Análise preditiva em bases desbalanceadas e comparação de técnicas de pré-processamento – Estudo de caso MOOC

Bruno Bastos Stoll
Departamento de Informática
Universidade Federal do Espírito Santo (UFES)
brunobstoll@hotmail.com

Lucas Varo Daros
Programa de Pós-Graduação em Engenharia de Controle e Automação (ProPECAut)
Instituto Federal do Espírito Santo (IFES)
lucasdaros@gmail.com

Davidson Cury
Departamento de Informática
Universidade Federal do Espírito Santo (UFES)
dede@inf.ufes.br

Crediné Silva de Menezes
Programa de Pós-Graduação em Educação
Universidade Federal do Rio Grande do Sul (UFRGS)
credine@inf.ufes.br

ABSTRACT
The field of teaching and learning is being revolutionized by online courses that are openly and massively offered (MOOC) as they serve as a knowledge platform for anyone, anytime and anywhere. However, most students who take these courses drop out, and it is a challenge to predict such a phenomenon because of the nature of this data imbalance, which occurs when the class attribute has a much larger number of examples than the other in predictive systems. The aim of this paper is to compare two preprocessing methods for predictive analysis on unbalanced bases, called Oversample and Undersample, both related to the Resampling approach. To perform the analysis, predictive models were generated in different ways using a real MOOC data set. The results initially indicate that the approach can significantly improve the accuracy and efficiency of predictive algorithms.

Keywords
Learning Analytics; Data Mining; Pre-Processing; Machine Learning.

RESUMO
O campo da ensino e aprendizado vem sendo revolucionado por cursos online que são ofertados de forma aberta e massiva (MOOC), pois servem como plataforma de conhecimento para qualquer um, a qualquer hora e em qualquer lugar. Contudo, a maioria dos alunos que participam desses cursos o abandonam, tornando-se um desafio prever tal fenômeno devido à natureza do desbalanceamento desses dados, que ocorre quando o atributo classe tem um número muito maior de exemplos que a(s) outra(s) em sistemas de preditivos. O objetivo deste artigo é comparar dois métodos pré-processamento para análise preditiva em bases desbalanceadas, chamados de Oversample e Undersample, ambos relacionados a abordagem de Resampling. Para efetuar a análise, foram gerados os modelos preditivos de diferentes formas usando um conjunto de dados de um MOOC real. Os resultados inicialmente indicam que a abordagem pode melhorar significativamente a acurácia e a eficiência de algoritmos preditivos.

Palavras-Chaves
Análise do Aprendizado; Mineração de Dados; Pré-Processamento; Aprendizado de Máquina.

1. INTRODUÇÃO
A ideia geral de cursos online que são ofertados de forma aberta e massiva, o inglês Massive Open Online Courses (MOOCs), é de servirem como plataformas de conhecimento para qualquer um, a qualquer hora, e em qualquer lugar, o que faz deles uma emergente e poderosa estratégia de aprendizagem com repercussão nas áreas tecnológica e educacional [6].

Contudo, há uma grande quantidade de evasão de alunos nesses cursos [6]. Conhecer o perfil desses alunos pode se tornar uma ótima estratégia para se atuar na redução da evasão. Dessa forma, se torna um desafio para algoritmos de aprendizado de máquina preverem tais fenômenos. Pois eventos raros são difíceis de prever e podem resultar em alto custo para algoritmos classificadores. O desbalanceamento de dados ocorre quando uma das classes tem um número muito maior de exemplos que a(s) outra(s). A classe mais prevalecente é chamada de majoritária, enquanto a classe mais rara é chamada de classe minoritária.

O objetivo deste artigo é comparar dois métodos pré-processamentos para análise preditiva em bases desbalanceadas. A abordagem utilizada chama-se de Resampling, que administra quantidade de amostra de dados realizando uma redistribuição. Há duas técnicas de pré-processamento relacionadas à essa abordagem, chamadas de Oversample e Undersample, ambos descritos em detalhes na seção 2 fundamentação teórica.

Este trabalho consiste na evolução das pesquisas [20] e [21] que corresponde a um framework para análise e intervenção no processo de aprendizado. Um framework é, em geral conceituado, como uma estrutura real ou conceitual destinada a servir de suporte ou guia para a construção de um software [16]. O conceito de framework para essas pesquisas é entendido como um modelo conceitual de elementos extensível de ferramentas de software caracterizados, que podem ser utilizados para implementar uma solução computacional.

Nesta pesquisa foi usada uma base de dados real de um MOOC. Foram gerados três grupos de modelos preditivos para comparar algoritmos e métodos de pré-processamentos. Na metodologia, utilizou-se uma abordagem quantitativa, de natureza aplicada, com o objetivo explicativo e com procedimentos experimentais. Como resultados, após aplicar as técnicas obteve-se uma melhora significativa na acurácia e na eficiência dos algoritmos.

O artigo foi organizado em 6 seções. Na seção 2 descreve fundamentação teórica com conceitos sobre Mineração de Dados Educacionais, Aprendizado de Máquina, Medidas de Aprendizado Supervisionado e Pré-Processamentos. Na seção 3 são apresentados os trabalhos correlatos. Os experimentos são descritos na seção 4. E na seção 5 é apresentada a conclusão.

2. FUNDAMENTAÇÃO TEÓRICA

Nesta seção são apresentados conceitos teóricos sobre análise do aprendizado, mineração de dados educacionais, aprendizado de máquina, técnicas de pré-processamento com abordagens Resampling e medidas para avaliação dos algoritmos.

A aprendizado, do inglês *Learning Analytics* (LA) é descrito como um campo emergente no qual ferramentas de análise de dados são usadas para melhorar o ensino e o aprendizado [1], bem como a aplicação de Web *Analytics*, tendo em vista o perfil do aluno, um processo de coleta e análise de detalhes de interações individuais de alunos em atividades de aprendizado online com o objetivo de melhorar o aprendizado [9].

A Mineração de Dados Educacionais, em inglês *Educational Data Mining* (EDM), é a aplicação de técnicas de mineração de dados para tipos específicos de conjuntos de dados provenientes de ambientes educacionais, usando modelos analíticos que permitem descobrir padrões interessantes e tendências em informações de cada aluno. O processo de EDM converte os dados brutos de sistemas educacionais em informações úteis que podem impactar na prática e na pesquisa educacional [18]. Em uma visão mais abrangente, definida em [19], o EDM pode ser visualizado como a combinação das principais áreas: Ciência da Computação, Educação e Estatística, conforme apresentado na Figura 1.

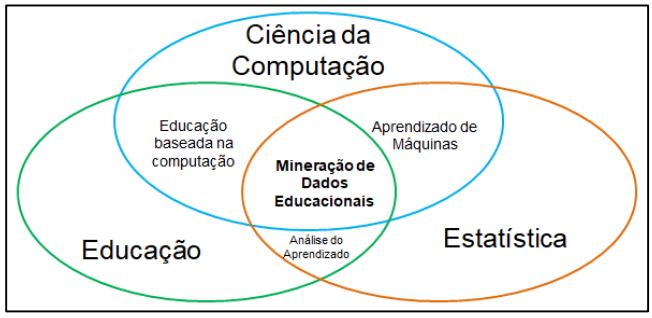

Figura 1. Combinação das áreas da aplicação do EDM

O aprendizado de máquina, do inglês *Machine Learning* (ML) é uma subárea da Inteligência Artificial, com o objetivo de desenvolver técnicas computacionais sobre o aprendizado, bem como a construção de sistemas capazes de adquirir conhecimento de forma automática. Sistema, que usa aprendizado de máquina, toma decisões baseadas em experiências acumuladas através de soluções bem-sucedidas de um problema anterior [17].

A árvore de decisão, do inglês *Decision Tree* (Tree), é um modelo de árvore recursivo baseado em partição para prever qual é o rótulo de uma instância [24]. O K Vizinhos Próximos, do inglês *K-Nearest Neighbors* (KNN) é um algoritmo de aprendizado de máquina que calcula a distância dos k (instâncias) vizinhos mais próximos [24]. A Máquina de Suporte a Vetor, do inglês *Support Vector Machine* (SVM), é um algoritmo de aprendizado de máquinas com o objetivo de encontrar o hiperplano ideal que maximiza a folga entre classes [24]. O classificador de Bayes, que é uma abordagem de classificação probabilística e usa o teorema de Bayes para predizer valores [24]. O Perceptron em Multicamadas, em inglês *Multi-Layer Perceptron* (MLP), é uma rede neural que apresenta uma ou mais camadas intermediárias de neurônios e uma camada de saída, onde funções de ativação não lineares, como a função *sigmoidal*, são utilizadas nas camadas intermediárias [5].

O processo de validação de qualquer modelo de aprendizado de máquina geralmente envolve a realização de experimentos controlados, em que se demonstre a sua efetividade. Em problemas binários, ou seja, duas classes alvos, usualmente uma classe é denotada positiva (+) e a outra é denominada negativa (-). A acurácia é a taxa total de acertos. A sensibilidade é a taxa de acerto na classe positiva. A especificidade é a taxa de acerto da classe negativa. E a eficiência é a média aritmética da sensibilidade e da especificidade [24]. O processo de aprendizagem guiado por medidas de desempenho global, como a acurácia induz um preconceito para a classe majoritária, enquanto os episódios raros permanecem desconhecidos, até mesmo se o modelo de predição produzir alta capacidade preditiva [7].

O pré-processando de dados é uma etapa na mineração de dados que inclui limpeza, normalização, transformação, extração de característica e seleção, etc. O produto final dele é o treinamento final do algoritmo para a geração de um modelo preditivo. O pré-processamento de dados pode ter frequentemente um impacto significante em desempenho desses algoritmos ML supervisionado [10].

Dentre as abordagens de pré-processamento, destaca-se a de Resampling que é utilizada neste trabalho. Ela faz a redistribuição de amostras em bases desbalanceadas para aliviar o efeito da distribuição de classe inclinada no processo de aprendizagem de máquina independentes do classificador selecionado. Nessa abordagem as técnicas de Oversample e Undersample. O Oversample consiste na geração de amostras sintéticas minoritárias. E o Undersample consiste em eliminar registros da classe majoritária [15]. Na Figura 2, é realizada a comparação da base desbalanceada com as amostras de dados após o uso da abordagem Resampling.

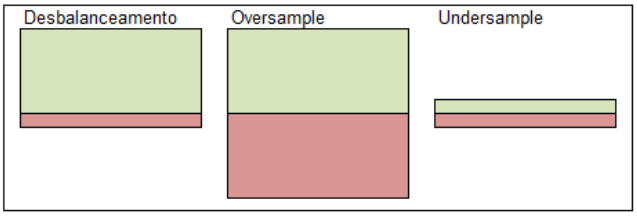

Figura 2. Comparação da base desbalanceada com as amostras após o uso das abordagens Resampling

3. TRABALHOS RELACIONADOS

Nos últimos anos diversos trabalhos têm explorado os benefícios que o MDE traz ao ambiente educacional. O uso de diferentes abordagens tem aumentado o conhecimento sobre o uso de aprendizado de máquina para prever resultados de alunos. No entanto, a natureza das bases de dados educacionais poderá degradar a capacidade de prever seus resultados devido à dificuldade na detecção de eventos raros a partir de uma perspectiva de aprendizagem desequilibrada (bases desbalanceadas).

Os métodos de modelagem para esse caso incluem técnicas como pré-processamento de dados, algoritmos de classificação e avaliação de modelos [3]. O trabalho [13] propõe a seleção dos

melhores atributos para redução de dimensionalidade e uso do algoritmo SMOTE para reequilíbrio dos dados utilizando a ferramenta Weka, em uma base de dados reais, com cerca de 670 estudantes do ensino médio de Zacatecas, no México. A pesquisa [22] usa técnicas para prever o desempenho de estudantes atacando questões relacionadas ao desbalanceamento de classes com uso da técnica Oversample, usando o algoritmo SMOTE para (re)distribuição das classes. A pesquisa [23] combina algoritmos de pré-processamento de seleção de atributos com o algoritmo SMOTE em bases desbalanceadas. Já o trabalho [12] demonstra que o Oversample melhora a precisão de algoritmos classificadores em bases desbalanceadas. E a pesquisa [4] analisa os dados de perfis de pacientes através de técnicas de classificação em bases desbalanceadas, no qual um rebalanceamento é realizado de forma aleatória para auxiliar algoritmos preditivos.

Observa-se que os trabalhos acima usam técnicas para tratamento de desbalanceamento. No entanto, nenhum dos trabalhos citados comparou os resultados das técnicas de Oversample e Undersample.

4. EXPERIMENTOS

Com o intuito de tornar os dados mais adequados para o uso de ML em bases de dados desbalanceados, objetiva-se aplicar a abordagem de pré-processamento Resampling usando as técnicas Undersample e Oversample. Após os pré-processamentos as classes ficaram com 50% cada uma (classe binária). Três grupos (b.1 / b.2 / b.3) de modelos preditivos foram treinados e testados. No primeiro grupo (b.1), os modelos foram gerados sem o uso de nenhuma técnica de pré-processamento Resampling, com o objetivo de servir de comparação com os demais grupos. No segundo (b.2), foram gerados modelos preditivos utilizando os dados rebalanceados com o uso da técnica de pré-processamento Oversample. No terceiro (b.3), foram gerados modelos preditivos utilizando os dados rebalanceados com o uso da técnica de pré-processamento Undersample.

A linguagem de programação Python foi usada para execução dos experimentos, com a biblioteca scikit-learn para uso de ML, foi utilizado o Pandas e Numpy para manipulação dos dados e imblearn para pré-processamento de dados desbalanceados. O Jupyter Notebook foi usado como plataforma de programação. Os experimentos foram realizados em um computador com o processador de 64 bits Core i5-6500T de 2.50GHz, 16,0 GB de memória RAM.

Conforme apresentado na Figura 3, os experimentos foram realizados em etapas, que são agrupados em (A) Inicialização, (B) Treinamento e Testes e (C) Resultados. Na (A) Inicialização foi realizado o tratamento nos dados para reaproveitá-los nas próximas etapas. No (B) Treinamento e Testes, foram gerados três conjuntos de modelos preditivos e suas medidas através de testes. Os resultados são analisados, comparados e discutidos na etapa (C) Resultados.

(A) Inicialização

Para realizar esta pesquisa utilizou-se a base de dados real MOOC, com a descrição do conjunto de atributos apresentado na Tabela 1. Essa massa de dados consiste em dados coletados e consolidados pela plataforma Edx das universidades de Harvard e MIT, no ano de 2013 (ano acadêmico de 2013: Fall 2012, Spring 2013 e Summer 2013). Esses dados são registros agregados, e cada registro representa a atividade de um indivíduo [14]. Conforme apresentado na Figura 4, Figura 3 a quantidade de alunos que concluem o curso (*certified* igual a 1) é muito menor do que os alunos que terminam o curso (*certified* igual a 1), com uma proporção de **97,24%** de alunos que não concluíram o curso e somente **2,76%** que concluíram o curso. Logo, configura-se em uma base de dados altamente desbalanceada.

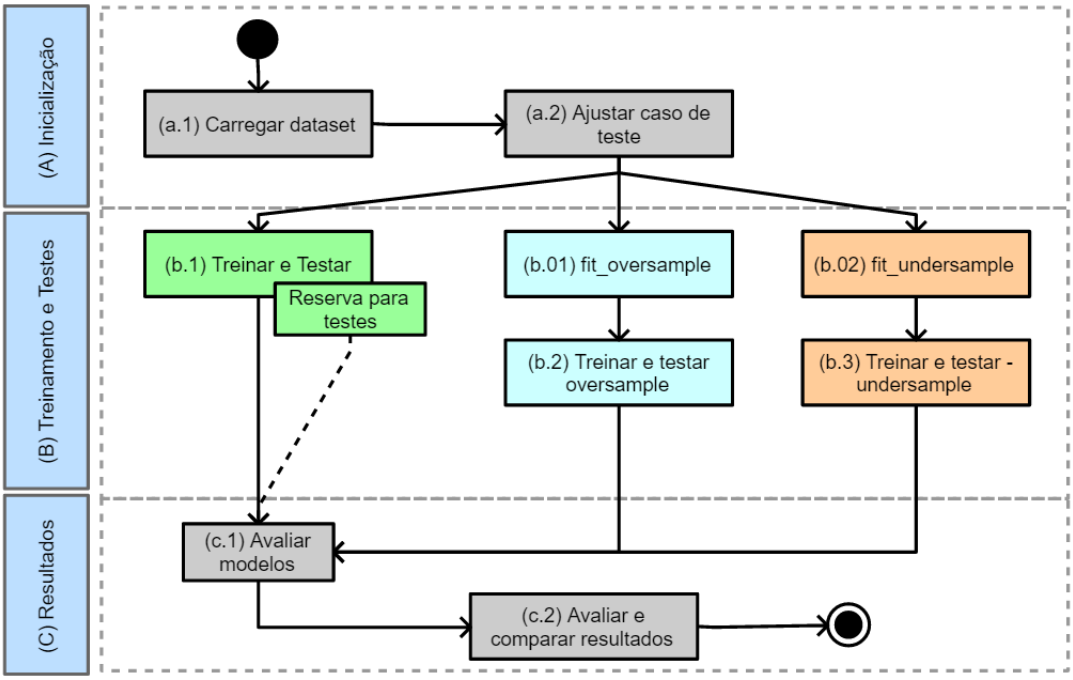

Figura 3. Método de análise preditiva

Tabela 1. Atributos do conjunto de dados

Coluna	Descrição
YoB	Ano do nascimento
start_month	Mês do início
start_year	Ano do início
last_e_year	Ano último evento
last_e_month	Mês último evento
course_id	Instituto (HarvardX / MitX) + Ano(2012 / 2013) + Semestre (Fall / Spring / Summer)
country	País do aluno
degree	Graduação do aluno (Bachelor's / Doctorate / Less than Secondary / Master's / Secondary)
gender	Gênero (f / m)
certified	Certificado - coluna classe (0: não certificado; 1: certificado)

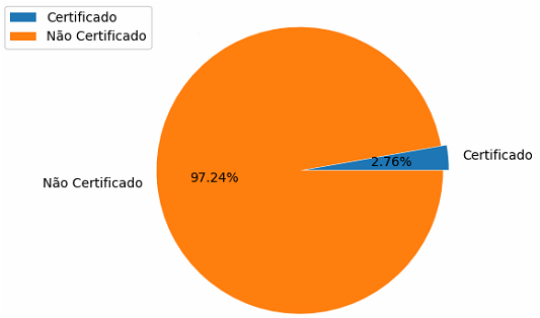

Figura 4. Desbalanceamento da classe

Etapa (a.1): Foi realizada a importação dos dados, carregado em memória e disponibilizados para a próximas etapa.

Etapa (a.2): Foram eliminadas as dimensões que não seriam utilizadas no experimento, reduzindo a quantidade de atributos para 10, conforme apresentado na Tabela 1. Nem todas as colunas foram usadas para tornar o teor tecnológico mais evidente. Foi realizada a binarização nas dimensões categóricas. O conjunto de dados, ao final dessas transformações, ficou com 68 dimensões e 641.138 instâncias. E por fim, os dados foram disponibilizados para os três grupos de experimentos (b.1, b.2 e b.3).

(B) Treinamento e Testes

Nestes três grupos de experimentos foram gerados modelos capazes de classificar alunos que terminam ou desistem do curso (certified 0 ou 1). Para avaliar os algoritmos foi usada a técnica de validação cruzada, onde é feita a divisão dos dados já rotulados para treinamento e teste para validar os acertos [5]. O parâmetro de separação de treinamento e teste foi usado com a proporção 70% e 30% respectivamente, de forma randômica. Foram utilizados os algoritmos Tree, KNN, SVM, MLP e Naive, pois cada um possui uma abordagem diferente. O algoritmo chamado Dummy também usado, que classifica todos os dados com a classe majoritária. As medidas utilizadas para analisar os algoritmos são a acurácia (**acur**), sensibilidade (**sens**), especificidade (**esp**), eficiência (**efic**) e tempo de processamento de treino em minutos (**TExec**). A eficiência foi utilizada como medida para avaliar a capacidade preditiva dos modelos e comparada com a medida de acurácia. Os hiperparâmetros de cada um dos algoritmos estão descritos na Tabela 2.

Tabela 2. Algoritmos e Hiperparâmetros

Sigla	Algoritmo	Hiperparâmetros
Tree	DecisionTreeClassifier	class_weight=None, criterion='gini', max_depth=5, max_features=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, presort=False, random_state=None, splitter='best'
KNN	KNeighborsClassifier	algorithm='auto', leaf_size=30, metric='minkowski', metric_params=None, n_jobs=None, n_neighbors=3, p=2, weights='uniform'
SVM	SVC SuportVectorClassifier	C=1.0, cache_size=200, class_weight=None, coef0=0.0, decision_function_shape='ovr', degree=3, gamma='auto_deprecated', kernel='sigmoid', max_iter=-1, probability=False, random_state=None, shrinking=True, tol=0.001, verbose=False
MLP	MLPClassifier	activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9, beta_2=0.999, early_stopping=False, epsilon=1e-08, hidden_layer_sizes=100, learning_rate='constant', learning_rate_init=0.001, max_iter=200, momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5, random_state=None, shuffle=True, solver='adam', tol=0.0001, validation_fraction=0.1, verbose=False, warm_start=False
Naive	GaussianNB	priors=None, var_smoothing=1e-09
Dummy	DummyClassifier	constant=None, random_state=None, strategy='prior'

Tabela 3. Medidas dos modelos: b.1) Sem técnica de pré-processamento; b.1) Uso da técnica Oversample; b.2) Uso da técnica Undersample

	algoritmo	acur	sens	esp	efic	TExec
Modelos b.1 Sem Técnica de Pré-processamento	Tree	97.20	97.20	0.00	48.60	0.04
	KNN	**96.81**	97.97	40.37	**69.17**	7.60
	SVM	97.20	97.20	0.00	48.60	27.78
	MLP	97.20	97.20	0.00	48.60	9.97
	Naive	65.73	99.40	6.66	53.03	0.04
	Dummy	97.20	97.20	0.00	48.60	0.00
	algoritmo	**acur**	**sens**	**esp**	**efic**	**TExec**
Modelos b.2 Oversample (SMOTE)	Tree	89.51	89.47	89.54	89.51	0.18
	KNN	**96.78**	96.56	97.00	**96.78**	47.88
	SVM	49.96	0.00	49.96	24.98	3348.35
	MLP	90.63	94.52	87.35	90.94	46.72
	Naive	76.60	88.63	70.26	79.45	0.08
	Dummy	50.04	50.04	0.00	25.02	0.01
	algoritmo	**acur**	**sens**	**esp**	**efic**	**TExec**
Modelos b.3 Undersample (NCR)	Tree	88.53	88.65	88.42	88.53	0.00
	KNN	**93.69**	94.17	93.22	**93.69**	0.12
	SVM	50.03	0.00	50.03	25.02	2.56
	MLP	65.84	99.53	59.45	79.49	0.07
	Naive	75.94	87.88	69.74	78.81	0.00
	Dummy	49.97	49.97	0.00	24.98	0.00

Etapa (b.1): No primeiro grupo (b.1) foram gerados modelos preditivos sem uso de técnicas de pré-processamentos para tratar desbalanceamentos. Os dados de testes foram reservados para avaliações dos modelos b.2 e b.3 na etapa c.1. Conforme apresentado na Tabela 3, destaca-se o algoritmo KNN que teve a maior eficiência, que obteve a acurácia de 96.81 e eficiência de 69.17. A eficiência é muito mais baixa que a acurácia devido ao desbalanceamento da base de dados e a dificuldade em prever fenômenos mais raros, neste caso a classe negativa, conforme a mostra a especificidade.

Etapa (b.01): No segundo grupo foram criadas novas instâncias da classe minoritária com a técnica de pré-processamento chamada de Oversample, usando o algoritmo SMOTE (*Synthetic Minority Oversampling Technique*), com o parâmetro *random_state* igual a 42, medida padrão do algoritmo. Esse algoritmo efetua super amostragem da classe positiva, criando novas instâncias "sintéticas", ao invés da seleção de amostras [2]. A execução do algoritmo SMOTE resultou num total de 623.451 instâncias para a classe minoritária, ou seja, foram geradas 605.764 instâncias, igualando a classe minoritária com a majoritária.

Etapa (b.02): No terceiro grupo foram eliminados registros da classe majoritária utilizando a técnica de undersample. O algoritmo NeighbourhoodCleaningRule (NCR) foi usado com o parâmetro *random_state* igual a 42, medida padrão do algoritmo. O algoritmo consiste em manter todas as instâncias da classe minoritária e reduzir número de instâncias da classe majoritária baseado em proximidade de vizinhos [11]. A execução do algoritmo NCR resultou um total de 17.687 classes majoritárias, ou seja, foram descartadas (limpas) 605.764 instâncias da classe majoritária.

Etapa (b.2): Após o pré-processamento de Oversample, foi executado o treinamento e testes dos algoritmos e foi obtido o resultado apresentado na Tabela 3. Destaca-se o algoritmo KNN com uma acurácia de 96.78 e uma eficiência 96.78. Por fim, os modelos treinados foram reservados para a avaliação na etapa c.1.

Etapa (b.3): Após o pré-processamento de Undersample, foi executado o treinamento e testes dos algoritmos e foi obtido o resultado destacado na Tabela 3. Dentre esse grupo de modelos, destaca-se o algoritmo KNN com uma acurácia de 93.69 e uma eficiência 93.69. Por fim, esse grupo foi reservado para a avaliação na etapa c.1.

(C) Resultados

Nesta etapa os resultados dos modelos gerados nas etapas b.1, b.2 e b.3 são avaliados, discutidos e comparados. Na etapa c.1 os modelos foram reavaliados com os dados originais. Como nas etapas b.2 e b.3 tiveram uma mudança significativa nos dados de treinamento, foram realizados testes com esses modelos usando os dados originais reservados na etapa b.1. Foram selecionados os algoritmos que tiveram a maior eficiência e comparados na etapa c.2. Os resultados desse experimento indicam a coerência das medidas de testes geradas na etapa c.1.

Etapa (c.1): Os modelos treinados nas etapas **b.2** e **b.3** foram reavaliados com dados reais reservados na etapa **b.1**, conforme apresentado na Tabela 4. Os modelos que tiveram a maior eficiência foram selecionados para comparação na etapa c.2. Dentre os modelos do grupo **b.2**, o algoritmo **KNN** foi selecionado, pois obteve uma acurácia de **93.19** e uma eficiência de **93.38**. E dentre os modelos do grupo **b.3**, o algoritmo KNN foi selecionado, e teve uma acurácia de **93.69** e uma eficiência de **93.69**. Os resultados desta etapa indicam a coerência nos resultados da etapa b.2 e b.3.

Tabela 4. Comparação entre os modelos classificadores treinados nas etapas b.2 e b.3 com dados reequilibrados com uso da abordagem Resampling e testados com a base real

	algoritmo	acur	sens	esp	efic
Modelos b.2 Oversample (SMOTE)	Tree	88.02	87.68	88.37	88.03
	KNN	**93.19**	90.46	96.31	**93.38**
	SVM	50.03	0.00	50.03	25.02
	MLP	89.88	93.14	87.08	90.11
	Naive	74.72	83.47	69.61	76.54
	Dummy	49.97	49.97	0.00	24.98
	algoritmo	**acur**	**sens**	**esp**	**efic**
Modelos b.3 Undersample (RCN)	Tree	88.53	88.65	88.42	88.53
	KNN	**93.69**	94.17	93.22	**93.69**
	SVM	50.03	0.00	50.03	25.02
	MLP	65.84	99.53	59.45	79.49
	Naive	75.94	87.88	69.74	78.81
	Dummy	49.97	49.97	0.00	24.98

Etapa (c.2): Os algoritmos que tiveram a melhor eficiência etapas b.1, b.2, b.3 e c.1 foram reservados e comparados, conforme apresentado na Tabela 5. A abordagem de Resampling com as técnicas de Oversample (b.2) e Resampling (b.3) apresentam a acurácia e a eficiência significativamente melhores, quando comparados com os mesmos algoritmos treinados sem uso da abordagem Resampling (b.1). O algoritmo selecionado na etapa **b.1** foi o **KNN**, que obteve uma eficiência de **69.17**. Dentre os grupos de modelos **b.2** e **b.3**, o algoritmo **KNN** foi selecionado, pois teve a maior eficiência. O algoritmo KNN no grupo de modelos **b.2**, obteve uma eficiência 24.21 maior que o mesmo algoritmo do grupo **b.1**. E o algoritmo KNN no grupo de modelos **b.3**, teve uma eficiência um pouco maior, com uma diferença de 24.52 da eficiência do KNN do grupo b.1. Em casos de desbalanceamento, a Acurácia tem seu desempenho disfarçado pela Especificidade ou Sensibilidade. E nesse experimento a medida Especificidade, que estava prejudicada, teve uma melhora.

Tabela 5. Comparação final da eficiência entre os modelos mais aderentes com a natureza dos dados

	Alg	Sintéticos / eliminados	Dados reais
Modelo b1. Sem Resampling	KNN	-	69.17
Modelo b2. Oversample SMOTE	KNN	96.78	93.38
Modelo b3. Undersample RCN	KNN	93.38	93.69

Após a análise, é possível destacar os pontos de discussão: I) o tempo de execução; II) as medidas de ML; e III) a ineficiência do algoritmo SVM.

I) Custo computacional para o algoritmo SVM, tendo mais de 55 horas para treinar do algoritmo. Esse algoritmo é muito lento, principalmente em grandes bases de dados [25]. Ele exige que cada instância de dados seja representada como um vetor de números reais, compondo uma parte do pré-processamento interno dele. Se houver atributos categóricos, primeiro é necessário convertê-los em dados numéricos, para representar o atributo em zero e um (por exemplo, o atributo categórico {*red, green, blue*} é representado como [0,0,1], [0,1,0] e [1,0,0]) [8]. Um fator-chave que atua na complexidade do tempo de execução para um SVM é o parâmetro C (regulação de folga), sendo um parâmetro altamente sensível em reação ao custo computacional e a capacidade preditiva.

II) As medidas de ML de eficiência e acurácia são igualmente boas para avaliação de um modelo, após o uso das técnicas de pré-processamento Oversample e Undersample. Dessa forma, a acurácia passa a ser uma medida de qualidade de algoritmo igualmente adequada para avaliar um algoritmo de ML, após o pré-processamento.

III) Na ineficiência do modelo SVM também é possível destacar, pois se percebe que o SVM obteve uma eficiência tão baixa quanto o algoritmo Dummy, que nem é um algoritmo de ML. Havendo uma necessidade de reavaliá-lo usando outros hiperparâmetros, principalmente o C (usado como '1') e o *kernel* (usado como '*sigmoid*').

5. CONCLUSÃO

A previsão de desempenho de alunos em MOOCs pode ser uma tarefa complexa. Não só porque existem vários problemas (como atores pessoais, familiares, sociais e econômicos que podem influenciar), como também devido à natureza de desbalanceamento das bases de dados. Nesse contexto, neste artigo foram realizados três experimentos, um sem o uso de técnicas de pré-processamento para bases desbalanceadas e os outros utilizando as técnicas de Oversample e Undersample na finalidade de confrontar os resultados. Os resultados iniciais indicam uma melhora na capacidade preditiva de algoritmos classificadores usando as técnicas de pré-processamento. Assim, é possível concluir que com o uso das técnicas de pré-processamento Oversample e Undersample, é possível ter um ganho de acurácia e eficiência em algoritmos classificadores, contudo diferenciando-se apenas pelo custo computacional. Sendo o Oversample uma técnica com um custo computacional elevado, já a técnica Undersample tem um custo computacional menor.

6. REFERÊNCIAS

[1] Chatti, M. A., Dyckhoff, A. L., Schroeder, U., & Thüs, H. (2013). A reference model for learning analytics. *International Journal of Technology Enhanced Learning*, *4*(5-6), 318-331.

[2] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321-357.

[3] Chawla, N. V., Japkowicz, N., & Kotcz, A. (2004). Special issue on learning from imbalanced data sets. *ACM Sigkdd Explorations Newsletter*, *6*(1), 1-6.

[4] Darós, L. V., Komati, K. S., & Resendo, L. C. (2018, October). Diferentes abordagens de Subamostragem para Balanceamento da Base de Dados aplicados ao estudo de caso da Classificação de Absenteísmo de Pacientes Clínicos. In *Anais da V Escola Regional de Sistemas de Informação do Rio de Janeiro* (pp. 54-61). SBC.

[5] Faceli, K., Lorena, A. C., Gama, J., & Carvalho, A. C. P. D. L. (2011). Inteligência Artificial: Uma abordagem de aprendizado de máquina.

[6] Fassbinder, A., Delamaro, M. E., & Barbosa, E. F. (2014). Construção e uso de moocs: uma revisão sistemática. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)* (Vol. 25, No. 1, p. 332).

[7] Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, *73*, 220-239.

[8] Hsu, C. W., Chang, C. C., & Lin, C. J. (2003). A practical guide to support vector classification.

[9] Johnson, W. B. (1988). Pragmatic considerations in research, development, and implementation of intelligent tutoring systems (pp. 191-207). Hillsdale, NJ: Lawrence Erlbaum.

[10] Kotsiantis, S. B., Kanellopoulos, D., & Pintelas, P. E. (2006). Data preprocessing for supervised leaning. *International Journal of Computer Science*, *1*(2), 111-117.

[11] Laurikkala, J. (2001, July). Improving identification of difficult small classes by balancing class distribution. In *Conference on Artificial Intelligence in Medicine in Europe* (pp. 63-66). Springer, Berlin, Heidelberg.

[12] Markowski, G., Grabczewski, K., & Adamczak, R. (2016). Oversampling negative class improves contact map prediction. *Int J Pharma Med Biol Sci*, *5*(4), 211-216.

[13] Márquez-Vera, C., Cano, A., Romero, C., & Ventura, S. (2013). Predicting student failure at school using genetic programming and different data mining approaches with high dimensional and imbalanced data. *Applied intelligence*, *38*(3), 315-330.

[14] MITx and HarvardX (2014). HarvardX-MITx Person-Course Academic Year 2013 De-Identified dataset, version 2.0.

[15] Mollineda, R., Alejo, R., & Sotoca, J. (2007, September). The class imbalance problem in pattern classification and learning. In *II Congreso Espanol de Informática (CEDI 2007). ISBN* (pp. 978-84).

[16] Pree, W. (1994, July). Meta patterns—a means for capturing the essentials of reusable object-oriented design. In *European Conference on Object-Oriented Programming* (pp. 150-162). Springer, Berlin, Heidelberg.

[17] Richert, W. (2013). Building machine learning systems with Python. Packt Publishing Ltd.

[18] Romero, C., & Ventura, S. (2010). Educational data mining: a review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *40*(6), 601-618.

[19] Romero, C., & Ventura, S. (2013). Data mining in education. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *3*(1), 12-27.

[20] Stoll, B. B., Cury, D., & de Menezes, C. S. (2018). Framework para predições e recomendações em dados acadêmicos. *RENOTE-Revista Novas Tecnologias na Educação*, *16*(2), 413-422.

[21] Stoll, B. B., Cury, D., de Lira Tavares, O., & de Menezes, C. S. (2019). Análise de dados acadêmicos baseado em previsão, recomendação e visualização. *RENOTE-Revista Novas Tecnologias na Educação*, *17*(1), 286-295.

[22] Thai-Nghe, N., Busche, A., & Schmidt-Thieme, L. (2009, November). Improving academic performance prediction by dealing with class imbalance. In *2009 Ninth International Conference on Intelligent Systems Design and Applications* (pp. 878-883). IEEE.

[23] Yang, R., Zhang, C., Zhang, L., & Gao, R. (2018). A two-step feature selection method to predict Cancerlectins by Multiview features and synthetic minority oversampling technique. *BioMed research international*, *2018*.

[24] Zaki, M. J., Meira Jr, W., & Meira, W. (2014). *Data mining and analysis: fundamental concepts and algorithms*. Cambridge University Press.

[25] Zeng, Z. Q., Yu, H. B., Xu, H. R., Xie, Y. Q., & Gao, J. (2008, November). Fast training support vector machines using parallel sequential minimal optimization. In *2008 3rd international conference on intelligent system and knowledge engineering* (Vol. 1, pp. 997-1001). IEEE.