, A. S. Gomes, R. Rodrigues, J. Silva, F. d. F. de Souza, E. de Gouveia Zambom, and L. Prado, "Um modelo preditivo da evasão dos alunos na ead a partir dos construtos da teoria da distância transacional," in *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, vol. 28, p. 1227, 2017.

[2] R. L. L. Silva Filho, P. R. Motejunas, O. Hipólito, and M. B. C. M. Lobo, "A evasão no ensino superior brasileiro.," *Cadernos de pesquisa*, vol. 37, no. 132, pp. 641–659, 2007.

[3] S. M. Jayaprakash, E. W. Moody, E. J. Lauría, J. R. Regan, and J. D. Baron, "Early alert of academically at-risk students: An open source analytics initiative," *Journal of Learning Analytics*, vol. 1, no. 1, pp. 6–47, 2014.

[4] R. Conijn, C. Snijders, A. Kleingeld, and U. Matzat, "Predicting student performance from lms data: A comparison of 17 blended courses using moodle," 2017.

[5] A. C. EAD, "Br: relatório analítico da aprendizagem a distância no brasil 2016= censo ead," *BR: Curitiba: InterSaberes*, 2017.

[6] S. Lonn and S. D. Teasley, "Saving time or innovating practice: Investigating perceptions and uses of learning management systems," *Computers & Education*, vol. 53, no. 3, pp. 686–694, 2009.

[7] L. P. Macfadyen and S. Dawson, "Mining lms data to develop an early warning system for educators: A proof of concept," *Computers & education*, vol. 54, no. 2, pp. 588–599, 2010.

[8] Á. F. Agudo-Peregrina, S. Iglesias-Pradas, M. Á. Conde-González, and Á. Hernández-García, "Can we predict success from log data in vles? classification of interactions for learning analytics and their relation with performance in vle-supported f2f and online learning," *Computers in human behavior*, vol. 31, pp. 542–550, 2014.

[9] G. Rieder and J. Simon, "Big data: A new empiricism and its epistemic and socio-political consequences," in *Berechenbarkeit der Welt?*, pp. 85–105, Springer, 2017.

[10] A. S. Alblawi and A. A. Alhamed, "Big data and learning analytics in higher education: Demystifying variety, acquisition, storage, nlp and analytics," in *2017 IEEE Conference on Big Data and Analytics (ICBDA)*, pp. 124–129, IEEE, 2017.

[11] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[12] G. Ridgeway, "The state of boosting," *Computing Science and Statistics*, pp. 172–181, 1999.

[13] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine learning*, vol. 63, no. 1, pp. 3–42, 2006.

[14] S.-P. M. O. K. T. R. H. S. M. N. Petkovic, D. and A. Vigil, "Using the random forest classifier to assess and predict student learning of software engineering teamwork," in *2016 IEEE Frontiers in Education Conference (FIE)*, pp. 1–6, IEEE, 2016.

[15] G. Kostopoulos, A.-D. Lipitakis, S. Kotsiantis, and G. Gravvanis, "Predicting student performance in distance higher education using active learning," in *International Conference on Engineering Applications of Neural Networks*, pp. 75–86, Springer, 2017.

[16] T. R. Hagedoorn and G. Spanakis, "Massive open online courses temporal profiling for dropout prediction," in *Tools with Artificial Intelligence (ICTAI), 2017 IEEE 29th International Conference on*, pp. 231–238, IEEE, 2017.

[17] D. de la Peña, J. A. Lara, D. Lizcano, M. A. Martínez, C. Burgos, and M. L. Campanario, "Mining activity grades to model students' performance," in *2017 International Conference on Engineering & MIS (ICEMIS)*, pp. 1–6, IEEE, 2017.

[18] V. Hegde and P. Prageeth, "Higher education student dropout prediction and analysis through educational data mining," in *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, pp. 694–699, IEEE, 2018.

[19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[20] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[21] F. J. Provost, T. Fawcett, R. Kohavi, *et al.*, "The case against accuracy estimation for comparing induction algorithms.," in *ICML*, vol. 98, pp. 445–453, 1998.

[22] D. L. Olson and D. Delen, *Advanced data mining techniques*. Springer Science & Business Media, 2008.