Predição de Risco de Evasão de Alunos Usando Métodos de Aprendizado de Máquina em Cursos Técnicos

Priscilla Busin de Bitencourt, Carlos Andres Ferrero

[1]Curso de Graduação em Ciência da Computação
Instituto Federal de Santa Catarina (IFSC), Lages, SC - Brasil

priscila.busin.bitencourt@gmail.com, andres.ferrero@ifsc.edu.br

Abstract. *Every year the number of vacancies in technical courses offered by public educational institutions increases. However, the number of student dropouts has alerted educators and researchers, not just for its social impact but also for economic losses. This work proposes the induction of decision tree based classification models to predict potential dropout students in technical courses. For this, we used only basic information about students and their attendances during the first week. Results show that classification models were able to identify 25% of potential student dropouts with 86% of precision. The models were considered promising to help educational managers to better control student dropout rates.*

Resumo. *Anualmente presenciamos o aumento de vagas em cursos técnicos em instituições públicas. No entanto o número de evasões de alunos vem alertando educadores e pesquisadores, pois é responsável não apenas por perdas de ordem social mas também de ordem econômica. Neste trabalho é proposto a construção de modelos de classificação baseados em árvores de decisão para predição de potenciais alunos evasores em cursos de nível técnico. Para isso foram utilizadas informações básicas dos alunos e as suas frequências nos primeiros sete dias de aula. Os modelos construídos conseguiram recuperar 25% dos potências evasores com uma precisão de 86%. Os modelos foram considerados promissores para o gerenciamento de ações de permanência e êxito.*

1. Introdução

Anualmente o sistema educacional vem ofertando um número crescente de vagas em cursos técnicos em instituições públicas. Isto se deve ao crescente número de oferta das industrias, que buscam trabalhadores com qualificação focada nas suas necessidades. Apesar de todos os seus atrativos, um número que vem alertando educadores é o de alunos que não chegam a completar seu curso. No Instituto Federal de Santa Cataria (IFSC) a evasão dos cursos com matrícula em 2017 foi, de 17.5%, e 25% dos cursos tiveram evasão maior do que 27.5%. Este problema se configura como um dos principais no ensino em nosso país, que não abrange somente ordem social mas também financeira. Segundo dados da plataforma Nilo Peçanha no ano de 2017 o gasto corrente por matrícula na instituição foi de R$ 15.267, representando um custo sem retorno ao governo, à sociedade e às instituições.

Estimar o risco de evasão de aluno é um tarefa fundamental para Instituições de Ensino, pois possibilita quantificar a evasão futura, e assim desenvolver melhores estratégias de permanência e êxito. No entanto, a tarefa preditiva de estimar o risco da evasão

DOI: 10.5753/cbie.wcbie.2019.149

não é trivial, pois depende da relação de múltiplos fatores, e as estatísticas simples não permitem estimar o risco à evasão com precisão. A Mineração de Dados é uma área de estudo que tem como objetivo extrair padrões e conhecimento relevante a partir de dados. Uma das formas de extrair esse padrões é utilizado técnicas de Aprendizado de Máquina, uma subárea da Inteligência Artificial, que tem como principal objetivo desenvolver técnicas computacionais capazes de adquirir conhecimento automaticamente (Han et al., 2011). Esses padrões podem ser úteis para construir modelos preditivos para auxiliar em processos de tomada de decisão associada à evasão de alunos.

A Mineração de Dados aplicada a área de educação vem se desenvolvendo cada vez mais. No Brasil, Brandão et al. (2003) foram os pioneiros nesta área e propuseram analisar os dados do Programa Nacional de Informática na Educação. Estudos como este na área de mineração de dados educacionais crescem constantemente. A maior parte dos trabalhos nessa área para predição de evasão tem sido com foco em ambientes virtuais de aprendizagem, onde existem informações constantes sobre atividade do aluno (Araújo and Rodrigues, 2018; Ramos et al., 2018). Cursos presenciais de nível superior também estão entre os que tem recebido maior atenção da comunidade acadêmica. Alguns trabalhos tem utilizado informações sobre o desempenho do aluno antes de ingressar no ensino superior como: o histórico escolar do ensino médio, a nota no Exame Nacional do Ensino Médio (ENEM) ou a nota no vestibular (Brito et al., 2015). Outros trabalhos, como o de Digiampietri et al. (2016), realizam a predição de evasão a partir do segundo semestre letivo usando dados do desempenho acadêmico nas disciplinas do primeiro ano do curso. No entanto, poucos trabalhos tem sido realizados no contexto de *cursos técnicos presenciais*, como o de Maria et al. (2016). No contexto desses cursos geralmente não se tem informações prévias do desempenho dos alunos e nem constantes informações sobre a contínua atividade do mesmo.

Neste trabalho propõe-se a construção de modelos preditivos para identificação de estudantes em risco de *evasão em cursos técnicos presenciais*, a partir da análise de fatores sociais e acadêmicos dos alunos e a integração desses modelos em uma ferramenta computacional para estudos prospectivos.

O restante do trabalho está organizado da seguinte maneira: na Seção 2 são descritos os materiais e métodos, na Seção 3 são apresentados e discutidos os resultados do trabalho e, na Seção 4 são apresentados a conclusão e os trabalhos futuros.

2. Material e Método

O método de desenvolvimento deste trabalho foi baseado no processo de Descoberta de Conhecimento em Bases de Dados (*Knowledge Discovery in Data Bases* – KDD) (Fayyad et al., 1996), o qual consiste nas etapas de: (1) seleção de dados, (2) pré-processamento, (3) mineração de dados e (4) pós-processamento. Essas etapas são descritas a seguir.

Etapa 1 – Seleção de Dados

Nesta etapa foi realizada a coleta e seleção dos dados, que serão utilizados nas próximas etapas. O conjunto de dados cedido para estudo pelo IFSC contém informações de alunos de cursos técnicos do IFSC que realizaram matrícula entre 01/2018 à 07/2019. Todos esses alunos apresentam status de ativo, cancelado ou trancado. O conjunto de dados está constituído de 772 instâncias de matrícula de alunos que incluem as informações

de: identificador id[1], data de nascimento, sexo, bairro, cidade, etnia, ano de ingresso, período de ingresso, nome do curso, turno do curso, status. Adicionalmente também foram cedidos dados de frequência dos alunos nos primeiros 7 dias corridos a partir de sua primeira aula, sendo eles a carga horária e o número de faltas. Na Tabela 1 é apresentada a descrição dos dados, sendo que para cada atributo é apresentado o tipo de dado e possíveis valores. Conforme Tabela 1 o conjunto de dados inicial está constituído de 13 atributos,

Tabela 1. Descrição dos Atributos do Conjunto de Dados.

#	**Atributo**	**Tipo de Dado**	**Possíveis valores**
01	id	Numérico	Valores inteiros sequenciais de 1 a 772.
02	$data\_nascimento$	Data	Valores no formato DD/MM/AAAA.
03	$genero$	Categórico	2 valores: Masculino e Feminino.
04	$bairro$	Categórico	117 valores: Centro, Universitário, entre outros.
05	$cidade$	Categórico	24 valores: Lages, Correia Pinto, entre outros.
06	$etnia$	Categórico	5 valores: branco, pardo, oriental, indígena, preta.
07	$ano\_ingresso$	Numérico	Valores 2018 e 2019.
08	$periodo\_ingresso$	Numérico	Valores semestre 1 e 2.
09	$nome\_curso$	Categórico	7 valores: Técnico em Informática, Técnico em Biotecnologia, entre outros.
10	$turno\_curso$	Categórico	3 valores: Matutino, Vespertino e Noturno.
11	$carga\_horaria$	Numérico	Valores inteiros.
12	$numero\_faltas$	Numérico	Valores inteiros.
13	$status$	Categórico	3 valores: Ativo, Cancelado e Trancado.

dos quais cinco atributos são numéricos, sete são categóricos e um possui formato de data. A partir desses dados levantados na próxima etapa será realizado o pré-processamento.

Etapa 2 – Pré-processamento

Nesta etapa foi realizado o pré-processamento dos dados, que inclui as fase de: (a) criação de novos atributos, (b) limpeza e transformação dos dados, (c) construção de conjunto de dados para posterior indução de modelos.

Na primeira fase foi realizada a criação dos atributos idade e percentual de faltas. O novo atributo $idade$ refere-se à idade do aluno quando ingressou no curso, e o valor é obtido utilizando a data de nascimento e uma data de referência. Essa data de referência é a data em que o aluno ingressou no curso. Após isso, foi computado o percentual de faltas nos primeiros sete dias corridos. Esse novo atributo foi necessário devido ao fato de que durante os primeiros sete dias corridos alunos de diferentes cursos podem ter tido diferentes cargas horárias, de acordo com a organização da grade horária de cada curso. Por esse fato, o valor de percentual de faltas em relação à carga horária dos primeiros sete dias corridos representa uma informação na mesma escala para todos os alunos. Sejam γ e ϵ os valores da carga horária e número de faltas, respectivamente, nos primeiros 7 dias corridos a partir da primeira aula, o percentual de faltas é calculado por $f_{pf}(\gamma, \epsilon) = \frac{\epsilon}{\gamma}$.

Na segunda fase, de limpeza e transformação de dados, foi realizada a identificação e exclusão de registros com valores discrepantes. Do total de 772 instâncias

[1]Essa identificação é um número sequencial que não permite revelar a identidade do aluno.

observou-se 8 registros da base de dados apresentaram valores de idade inferiores 13 anos ou superiores a 100 anos. Sendo que esses 8 registros representam apenas 1,04% do total de instâncias, decidiu-se por excluir esses registros do conjunto de dados, restando 764 registros.

Posteriormente foi realizada a transformação do atributo classe, o status. Este atributo inicialmente possui os valores de: *Ativo*, *Cancelado* e *Trancado*. Considerando que o foco deste trabalho é de predizer evasão, neste trabalho consideramos *Evasão* qualquer status diferente *Ativo*. Ainda consideramos o problema de predição de evasão um problema de classificação binária, isto é, com duas classes, as quais comumente são chamadas de classe *positiva* e *negativa*. Neste problema definimos como classe *positiva* a dos alunos que evadiram e a classe *negativa* o restante dos alunos.

Com isso o conjunto de dados ficou constituído por 764 instâncias, representados por 9 atributos: gênero, bairro, cidade, etnia, período de ingresso, nome do curso, turno do curso, percentual de faltas e status (classe). A distribuição das instâncias entre as classes foi de 291 (38.1%) instâncias da classe *positiva* e 473 (61.9%) instâncias da classe *negativa*. Nesse sentido, a probabilidade *apriori* de um aluno evadir é de 38.1%. Assim, para que um modelo preditivo apresente um ganho de conhecimento, este deve apresentar uma acurácia maior do que 61.9% que é a acurácia de um classificador ingênuo que classifica qualquer instância com a classe *negativa*.

Na Fase (c) foram construídos dois conjuntos de dados D_1 e D_2. O conjunto D_1 contém 8 dos 9 atributos (sendo excluído o atributo de percentual de faltas). Esse conjunto de dados tem como objetivo verificar a capacidade dos modelos preditivos em predizer a evasão do aluno com informações antes de iniciar suas atividades acadêmicas no curso técnico. Já o conjunto D_2 que contém todos os 9 atributos. Esse conjunto de dados tem como objetivo avaliar a capacidade dos modelos em predizer a evasão do aluno usando dados de atividade acadêmica da primeira semana de aula. Cada um desses conjuntos foi separado em conjunto de treinamento (60%) e conjunto de teste (40%). Essa separação ocorreu de forma estratificada, o que indica que foi mantida a proporção das classes em cada conjunto.

Etapa 3 – Mineração de Dados

Nesta etapa, foi realizada a definição dos algoritmos para indução de modelos preditivos e da estratégia de ajuste de parâmetros dos modelos. Quanto aos algoritmos foram utilizados *Decision Trees*, *Random Forest* e *XGBoost*, disponíveis nas bibliotecas *Scikit-learn* e *XGBboost* para *Python*. O algoritmo *Decision Trees* é uma implementação do algoritmo C4.5 proposto por Quinlan (2011), *Random Forest* do algoritmo proposto por Breiman em Breiman (2001) e *XGBoost* do algoritmo proposto por Chen and Guestrin (2016).

Cada um desses algoritmos possui parâmetros para construir os modelos e os valores apropriados para esses parâmetros não são os mesmos para qualquer conjunto de dados. O processo de encontrar valores de parâmetros adequados é chamado de *tuning* ou *hiperparametrização*. Uma das estratégias para definição desses parâmetros consiste em testar várias combinações de valores de parâmetros e escolher a melhor. Seguindo essa ideia, a seguir são apresentados os algoritmos, os parâmetros e valores avaliados.

O algoritmo *Decision Trees* consiste na construção de uma estrutura hierárquica de classificação denominada árvore, onde os nós internos (ou de decisão) representam

testes sobre os valores dos atributos e as folhas indicam a classificação. Para definir cada nó de decisão os atributos são avaliados por um critério de qualidade que usa o conjunto de dados e é escolhido o melhor. Os critérios mais comuns são Ganho de Informação e índice Gini. O algoritmo inicia escolhendo o atributo que compõe a raiz da árvore e criando o nó de decisão, subdividindo o conjunto de dados de acordo com os possíveis valores do atributo em *subconjuntos*. O processo de escolha de atributo, criação de nós de decisão e divisão do conjunto de dados é feito recursivamente para cada *subconjunto*, até que cada subconjunto tenha instâncias da mesma classe. Quando isso ocorrer cada subconjunto é transformado em um nó folha para essa classe. Na Tabela 2 são apresentados os parâmetros e valores avaliados para o algoritmo *Decision Trees*.

Tabela 2. Valores dos parâmetros para o algoritmo *Decision Trees*.

Parâmetro	**Valores**	**Descrição**
$criterion$	$2 : (InformationGain, GiniIndex)$	Função que mede a impureza dos dados, usada para avaliar e escolhe o atributo para cada nó da árvore.
max_depth	$5 : (3, 6, ..., 15)$	Profundidade máxima da árvore.
$min\_samples\_leaf$	$6 : (1, 3, 6, ..., 15)$	Número mínimo de instâncias em cada nó folha da árvore.
Número total de configurações: 60		

O algoritmo *Random Forest* proposto por Breiman (2001) é um algoritmo baseado em árvores de decisão que combina os conceitos de escolha aleatória de atributos e exemplos para construir cada árvore de decisão e junta várias árvores de decisão em um único classificador. Nesse algoritmo, para avaliar e escolher cada nó de decisão da árvore são escolhidos aleatoriamente um conjunto de atributos e de exemplos. Essa estratégia permite a construção de diferentes árvores de decisão para classificar, isto é, gerar diferentes regras para dividir o espaço de classificação e definir a classe de um exemplo. Seguindo essa ideia são construídas várias árvores de decisão. Cada novo exemplo é classificado por todas as árvores de decisão e é utilizada uma função que combina o resultado de classificação dessas árvores. Essa função pode ser por votação, escolhendo a classe mais frequente nos resultados de classificação Breiman (2001), ou calculando o valor da média de probabilidade de cada classe nos resultados de classificação, e escolhendo a classe com maior probabilidade média. Na Tabela 3 são apresentados os parâmetros e valores avaliados para o algoritmo *Random Forest*.

O algoritmo *XGBoost* proposto por Chen and Guestrin (2016) também consiste na junção de várias árvores de decisão, mas utiliza uma estratégia denominada *Gradient Boosting* para construir cada árvore. Com essa estratégia cada nova árvore colabora na correção dos erros de classificação produzidos pela árvore treinada anteriormente, o que é feito por meio da atribuição de pesos para aquelas instâncias classificadas incorretamente. Na Tabela 4 são apresentados os parâmetros e valores avaliados para esse algoritmo.

Para encontrar a melhor configuração desses parâmetros para cada modelo foi aplicada a estratégia *Grid Search 10-Fold Cross-Validation*. De acordo com essa estratégia cada configuração de modelo é avaliada utilizando um processo chamado de validação cruzada. Nesse processo o conjunto de dados é divido em n partições, sendo $n - 1$ utili-

Tabela 3. Valores dos parâmetros para o algoritmo *Random Forest*.

Parâmetro	**Valores**	**Descrição**
$n\_estimators$	$10 : (5, 10, 15, ..., 50)$	Número de árvores de decisão.
$criterion$	$2 : (entropia, gini)$	Função que mede a impureza dos dados, usada para escolher o atributo para cada nó da árvore.
max_depth	$5 : (3, 6, ..., 15)$	Profundidade máxima da árvore.
$min\_samples\_leaf$	$6 : (1, 3, 6, ..., 15)$	Número mínimo de exemplos em cada nó folha.
$max\_features$	$2 : (sqrt, log2)$	Número máximo de atributos avaliados para escolher cada nó. Os valores $sqrt$ e $log2$ correspondem à raiz quadrada e ao logaritmo na base 2, respectivamente, do número de atributos.
Número total de configurações: 1200		

Tabela 4. Valores dos parâmetros para o algoritmo *XGBoost*.

Parâmetro	**Valores**	**Descrição**
$n\_estimators$	$10 : (5, 10, ..., 50)$	Número de árvores de decisão.
$colsample\_bytree$	$5 : (0.2, 0.4, ..., 1.0)$	Proporção da amostra de atributos utilizados pelo algoritmo para cada árvore de decisão.
max_depth	$5 : (3, 6, ..., 15)$	Profundidade máxima da árvore.
Número total de configurações: 250		

zadas para o treinamento e uma para o validação. Este processo é repetido n vezes, assim permitindo que cada uma das partes seja utilizada como conjunto de validação. Neste trabalho foram consideradas cinco partições e utilizado o valor da proporção de instâncias classificadas corretamente (acurácia) nos conjuntos de validação para estimar o erro verdadeiro de cada modelo e seus parâmetros. A melhor configuração de parâmetros para cada modelo foi aquela que apresentou maior acurácia.

Neste trabalho optou-se por modelos baseados em árvores de decisão em função da vantagem, em relação a outros modelos, de interpretar as classificações. A partir desses algoritmos é possível obter os atributos mais importantes para a construção das árvores. No caso dos modelos usando *Decision Trees* a classificação de cada nova instância é interpretável, já que é possível extrair a regra de classificação utilizada para predizer a classe. Nos modelos utilizando *RandomForest* e *XGBoost*, embora a interpretação não seja tão simples em função de serem muitas regras utilizadas para classificar uma nova instância (uma regra para cada árvore), ainda é possível extrair dessas regras quais são os atributos que mais influenciaram na classificação.

Etapa 4 – Avaliação de Modelos

A avaliação dos modelos preditivos foi realizada por meio da extração medidas da tabela de confusão e posteriormente pela avaliação das curvas *Precision-Recall* do melhor modelo para ajuste do limiar de decisão.

A tabela de confusão é uma técnica estatística para avaliar as predições realizadas

por um modelo. Para uma instância verdadeiramente negativa um modelo pode acertar, classificando-a como da classe negativa ou errar, classificando-a como da classe positiva. E o mesmo para instâncias verdadeiramente positivas. A tabela de confusão é constituída de quatro possíveis valores:

- TP (*True Positive*): número de instâncias positivas classificadas como positivas;
- FP (*False Positive*): número de instâncias negativas classificadas como positivas;
- TN (*True Negative*): número de instâncias negativas classificadas como negativas;
- FN (*False Negative*): número de instâncias positivas classificadas como negativas.

A partir desses valores é possível calcular as medidas de Acurácia, *Precision* e *Recall*. Acurácia é definida como o número de instâncias classificadas corretamente do total de instâncias classificadas; *Precision* é a proporção de instâncias classificadas como positivas que de fato eram positivas; e *Recall* é a proporção de instâncias da classe positiva que foram classificadas como positivas. Seja N o número total de exemplos classificados pelo modelo, as Equações 1, 2 e 3, apresentam o cálculo das três medidas.

$$Acurácia = \frac{TP + TN}{N} \quad (1) \qquad Precision = \frac{TP}{TP + FP} \quad (2) \qquad Recall = \frac{TP}{TP + FN} \quad (3)$$

Além de classificar instâncias como positivas e negativas, os modelos preditivos conseguem apresentar para cada instância classificada um *score* no intervalo $[0.0, 1.0]$, comumente entendido como a probabilidade de ser da classe positiva. Naturalmente os algoritmos definem o limiar de decisão em 0.5 para classificar a instância como positiva ou negativa. No entanto, explorar a curva *Precision vs. Recall* para diferentes valores de limiar de decisão permite ter uma compreensão mais completa do potencial do modelo de classificação. Nesse sentido, neste trabalho foi selecionado o modelo com melhor qualidade preditiva para analisar a curva *Precision vs. Recall* e ajustar do limiar de decisão.

3. Resultados e Discussão

Na Tabela 5 são apresentados os resultados experimentais dos modelos preditivos que tiveram seus parâmetros ajustados pela técnica *Grid Search 10-Folds Cross-Validation* para o conjunto de dados, D_1. Esses modelos foram avaliados com o conjunto de dados do treino usando de *10-folds cross-validation* e no conjunto de teste, usando as medidas de Acurácia, *Precision* e *Recall*.

Tabela 5. Resultados de classificação para o conjunto de dados D_1 no conjunto de treino, usando *10-folds Cross-Validation*, e no conjunto de teste.

	Treino (*cross-validation*)			**Teste**		
	Acurácia	**Precision**	**Recall**	**Acurácia**	**Precision**	**Recall**
Decision Trees	0.629	0.510	0.600	0.536	0.406	<u>0.479</u>
Random Forest	<u>0.644</u>	<u>0.526</u>	<u>0.640</u>	0.497	0.366	0.444
XGBoost	0.627	0.508	0.554	<u>0.562</u>	<u>0.420</u>	0.402

Observa-se na Tabela 5 que os melhores valores de acurácia, *precision* e *recall* no conjunto de treino foram alcançados pelo modelo de *Random Forest*. Já para o conjunto de teste os melhores valores de acurácia e *precision* foram alcançados pelo algoritmo

XGBoost e de *recall* pelo modelo de *Decision Trees*. Nota-se também que os valores de qualidade dos modelos são menores no conjunto de teste do que no de treino, sendo indiciativo de *overfitting* do modelo sobre o conjunto de treino. Além disso, os valores de acurácia no conjunto de teste não foram suficientes para alcançar a acurácia da classe majoritária, que é 61.9%. Portanto, os modelos construídos para o conjunto D_1 não apresentaram ganho de conhecimento.

Na Tabela 6 são apresentados os resultados experimentais para o conjunto D_2. Verifica-se que o melhores valores de acurácia e *precision* no conjunto de treino foram alcançados pelo *XGBoost*, e de *recall* pelo modelo *Decison Trees*. Já para o conjunto de teste os melhores valores de acurácia e *precision* também foram alcançados pelo modelo *XGBoost*, e o de *recall* pelo modelo *Random Forest*. Diferentemente dos resultados para o conjunto D_1, para o conjunto D_2 não se observou *overfitting* e foram alcançados valores de acurácia notadamente acima da acurácia da classe majoritária, obtendo um ganho de conhecimento de 10.4% (de 61.9% para 73.3%). O modelo *XGBoost* foi considerado o modelo com melhor desempenho, devido ao fato de ter apresentado dois dos três melhores valores de qualidade tanto no conjunto de treino quanto no de teste.

Tabela 6. Resultados de classificação para o conjunto de dados D_2 no conjunto de treino, usando *10-folds Cross-Validation*, e no conjunto de teste.

	Treino (*cross-validation*)			**Teste**		
	Acurácia	**Precision**	**Recall**	**Acurácia**	**Precision**	**Recall**
Decision Trees	0.685	0.582	<u>0.606</u>	0.711	0.611	0.658
Random Forest	0.714	0.630	0.594	0.711	0.609	<u>0.667</u>
XGBoost	<u>0.738</u>	<u>0.781</u>	0.429	<u>0.737</u>	<u>0.765</u>	0.444

Os resultados alcançados, no entanto, estão abaixo de alguns resultados reportados na literatura. Maria et al. (2016) construíram uma Rede Bayesiana para classificação de evasão em cursos técnicos, reportando acurácia média de 85.6% em seus experimentos. Paz and Cazella (2017) utilizaram árvores de decisão para classificação de evasão em cursos de graduação, reportando valores de acurácia superiores a 90.0%.

Para o modelo *XGBoost*, foi realizada a análise da curva *Precision vs. Recall* baseado no score de classificação estimado pelo modelo para cada instância. Nas Figuras 1 e 2 são apresentadas as curvas para os conjuntos de treino e teste, respectivamente. Em cada curva é possível verificar para determinado valor de $Recall$ qual é o valor de $Precision$. Por exemplo, na Figura 1 o ponto destacado indica que é possível recuperar 25.1% das instâncias da classe positiva com uma precisão 86.3%.

Observa-se nas Figuras 1 e 2 que as curvas *Precision vs Recall* no treino e no teste são semelhantes, o que reforça a ideia de ausência de *overfitting* do modelo. Optou-se por definir um limiar de decisão mais estrito, afim de recuperar um menor número de instâncias com uma maior precisão. O ponto destacado no gráfico representa a escolha desse limiar mais estrito, definido em 0.593, com o qual é possível recuperar um número de menor de instâncias mas com maior precisão. Nesse contexto, o modelo *XGBoost* com limiar ajustado tem o potencial de recuperar 25.1% dos alunos com potencial de evasão com uma probabilidade de acerto de 86.3%.

Os resultados alcançados por esse modelo preditivo foram considerados importan-

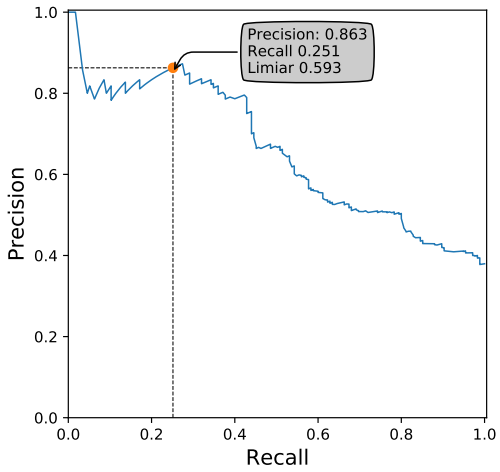

Figura 1. Curva *Precision vs. Recall* no conjunto de treino, usando *cross-validation*.

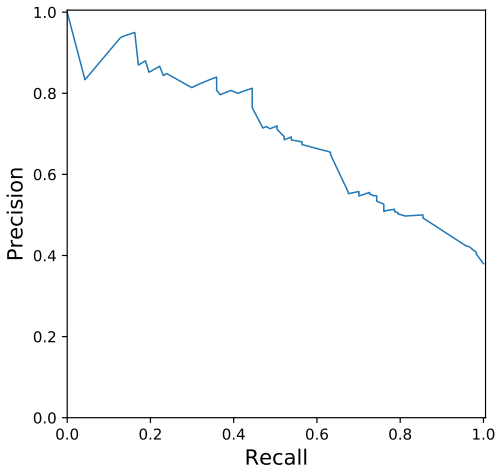

Figura 2. Curva *Precision vs. Recall* no conjunto de teste.

tes e promissores pelo setor administrativo da instituição pois, já após uma semana após o inicio das aulas de cada discente será possível ter uma estimativa de evasão de um grupo razoável de alunos. Para ser mais específico foi realizada a classificação nos conjuntos de treino e de teste usando o modelo com limiar ajustado para classificar os alunos. A Tabela 7 apresenta as tabelas de confusão.

Tabela 7. Tabelas de confusão do modelo *XGBoost* para limiar ajustado para o conjunto D_2.

Predito	**Treino (*10-folds cross-validation*)**		**Teste**	
Observado	**Negativo**	**Positivo**	**Negativo**	**Positivo**
Negativo	279	7	183	8
Positivo	131	44	82	35

Na Tabela 7 observa-se que do total de 461 alunos do conjuntos de treino foram classificados 44 alunos evasores com a classse positiva e 7 não evasores classificados com essa classe. De forma semelhante acontece no conjunto de teste em que foram classificados 35 alunos evasores com classe positiva e apenas 8 não evasores com essa classe. É importante ressaltar que executar ações de permanência e êxito específicas para esses alunos requer uma demanda importante de funcionários da instituição especialistas no assunto que certamente não consegue atender todos os possíveis evasores. Por isso a importância do modelo recuperar um grupo menor de alunos mas com maior precisão.

O modelo construído está sendo integrado à uma ferramenta computacional para utilização pelo setor administrativo do IFSC capus Lages, como uma ação de permanência e exito. A ferramenta está sendo implementada em linguagem *Dash*[2], para construção de *Dashboards* e *Flask*[3] para a criação do serviço Web. Esta ferramenta será utilizada para estudos prospectivos no próximo semestre letivo e poderá auxiliar aos grupos de trabalho de permanência e êxito a direcionar ações específicas que incentivem os alunos a permanecerem na instituição ou bem solicitar o cancelamento de matrícula para permitir que outros alunos interessados, presentes na lista de espera, preencham essas vagas.

[2]`https://plot.ly/dash/`
[3]`https://palletsprojects.com/p/flask/`

4. Conclusão

A oferta do número de vagas em cursos técnicos em instituições públicas do pais vem crescendo anualmente. No entanto, o número de alunos que não chegam a completar o curso vem alertado profissionais da educação. Criar modelos preditivos para estimar o risco de evasão é uma tarefa fundamental para que as instituições possam desenvolver estratégias de permanência e êxito. Neste trabalho foram construídos modelos baseados em árvores de decisão para predizer alunos evasores. Com esses modelos foi possível recuperar 25% dos potenciais evasores com precisão de 86%, resultados considerados promissores pelo setor administrativo para gerenciar ações de permanência e êxito. Trabalhos futuros incluem a inclusão de atributos provenientes de questionários socio-econômicos e a investigação de métodos de para estimar o risco de evasão ao longo do tempo.

Agradecimentos

Ao Instituto Federal de Santa Catarina pelo apoio por meio linha de financiamento de bolsas e projetos do Programa de Iniciação ao Desenvolvimento Tecnológico e Inovação.

Referências

Araújo, F. d. A. and Rodrigues, R. L. (2018). Análise de regressão aplicada a previsão de reprovação de alunos em plataforma de ensino a distância. *Revista de Engenharia e Pesquisa Aplicada*, 3(3).

Brandão, M. F. R., Santos Ramos, C. R., and Tróccoli, B. T. (2003). Análise de agrupamento de escolas e núcleos de tecnologia educacional: mineração na base de dados de avaliação do programa nacional de informática na educação. In *Anais do Simpósio Brasileiro de Informática na Educação (SBIE)*, volume 1, pages 366–374.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Brito, D., Lemos, M. O., Pascoal, T. A., Rêgo, T. G., and Araújo, J. G. G. O. (2015). Identificação de Estudantes do Primeiro Semestre com Cisco de Evasão Através de Técnicas de Data Mining. *Nuevas Ideas en Informática Educativa*, pages 459–463.

Chen, T. and Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 785–794. ACM.

Digiampietri, L. A., Nakano, F., and de Souza Lauretto, M. (2016). Mineraçao de dados para identificaçao de alunos com alto risco de evasao: Um estudo de caso. *Revista de Graduação USP*, 1(1):17–23.

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37–37.

Han, J., Kamber, M., and Pei, J. (2011). *Data Mining: Concepts and Techiniques*. Morgan Kaufmann, 3th edition.

Maria, W., Damiani, J. L., and Pereira, M. (2016). Rede Bayesiana para Previsão de Evasão Escolar. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 5, pages 920–929.

Paz, F. and Cazella, S. (2017). Identificando o Perfil de Evasão de Alunos de Graduação o Através da Mineração de dados Educacionais: um Estudo de Caso de uma Universidade Comunitária. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 6, pages 624–633.

Quinlan, J. R. (2011). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Ramos, J. L. C., Silva, J., Prado, L., Gomes, A., and Rodrigues, R. (2018). Um Estudo Comparativo de Classificadores na Previsão da Evasão de Alunos em EAD. In *Anais do Simpósio Brasileiro de Informática na Educação (SBIE)*, volume 29, pages 1463–1472.