UNIVERSIDADE DO EXTREMO SUL CATARINENSE - UNESC
CURSO DE CIÊNCIA DA COMPUTAÇÃO

LAÍNE DIMER

MODELO DE PREDIÇÃO UTILIZANDO COMITÊ DE CLASSIFICADORES PARA IDENTIFICAÇÃO DE PERFIS DE INTERAÇÃO NO AMBIENTE VIRTUAL DE APRENDIZAGEM

CRICÍUMA
2019

LAÍNE DIMER

MODELO DE PREDIÇÃO UTILIZANDO COMITÊ DE CLASSIFICADORES PARA IDENTIFICAÇÃO DE PERFIS DE INTERAÇÃO NO AMBIENTE VIRTUAL DE APRENDIZAGEM

Trabalho de Conclusão de Curso, apresentado para obtenção do grau de Bacharel no curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC.

Orientadora: Profa. Dra. Merisandra Côrtes de Mattos Garcia

CRICIÚMA

2019

LAÍNE DIMER

MODELO DE PREDIÇÃO UTILIZANDO COMITÊ DE CLASSIFICADORES PARA IDENTIFICAÇÃO DE PERFIS DE INTERAÇÃO NO AMBIENTE VIRTUAL DE APRENDIZAGEM

Trabalho de Conclusão de Curso aprovado pela Banca Examinadora para obtenção do Grau de Bacharel, no Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC, com Linha de Pesquisa em Inteligência Artificial.

Criciúma, 28 de julho de 2019

BANCA EXAMINADORA

Profa. Merisandra Côrtes de Mattos Garcia – Doutora – (UNESC) - Orientadora

Profa. Graziela Fátima Giacomazzo - Doutora – (Programa de Pós-Graduação em Educação e Setor de Educação à Distância – UNESC)

Prof. Kristian Madeira - Doutor – (UNESC)

Prof. Luiz Ricardo Fiera - Mestre – (ESUCRI)

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AVA	Ambiente Virtual de Aprendizagem
DCBD	Descoberta de Conhecimento em Bases de Dados
EAD	Educação a Distância
EDM	*Educational Data Mining*
KDD	*Knowledge Discovery in Database*
RS	*Random Subspace*
WEKA	*Waikato Environment for Knowledge Analysis*

SUMÁRIO

AGRADECIMENTOS

Gostaria de agradecer o setor de Educação a Distância da UNESC por disponibilizar os dados utilizados na pesquisa e por todo apoio para realização deste trabalho.

Agradeço ao setor de Tecnologia e Informação da UNESC pela prontidão e por disponibilizar a base de dados utilizada.

Agradecer a minha orientadora, Professora Doutora Merisandra Cortês de Mattos Garcia e colega Alini Marangoni Eyng, pelo apoio, dedicação e envolvimento para tornar este trabalho possível.

RESUMO

A procura por padrões de comportamento, hábitos e escolhas ocorre desde o início da vida humana. Atualmente, a sociedade se abastece de informações, as quais são encontradas em sua forma bruta, em dados, sendo necessário compreendê-los. A descoberta de conhecimento em base de dados, do inglês *knowledge discovery in database*, é um processo constituído por etapas para encontrar padrões e modelos válidos. A etapa de *data mining*, é responsável por explorar grandes quantidades de dados para encontrar os modelos, com o aumento da procura por padrões e perfis de estudantes em ambientes virtuais de aprendizagem, surgiu a mineração de dados educacionais, do inglês *educational data mining*, a qual adapta tarefas originalmente de *data mining* para explorar dados educacionais. A classificação é uma de suas tarefas e para obterem-se resultados precisos ao invés de utilizar apenas um único classificador pode-se utilizar a técnica de comitê de classificadores, a qual combina diversos classificadores básicos que ao final geram saídas diferentes, sendo necessário combiná-las de alguma forma. Em comitê de classificadores existem diversas técnicas que variam na forma da construção dos algoritmos e a combinação das saídas, pode-se citar o método de *boosting,* e o algoritmo de variação *adaboost.M1*, o qual tem foco em instâncias difíceis de classificar, atribuindo pesos ponderados. Outro algoritmo é o *random subspace*, que por sua vez utiliza um subconjunto aleatório de características dos dados para melhorar a relação entre a instância e a característica. O objetivo dessa pesquisa é aplicar *educational data mining* em uma base de dados do Moodle da Universidade do Extremo Sul Catarinense de modalidade a distância, a fim de definir perfis de interação dos alunos, baseando-se nas três interações de Moore, como: aluno-ambiente, aluno-professor e aluno-aluno, a ponto de predizer quais alunos tem mais chances de aprovação e reprovação por meio dos modelos obtidos pelos algoritmos de comitê de classificação *adaboost.M1* e *random subspace*. A fim de identificar o modelo que obteve melhor desempenho, os resultados são comparados por meio de medidas de qualidade em classificação, para isso foram realizados 6 experimentos para verificar quais técnicas de pré-processamento interferem nos resultados das medidas utilizadas, tais como acurácia, coeficiente Kappa, *TP-Rate* e *F-Measure*. Pode-se observar que melhores resultados foram encontrados ao utilizar técnica de balanceamento de classes, podendo destacar os algoritmos *adaboost.M1* e *random subspace* utilizando o classificador *random forest* como base, que chegaram a percentuais de acurácia como 93,51% e 93,77%, respectivamente. O modelo final encontrado, após analises, foi do algoritmo *random subspace* utilizando *random forest,* que alcançou em *TP-Rate*, 0,975 para classe “A” e 0,904 para “R”, mostrando resultados adequados para o objetivo proposto.

Palavras-chave: Educational Data Mining. Comitê de Classificadores. Adaboost.M1. Random Subspace.

ABSTRACT

The search for patterns of behavior, habits and choices occur from the beginning of human life. At present, society is supplied with information, which is found in its raw form, in data, and it is necessary to understand them. The knowledge discovery in database, is a process consisting of steps to find valid patterns and models. The data mining stage is responsible for exploring large amounts of data to find the models, with increasing demand for patterns and profiles of students in virtual learning environments, emerged educational data mining, to which adapts data mining tasks originally to explore educational data. Classification is one of its tasks and to obtain accurate results instead of using only a single classifier can be used the classifier committee technique, which combines several basic classifiers that in the end generate different outputs, in some way. In the committee of classifiers there are several techniques that vary in the form of the construction of the algorithms and the combination of the outputs, we can mention the boosting method, and the algorithm of variation adaboost.M1, which has focus on instances hard to classify, assigning weighted weights. Another algorithm is random subspace, which in turn uses a random subset of data characteristics to improve the relationship between the instance and the characteristic. The objective of this research is to apply educational data mining in an online educational platform database, Moodle, of the Universidade do Extremo Sul Catarinense, in order to define students profile interaction, based on the three interactions of Moore, such as: student- environment, student-teacher and student-student, to the point of predicting which students are more likely to approve and disapprove through the models obtained by adaboost.M1 and random subspace classification committee algorithms. In order to identify the model that had better performance, the results are compared by means of quality measures in classification, for this were carried out 6 experiments to verify which techniques of pre-processing interfere in the results of the measures used, such as accuracy, coefficient Kappa, TP-Rate and F-Measure. It can be observed that better results were found when using class-balancing technique, and it was possible to point out the adaboost.M1 and random subspace algorithms using the random forest classifier, which reached 93.51% and 93.77% %, respectively. The final model found, after analysis, was random subspace algorithm using random forest, which reached in TP-Rate, 0.975 for class "A" and 0.904 for "R", showing appropriate results for the proposed objective.

Keywords: Educational Data Mining. Classification Committee. Adaboost.M1.

Random Subspace.

1 INTRODUÇÃO

Desde o início da vida humana as pessoas buscam por padrões de comportamento, hábitos e escolhas. A sociedade se abastece de informações, as quais são encontradas em sua forma bruta, em dados, e devido ao seu volume, que cresce exponencialmente, a interpretação deles se torna cada vez mais difícil, sendo necessário entendê-los (FRANK et al, 2017, tradução nossa).

A descoberta de conhecimento em base de dados, do inglês *Knowledge Discovery in Database* (*KDD*), é constituída por etapas para identificação de modelos válidos, tais como: pré-processamento, *data mining* e pós-processamento (GOLDSCHMIDT; PASSOS; BEZERRA, 2015).

Data Mining é a etapa de descoberta de padrões e modelos prévios de forma automatizada em bases de dados (ZAQUI; MEIRA, 2014, tradução nossa). Devido a necessidade de explorar quais fatores interferem na aprendizagem, juntamente com a expansão de cursos a distância, surgiu uma área conhecida como Mineração de Dados Educacionais, do inglês *Educational Data Mining* (*EDM*), que pode ser definida como um campo de pesquisa que visa desenvolver métodos para observar dados retirados de ambientes educacionais (BAKER; ISOTANI; CARVALHO, 2011).

Dentre as subáreas da *EDM* tem-se a predição, que compreende em criar modelos para deduzir características específicas dos dados por meio da análise dos aspectos encontrados (BAKER; ISOTANI; CARVALHO, 2011).

A predição consiste na classificação dos dados, expressada por uma variável alvo, em que no processo de *data mining* os registros são examinados, sendo que cada um deles contém informações sobre a variável alvo, aprendendo-se a relação entre os registros observados (LAROSE; LAROSE, 2014, tradução nossa).

Ao aplicar o modelo de classificação é necessário calcular o nível de precisão dos resultados apresentados. Espera-se que a previsão tenha uma taxa de erro baixa e uma diminuição na variância dos algoritmos. No entanto, ao utilizar certos classificadores, as medidas citadas anteriormente podem obter taxas maiores, afetando a precisão do classificador (LAROSE; LAROSE, 2015, tradução nossa).

A fim de se obter resultados mais precisos, ao invés de utilizar apenas um classificador, utiliza-se a combinação deles. Esse método é chamado de comitê de classificadores, meta classificadores ou *ensembles*, o qual divide os dados em partes menores e mais fáceis de aprender, cada classificador fica responsável por uma partição específica, e ao final é calculada a média ou votação das saídas de cada um (POLIKAR, 2006, tradução nossa).

O comitê de classificadores pode ser do tipo *boosting,* o qual atribui pesos a todos os exemplos de treinamento (KUMAR; STEINBACH; TAN, 2009). Dentre os algoritmos do tipo *boosting*, tem-se o *Adaboost.M1*, que refaz a reamostragem se o classificador base não puder lidar com as instâncias ponderadas. Outro método é o *Random Subspace,* que cria um conjunto de classificadores, em que cada um é treinado usando um subconjunto de características selecionadas aleatoriamente do espaço de recurso disponível (FRANK et al, 2017, tradução nossa).

A classificação de dados está presente em diversas pesquisas relacionadas à *EDM* na Educação a Distância (EaD), em busca de perfis de participação por meio de dados retirados de Ambientes Virtuais de Aprendizagem (AVA), como nos estudos de Gottardo, Kaestner e Noronha (2013), Morais e Fechine (2013) e Santana, Maciel e Rodrigues (2014).

Atualmente, a necessidade de inovação tecnológica e pedagógica é apontada como uma das dificuldades da EaD, incluindo-se neste contexto a análise de desempenho dos alunos e a procura por metodologias que favoreçam o aprendizado, aumente o interesse, e facilitem a adaptação, para que por consequência diminuem as taxas de evasão desta modalidade de ensino (ABED, 2016).

Segundo Moore e Kearsley (2007), para que a educação na modalidade a distância seja eficiente, é importante a facilidade de entendimento entre discentes e docentes por meio de plataformas virtuais, para isso definem-se três tipos de interações rotuladas como: aluno-conteúdo, que consiste no processo de compreensão sobre o tema; aluno-aluno, definidos por grupos virtuais e reais; e aluno-professor, em que acontece o processo de imediação para a compreensão dos conteúdos.

A proposta desta pesquisa é aplicar *EDM* em uma base de dados do ambiente virtual de aprendizagem (AVA) da UNESC, procurando definir os perfis de interação dos alunos baseando-se nos estudados por Moore, a ponto de predizer quais alunos têm mais chances de aprovação e de reprovação por meio dos modelos obtidos. Para isso, os dados são analisados por meio da tarefa de classificação pelos algoritmos de comitê de classificação do tipo *boosting*, *Adaboost.M1* e *Random Subspace*, comparando-os por meio de medidas de qualidade em classificação de dados a fim de identificar o modelo que apresenta melhores resultados.

1.1 OBJETIVO GERAL

Propor um modelo de predição, por meio de comitê de classificadores, utilizando perfis de interações dos alunos estudados por Moore, para identificar quais alunos tem chances de aprovação e reprovação.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos desta pesquisa consistem em:

a) esclarecer *Educational Data Mining*, comitê de classificadores, os algoritmos *Adaboost.M1* e *Random Subspace*;
b) aplicar perfis de interação de Moore em dados do ambiente virtual de aprendizagem de disciplina na modalidade à distância na UNESC;
c) empregar os algoritmos *Adaboost.M1* e *Random Subspace*;
d) investigar por meio de medidas de qualidade as predições geradas pelos algoritmos *Adaboost.M1* e *Random Subspace*.

1.3 JUSTIFICATIVA

Data mining destaca-se por ser capaz de transformar grandes volumes de dados em conhecimentos específicos para sociedade. Esses conhecimentos

adquiridos são utilizados para análise de mercado, produção, negócios, entre outras aplicações (HAN; KAMBER, 2001, tradução nossa).

Com o aumento da procura por padrões e perfis dos aprendizes em ambientes virtuais de aprendizagem, surgiu a *EDM*, onde algoritmos de *data mining* são adaptados para melhores resultados na busca por conhecimentos nas bases de dados educacionais (BAKER; ISOTANI; CARVALHO, 2011).

Diversas técnicas são utilizadas em *EDM*, e uma delas é a predição, que auxilia, por exemplo, na identificação das vantagens de se utilizar estratégias educacionais em um determinado grupo de estudantes, tendo como uma de suas vantagens gerar um modelo utilizando parte dos dados podendo aplica-lo posteriormente em dados mais extensos, dessa forma, pode-se analisar e estimar benefícios educacionais sobre técnicas antes mesmo de aplica-las (BAKER; ISOTANI; CARVALHO, 2011).

A predição está presente em diversas pesquisas relacionadas à EaD, em busca de perfis de participação por meio de dados retirados de AVA, como o estudo de Santana, Maciel e Rodrigues (2017) que se basearam em técnicas de Moore para criar dimensões de aprendizado, por meio dos três perfis de interação, utilizando os algoritmos classificadores *J48* e *Random Forest*. Rabelo et al. (2017) empregaram algoritmos classificadores de árvores de decisão *ID3* e *J48* na predição de desempenho.

O comitê de classificadores tem como estratégia fazer com que vários classificadores treinem uma parte dos dados para combinar suas saídas, a fim de melhorar a predição resultante (POLIKAR, 2006, tradução nossa). Comparado com determinados algoritmos de classificação, o comitê de classificadores consegue reduzir a taxa de erro da predição e diminuir a variância, um problema que certos classificadores instáveis como os de árvores de decisão e de redes neurais artificiais apresentam (LAROSE; LAROSE, 2015, tradução nossa).

O algoritmo *Adaboost.M1*, uma variação do tipo *boosting*, popular versão adaptada de *Adaboost*[1], tem foco em instâncias mal classificadas, aumentando seus pesos, além disso garante que o erro de treinamento seja inferior a um meio,

[1] Algoritmo de comitê de classificação do tipo *boosting*, desenvolvido por Yoav Freund e Robert Schapire (ZHANG; MA, 2012, tradução nossa).

diminuindo à medida que o conjunto cresce (ZHANG; MA, 2012, tradução nossa). O método *Random Subspace* utiliza subconjuntos aleatórios, os quais são facilmente treinados e melhoram a relação de instância e característica, pois são divididos em menores, levando em consideração que a substituição de um conjunto de classificador não prejudica a precisão (KUNCHEVA et al., 2010, tradução nossa).

Segundo Baker, Isotani e Carvalho (2011) a utilização de sistemas computacionais no gerenciamento de cursos promovem o aumento da aplicação de *EDM*, além disso, a educação a distância obteve um crescimento nos últimos anos no Brasil. Em 2017, segundo o censo efetuado pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP), houve um aumento no número de ingressantes na modalidade a distância equivalente a 27,3% entre 2016 e 2017, já em relação as matrículas obteve um aumento de 17,6% no mesmo período, representando 46,8% do total. Ainda em 2017, a educação na modalidade a distância teve uma oscilação de 19,7% para 21,0% no percentual de concluintes, após queda no ano anterior (INEP, 2017).

Conforme censo realizado em 2016, pela Associação Brasileira de Educação a Distância (ABED), um dos maiores desafios é a inovação nas abordagens pedagógicas para que o discente tenha suas necessidades supridas no meio virtual. No ano de 2017 o atendimento ágil às necessidades dos alunos, tecnologia e metodologia confiável e inovadora foram apontados como itens diretamente relacionados a qualidade na EaD (ABED, 2017). Para isso, são necessários investimentos nesse campo, seguido da análise de desempenho dos alunos em busca de melhorias nas metodologias abordadas durante o curso. Assim, pode-se incentivar os alunos e diminuir as taxas de evasão (ABED, 2016).

1.4 ESTRUTURA DO TRABALHO

Esta pesquisa é dividida em seis capítulos, em que o primeiro capítulo contém uma introdução, seus objetivos gerais, específicos e uma justificativa.

O segundo capítulo aborda a descoberta de conhecimento em bases de dados e seus processos, aprofundando-se no processo de *data mining* e na tarefa de classificação.

O terceiro capítulo apresenta o funcionamento do comitê de classificadores, seu método *boosting*, aprofundando-se em sua variação para problemas de classificação multi-classe, o algoritmo *Adaboost.M1* e o método *Random Subspace.* Serão abordados métodos de qualidade para avaliar classificadores.

No quarto capítulo serão tratados conceitos de EAD, ambientes virtuais de aprendizagem, as três interações observadas por Moore, apresentam conceitos e técnicas em *EDM* e traz trabalhos que tiveram relação com o uso de dados educacionais e o uso dos algoritmos *Adaboost.M1* e *Random Subspace*.

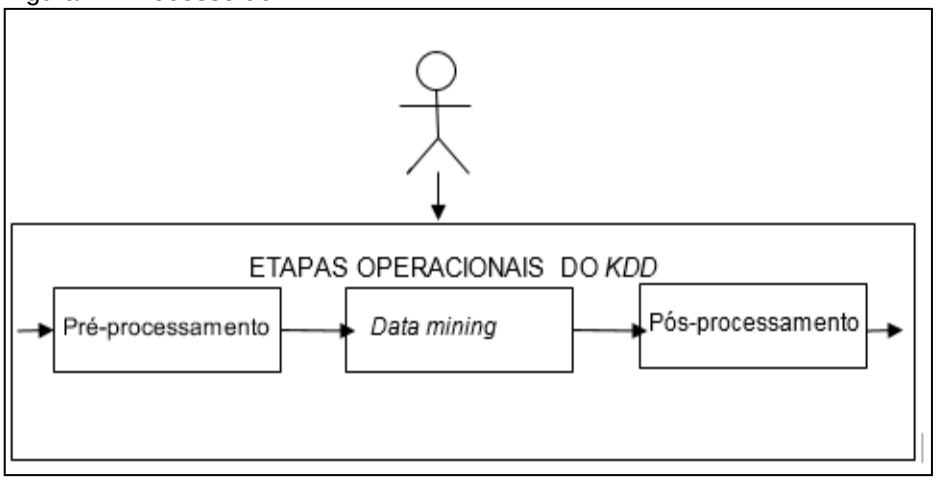
No quinto capítulo é descrito o trabalho desenvolvido nesta pesquisa, trazendo breves explicações sobre o banco de dados utilizado, as técnicas de pré-processamento empregadas, a execução do *data mining*, os experimentos realizados, resultados obtidos e discussões dos mesmos.

No sexto capítulo, por fim, é apresentada a conclusão do trabalho e sugestões para trabalhos futuros.

2 DESCOBERTA DE CONHECIMENTO EM BASES DE DADOS

A Descoberta de Conhecimento em Bases de Dados (DCBD), do inglês *Knowledge Discovery in Databases* (*KDD*), é um processo de transformação de dados em modelos válidos, compreensíveis e úteis para determinada aplicação, realizado por meio de métodos de inferência ou busca (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996, tradução nossa). O processo de *KDD* é dividido basicamente em três partes, tais como: pré-processamento, *data mining* e pós-processamento (figura 1) (TAN; STEINBACH; KUMAR, 2009).

Figura 1 - Processo de *KDD*.



Fonte: Adaptado de Goldschmidt, Passos e Bezerra (2015).

O pré-processamento compreende a etapa de preparação e transformação dos dados de entrada em um formato adequado para ser analisado. Os dados podem ser encontrados em diversas formas, para isso, são utilizadas técnicas como a normalização para unir dados de diversas fontes e a limpeza para tratar dados ausentes, incorretos, discrepantes e inconsistentes (TAN; STEINBACH; KUMAR, 2009).

O processo de *data mining* é a etapa responsável pela descoberta de padrões e tendências nos dados, utiliza tarefas apropriadas para reconhecimento do modelo, baseando-se na estatística, matemática e aprendizado de máquina (LAROSE; LAROSE, 2015, tradução nossa).

Na etapa de pós-processamento são analisados e compreendidos os resultados dos modelos obtidos no processo de *data mining,* para isso empregam-se

técnicas para melhorar a compreensão sobre os padrões, tais como: simplificação de modelo de conhecimento, que elimina informações não relevantes para tornar os padrões menos complexos; transformação do modelo de conhecimento, que modifica a forma de representação, como por exemplo, em diagrama (GOLDSCHMIDT; PASSOS; BEZERRA, 2015); método de apresentação dos resultados, que favorece a visualização e interpretação do usuário sobre os padrões obtidos (KANTARDZIC, 2011, tradução nossa).

2.1 DATA MINING

Data mining é explorada na comunidade científica, bem como na área de negócios. Atualmente, diversos setores aderiram a esse processo, a indústria, por exemplo, utiliza com o intuito de compreender clientes e analisar estratégias para mantê-los; setores de varejo analisam histórico de vendas, com objetivo de encontrar padrões de consumo; setores bancários aplicam para previsões de fraudes, análises de riscos e tendências (KANTARDZIC, 2011, tradução nossa). As áreas que aplicam o processo de *data mining* exploram grandes quantidades de dados para novas descobertas (TAN; STEINBACH; KUMAR, 2009).

Existem desafios acerca de *data mining*, como análise de escala para grandes bancos de dados, que exploram alternativas para manipular conjuntos de dados que não podem ser carregados na memória principal; análise de dimensionamento, que investigam formas para realizar análises estatísticas em conjuntos de dados com muitas variáveis; pesquisas automatizadas, as quais atribuem tarefas de geração de hipóteses a algoritmos específicos; e busca por padrões compreensíveis e úteis, que extraem modelos de fácil compreensão, por meio de metodologias com o intuito de buscar precisão nos resultados e utilidade ao usuário (FAYYAD; UTHURUSAMY, 2002, tradução nossa).

Os modelos encontrados são utilizados para fins específicos dependendo da área de aplicação, em *data mining* as modelagens obtidas podem ser divididas em duas categorias, preditiva e descritiva (TAN; STEINBACH; KUMAR, 2009). A predição faz a indução nos dados explorados para prever valores não conhecidos (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996, tradução nossa), já a modelagem

descritiva deriva padrões por meio dos dados de entrada (TAN; STEINBACH; KUMAR, 2009).

As tarefas de *data mining* utilizam algoritmos de aprendizado de máquina para obter modelos preditivos ou descritivos, como por exemplo: classificação e agrupamento são tarefas preditivas; na descritiva podem ser citadas associação, descoberta de sequência, sumarização, detecção de desvios e regressão (KANTARDZIC, 2011, tradução nossa).

A associação investiga dados frequentes e presentes simultaneamente em transações de banco de dados (ZAQUI, 2000, tradução nossa). Esse método normalmente é utilizado em análises de compras em mercados, por exemplo, quando um cliente adquire determinado produto também compra outro, o objetivo é analisar quais produtos estão presentes ao mesmo tempo (REFAAT, 2007, tradução nossa).

A descoberta de sequência é semelhante a tarefa de associação, contudo é responsável por encontrar a relação entre as associações (GOLDSCHMIDT; PASSOS; BEZERRA, 2015). Exemplificando a aplicação, em uma análise de compras em mercado, a descoberta de sequência deve informar qual produto potencializa a compra de outro (MADNI; ANWAR; SHAH, 2017, tradução nossa).

Análise de grupo ou agrupamento é uma tarefa em que grupos são formados por partilharem elementos comuns entre si, porém são heterogêneos entre os grupos (HAN; KAMBER; PEI, 2012, tradução nossa). As aplicações de agrupamento estão presentes na segmentação demográfica, astronômica e na área de mercado (ZAQUI, 2000, tradução nossa).

A sumarização envolve técnicas para encontrar uma descrição compacta de um subconjunto de dados, normalmente utilizada para tabular os desvios padrões e média, visualização de dados e geração de relatórios automatizados (CHANDOLA; KUMAR, 2006, tradução nossa). Na área de mercado, por exemplo, utiliza para identificação de características comuns entre clientes que assinam uma determinada revista, como faixa etária, área de atuação, entre outros (GOLDSCHMIDT; PASSOS; BEZERRA, 2015).

A detecção de desvios ou anomalias identifica características diferentes do restante dos dados, são conhecidos também por *outliers,* muito utilizados para

detecção de fraudes em cartões de crédito (HAN; KAMBER; PEI, 2012, tradução nossa). O objetivo da tarefa é descobrir desvios verdadeiros e não rotular objetos normais como anômalos, ou seja, deve conter altas taxas de detecções verdadeiras (TAN; STEINBACH; KUMAR, 2009).

A tarefa de regressão tem como objetivo aprender uma função que examina um conjunto de dados e atribui a um rótulo de classe pré-determinado, contudo é restrita apenas a atributos contínuos[2] (TAN; STEINBACH; KUMAR, 2009). As aplicações de regressão podem ser para fins de diagnósticos, como por exemplo, definir a probabilidade de sobrevivência de um paciente por meio de resultados encontrados, bem como prever a procura por um determinado produto, entre outros (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996, tradução nossa).

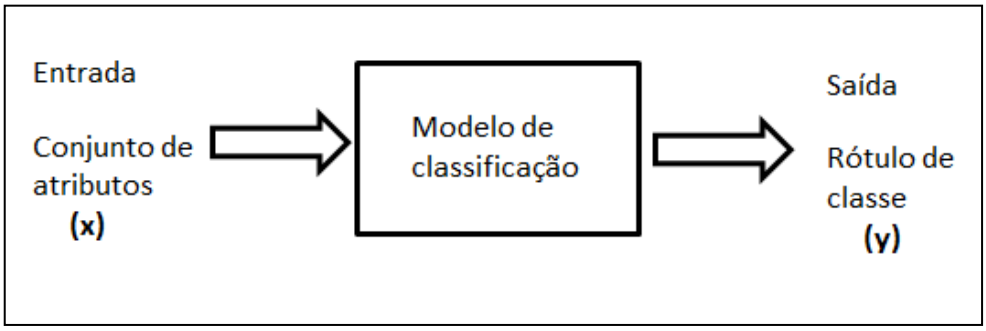
A tarefa de classificação é semelhante à regressão, consiste em mapear um conjunto de objetos para uma determinada classe por meio do aprendizado de uma função alvo, no entanto trabalha com atributos discretos[3]. Pode ser aplicada na área de *marketing* ou para pressupor a partir de resultados de exames médicos se o paciente possui ou não determinada doença (TAN; STEINBACH; KUMAR, 2009).

2.2 CLASSIFICAÇÃO

Os dados de entrada na tarefa de classificação são um conjunto de instâncias, cada uma delas é caracterizada por uma dupla *(x,y)* , sendo *x* o conjunto de atributos, que podem receber registros tanto discretos quanto contínuos, e *y* o rótulo da classe, delimitado para receber apenas registros discretos. Na tarefa de classificação é realizado o mapeamento de um conjunto de atributos *x* ao seu rótulo correspondente *y* (figura 2) (TAN; STEINBACH; KUMAR, 2009). Os classificadores podem resolver problemas binários, assumindo um número de classes igual a dois, representado por $K = 2$, ou multi-classe, que assumem um número de classes superiores a dois valores, simbolizado por $K > 2$ (DIETTERICH; BAKIRI, 1995, tradução nossa).

Figura 2 - Tarefa de classificação.

[2] Atributos contínuos são constituídos por números reais (TAN; STEINBACH; KUMAR, 2009).
[3] Atributos discretos possui um conjunto de valores finitos (TAN; STEINBACH; KUMAR, 2009).



Fonte: Adaptado de Tan, Steinbach e Kumar (2009).

A classificação pode ser dividida em duas etapas, a de treinamento e a de testes. No treinamento realiza-se o aprendizado por meio dos dados de entrada, sendo definido o seu modelo, para isso são utilizadas técnicas apropriadas por meio de algoritmos de aprendizagem capazes de se adaptarem aos dados, prevendo corretamente os rótulos de classes não vistos anteriormente. Na segunda etapa são examinados os novos registros e atribuídos rótulos aos dados a partir do modelo encontrado na fase de treinamento (HAN; KAMBER; PEI, 2012, tradução nossa).

O modelo de classificação é avaliado por meio dos registros classificados corretamente na etapa de testes, considerando que os algoritmos selecionados buscam por modelos que alcançam maior nível de precisão (TAN; STEINBACH; KUMAR, 2009).

Com a expansão em conhecimento de modelos de classificação, surgiu como um avanço em reconhecimento de padrões, a combinação de classificadores (KUNCHEVA, 2004, tradução nossa), que comparado aos individuais têm mostrado resultados superiores em diversos cenários da inteligência computacional e aprendizado de máquina (DIETTERICH, 2000, tradução nossa).

3 COMITÊ DE CLASSIFICADORES

Comitê de classificadores, metaclassificadores ou *ensembles* são conjuntos de classificadores básicos, representado por *C*, construídos a partir de dados de treinamento, *D* (TAN; STEINBACH; KUMAR, 2009), que produzem diversas saídas, as quais são combinadas de alguma forma, representada por *C*,* para classificar novos exemplos (DIETTERICH, 2000, tradução nossa). O pseudocódigo da figura 3 mostra o funcionamento do comitê de classificadores.

Figura 3 - Pseudocódigo de comitê de classificadores.

Suponha que D, os dados de treinamento originais, k denote o número de classificadores básicos e T sejam os dados de teste.
para até k **faça:**
    Crie um conjunto de treinamento D_i a partir de D.
    Construa um classificador básico C_i a partir de 1 D_i.
fim para
    Para cada registro de teste $x \in T$ **faça:**
        $C*(x) = Voto\,(C1(x), c2(x), \ldots, Ck(x))$
fim para

Fonte: Adaptado de Tan, Steinbach e Kumar (2009).

Segundo Dietterich (2000, tradução nossa), os comitês de classificadores têm obtido resultados superiores comparados a um único classificador pelas seguintes razões**:**

a) **estatística**: alguns classificadores podem ter desempenho de generalização semelhantes ou diferentes, no entanto caso o conjunto de testes utilizado para definir a generalização não seja suficiente, combinar saídas de diversos classificadores reduz o risco de exemplos serem atribuídos a classes incorretas (POLIKAR, 2006, tradução nossa);
b) **computacional**: alguns classificadores são sensíveis ao treinamento, combinando classificadores é possível suavizar a sensibilidade para melhorar as previsões obtidas;
c) **representacional**: na maioria dos casos não é possível representar a função do sistema encontrado, por meio de fusão ou ponderação das

> hipóteses, é possível ampliar a função de representação (POLIKAR, 2006, tradução nossa).

Os classificadores têm suas saídas combinadas com o propósito de minimizar as decisões incorretas e amplificar as corretas. Existem diversas formas para construir combinações, pode-se destacar a votação majoritária e a votação majoritária ponderada (POLIKAR, 2006, tradução nossa). Na votação majoritária, o resultado final é baseado nas saídas que receberam o maior número de votos, tendo em vista que todos possuem o mesmo peso (ZHANG; MA, 2012, tradução nossa). No método de votação majoritária ponderada, o algoritmo produz saídas com pesos diferentes de acordo com a precisão dos classificadores obtida no conjunto de treinamento, assim o método se concentra em saídas com pesos maiores para alcançar o resultado final (POLIKAR, 2006, tradução nossa).

Na construção de um comitê de classificadores são selecionados métodos para produzir um conjunto diversificado (POLIKAR, 2006, tradução nossa), o objetivo da diversidade é criar vários classificadores cujos limites de decisão sejam diferentes entre si. Assim os exemplos que não foram classificados corretamente apresentam seu erro reduzido ao realizar combinações estratégicas (KUNCHEVA, 2004, tradução nossa).

Diversos métodos de combinação de classificadores foram desenvolvidos, entre eles destacam-se *boosting, bagging* e *random subspace.* No método de *bagging* os conjuntos de treinamento são obtidos aleatoriamente a partir de um conjunto de dados cuja distribuição de probabilidade é uniforme. O método treina por meio de um classificador base as amostras de *bootstrap*, ao final a instância é atribuída a classe que receber o maior número de votos, combinando as saídas por voto majoritário (TAN; STEINBACH; KUMAR, 2009). Em *random subspace,* os classificadores são criados em subespaços aleatórios a partir dos dados disponíveis, posteriormente, apenas os classificadores com erro de classificação igual a zero são combinados por voto majoritário (KUNCHEVA, 2004, tradução nossa). No método de *boosting* os classificadores são construídos por meio de versões ponderadas do conjunto de treinamento, inicialmente todas as instâncias possuem pesos iguais, após a primeira rodada, os pesos são reajustados por meio do desempenho do

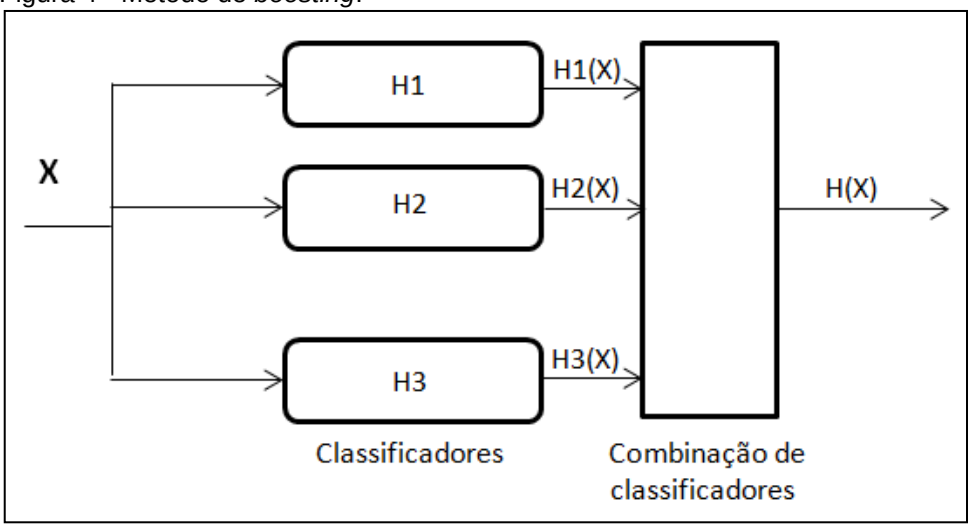
classificador. A combinação das saídas pode ser realizada por votação majoritária simples ou ponderada (ZHANG; MA, 2012, tradução nossa).

3.1 MÉTODO DE BOOSTING

Boosting é um método geral de aprendizagem de máquina que constrói um classificador preditivo de alta taxa de precisão, a partir da combinação de vários algoritmos de aprendizagem fracos (FREUND; SCHAPIRE, 1996, tradução nossa). O algoritmo de aprendizagem considerado fraco ou *weaklearner* é capaz de gerar classificador cuja taxa de erro é um pouco melhor do que a adivinhação aleatória (ZHANG; MA, 2012, tradução nossa). Em *boosting* esses classificadores são submetidos sequencialmente e repetidamente em versões dos dados modificados (HASTIE; TIBSHIRANI; FRIEDMAN, 2009, tradução nossa). O algoritmo que realiza a combinação é capaz de produzir classificadores fortes a partir de algoritmos fracos, em que sua taxa de erro é muito pequena (ZHANG; MA, 2012, tradução nossa).

A estratégia é aprender classificadores fracos e combinar suas saídas de alguma forma para transformá-los em um classificador forte (figura 4), em vez de aprender apenas um único classificador complexo (FREUD, 1990, tradução nossa).

Figura 4 - Método de *boosting*.



Fonte: Adaptado de Zhang e Ma (2012).

O método cria três classificadores fracos, um de cada vez. Dado um conjunto de dados *x*, primeiro é selecionado um subconjunto aleatório dos dados disponíveis, para treinar o primeiro classificador, *H1* (DUDA; HART; STORK, 2001, tradução nossa). O segundo classificador, *H2*, é treinado em um conjunto de dados diferente dos dados originais, e exatamente metade do conjunto deve ser classificado corretamente por *H1* e metade classificado incorretamente. O terceiro classificador, *H3*, é treinado com instâncias em que *H1* e *H2* discordaram (ZHANG; MA, 2012, tradução nossa).

Os três classificadores são combinados considerando os padrões que *H1* e *H2* concordaram, e nos casos em que discordaram, utiliza-se os padrões obtidos por *H3*, a hipótese final é calculada por meio do voto majoritário (DUDA; HART; STORK, 2001, tradução nossa). O pseudocódigo do método *boosting* é representado pela figura 5.

Figura 5 - Pseudocódigo de *Boosting*.

Entrada: Dado o conjunto de dados $Z = \{z_1, z_2, \ldots, z_n\}$, com $z_i = x_i, y_i$, quando $x_i \in X$ e $y_i \in \{-1, +1\}$.
Saída: Um classificador $H: X \to \{-1, +1\}$.
1: Selecione aleatoriamente, sem substituição, $L_1 < N$ amostras de Z para obter z_1^*.
2: Rode o *WK* em z_1^*, produzindo o classificador H_1.
3: Selecione $L_2 < N$ amostras de Z, com metade das amostras erroneamente classificadas por H_1, para obter z_2^*.
4: Rode o *WK* em z_2^*, produzindo o classificador H_2.
5: Selecione todas as amostras de Z, nas quais H_1 e H_2 descordaram, produzindo z_3^*.
6: Rode o *WK* em z_3^*, produzindo o classificador H_3.
7: Produzir o classificador final com um voto majoritário:

$$H(x) = sign(\sum_{b=1}^{3} H_b(x))$$

Fonte: Adaptado de Zhang e Ma (2012).

Adaptações de *boosting*, como *adaboost*, foram desenvolvidas para problemas binários, e manipulam os dados de treinamento implícito re-ponderando, para que os algoritmos fracos encontrem resultados novos a cada iteração (FREUND; SCHAPIRE, 2012, tradução nossa). O método gera distribuições que concentram nos elementos mais difíceis de serem classificados, devido à atribuição de pesos, obrigando os algoritmos fracos a terem melhor desempenho nas amostras difíceis de classificar. Para obtenção do resultado são combinados os padrões

encontrados na etapa de treinamento, em que a variância é menor do que a produzida por um único algoritmo fraco (FREUND; SCHAPIRE, 1996, tradução nossa).

Outras variações de algoritmos *boosting* foram desenvolvidas*,* as implementações são modificadas em termos de como são utilizados os pesos e a combinação das saídas dos classificadores (TAN; STEINBACH; KUMAR, 2009). *Adaboost.M1* é um exemplo de variações para problemas binários e múltiplas classes proposto por Freud e Schapire em 1996 (WITTEN; FRANK, 2005, tradução nossa).

3.1.1 Adaboost.M1

Adaboost.M1 provém de *Adaptive Boosting,* em que *M* significa multi-classe, e o 1 é relacionado a primeira extensão do algoritmo (FREUND; SCHAPIRE, 2012, tradução nossa). Como no método de *boosting,* o algoritmo *weaklearn* gera hipóteses *h*, atribuindo cada instância *x* a um determinado rótulo *y*, tendo *h: X -> Y* (FREUND; SCHAPIRE, 2012, tradução nossa). Cada rótulo *y* pertence ao conjunto *Y* de rótulos disponíveis, que podem assumir cardinalidade maior que 2, nos casos de múltiplas classes (FREUND; SCHAPIRE, 1996, tradução nossa).

A distribuição *D* atribui um peso a cada instância *x* de treinamento, inicializada com pesos uniformes para que no primeiro conjunto de treinamento todas as instâncias tenham a mesma chance de serem sorteadas (ZHANG; MA, 2012, tradução nossa). Na rodada *t*, o algoritmo *weaklearn* é chamado com uma distribuição em um conjunto de treinamento, como resposta o algoritmo gera classificadores que minimizem o erro de treinamento. Satisfazendo o erro de treinamento menor ou igual a ½, são reajustados os pesos das instâncias, aumentando os pesos dos exemplos mal classificados. A hipótese final *H*, é calculada por meio da votação majoritária ponderada com as previsões geradas (figura 6) (FREUND; SCHAPIRE, 2012, tradução nossa).

Figura 6 - Pseudocódigo de *adaboost.M1*.

Dado:= $(x_i,), \ldots, (x_m, y_m)$ onde $x_i \in X, y_i \in Y$.
Inicialize: $D_i(i) = \frac{1}{m}$ para $i = 1, \ldots, m$.
para $t = 1, \ldots, T$ **faça:**
  Treine o aprendiz fraco usando distribuição D_i.
  Obtenha hipótese fraca $h_t = X \rightarrow Y$.
  Alvo: selecione h_t para minimizar o erro:
    $\epsilon_t = \boldsymbol{Pr} \sim D_t [h_t(x_i \neq y_i)]$
  Se $\epsilon_t \geq \frac{1}{2}$, então defina $T = t - 1$ e saia do loop.
  Escolha $\alpha_t = \frac{1}{2} ln\left(\frac{1-\epsilon_i}{\epsilon_i}\right)$
  Atualize, para $i = 1, \ldots, m$ faça:
    $D_{t+1}(i) = \frac{D_t(i)}{Z_t} x \begin{cases} e^{-\alpha_t} if\ h_t(x_i) = y_i \\ e^{\alpha_t} if\ h_t(x_i) \neq y_i \end{cases}$
  Onde Z_t é um fator de normalização(escolhido para que D_{t+1} seja uma distribuição).

Fim para:
  Gere a hipótese final:
    $H(x) = argmax_{y \in Y} \sum_{t=1}^{T} \alpha_t 1\{h_t(x) = y\}$

Fonte: Adaptado de Freud e Schapire (2012).

O objetivo do algoritmo *weaklearn* é encontrar hipóteses que minimizem o erro de treinamento, a hipótese encontrada deve classificar corretamente grande parte do conjunto de treinamento. Para isso é calculada a taxa de erro $\boldsymbol{\epsilon}$, representada pela fórmula (1) (FREUND; SCHAPIRE, 2012, tradução nossa).

$$\epsilon_t = \boldsymbol{Pr} \sim D_t [h_t(x_i \neq y_i)] \quad (1)$$

A taxa de erro por sua vez, calcula a probabilidade em relação à distribuição D_t, fornecida pelo *weaklearn,* de a instância não pertencer ao rótulo. O resultado obtido por ϵ_t é submetido a uma condição, dada por (ZHANG; MA, 2012, tradução nossa):

$$\epsilon_t \geq \frac{1}{2} \quad (2)$$

A condição somente é satisfeita caso a taxa de erro de treinamento seja menor ou igual a ½, assim como em problemas de classificação binária. Caso seja superior, o processo é abortado e o voto majoritário ponderado é calculado com as hipóteses geradas até o momento (ZHANG; MA, 2012, tradução nossa).

Problemas de classificação com múltiplas classes, em que o número de classes é superior a dois, $K > 2$, adotar algoritmos apenas ligeiramente melhor que

adivinhação aleatória não é suficiente, já que a probabilidade de uma predição aleatória correta é de ½. Exemplificando, se *K > 10* classes, adivinhar aleatoriamente resultaria em uma precisão de 10%, inferior ao requisito de 50% (FREUND; SCHAPIRE, 2012, tradução nossa).

Caso a taxa de erro seja satisfeita, é calculada a taxa de importância α_t do classificador formado, permitindo valores entre 0 e 1, representada pela fórmula 3 (EIBL; PFEIFFER, 2002, tradução nossa).

$$\alpha_t = \frac{1}{2} ln\left(\frac{1-\epsilon_i}{\epsilon_i}\right) \tag{3}$$

Ao calcular a taxa de importância é possível observar que classificadores com baixa taxa de erro, consequentemente terão maiores taxas de importâncias atribuídas a eles (EIBL; PFEIFFER, 2002, tradução nossa).

Em seguida é calculada a distribuição dos pesos, D_t, utilizando a seguinte fórmula 4 (ZHANG; MA, 2012, tradução nossa).

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t}\, x \begin{cases} e^{-a_t}\, if\ h_t(x_i) = y_i \\ e^{a_t}\, if\ h_t(x_i) \neq y_i \end{cases} \tag{4}$$

Sendo Z_t um fator de normalização, utilizado para que a soma dos pesos continue igual a 1, é representado por (ZHANG; MA, 2012, tradução nossa):

$$Z_t = D_{t+1}(i) = 1 \tag{5}$$

Nessa etapa, os exemplos classificados incorretamente têm seus pesos incrementados, e os classificados corretamente têm seus pesos reduzidos, fazendo com que o algoritmo se concentre em instâncias mais difíceis de classificar nas próximas rodadas (FREUND; SCHAPIRE, 1996, tradução nossa).

A hipótese final *H*, é combinada por votações, ponderadas por meio da taxa de importância calculada anteriormente, em que quanto maior a importância atribuída à hipótese, melhor é seu desempenho. O cálculo para geração da hipótese final é dado pela fórmula 6 (EIBL; PFEIFFER, 2002, tradução nossa).

$$H(x) = argmax_{y \in Y} \sum_{t=1}^{T} \alpha_t \, 1\{h_t(x) = y\} \quad (6)$$

Adaboost.M1 se torna inadequado para problemas com múltiplas classes ao utilizar classificadores de base fraca, cuja precisão é pouco melhor que adivinhação aleatória. O método funciona de forma superior ao utilizar classificadores de base forte, cuja precisão superior a ½, inclusive em distribuições difíceis de classificar (FREUND; SCHAPIRE, 2012, tradução nossa).

3.2 RANDOM SUBSPACE

Random Subspace (*RS)* ou subespaços aleatórios é um método proposto por Ho em 1998, em que classificadores são construídos em um subconjunto de características dos dados disponíveis de tamanho pré-definido, amostrados aleatoriamente (KUNCHEVA et al., 2010, tradução nossa).

Para construir um conjunto *RS*, são coletadas *L* amostras de tamanho *M*, extraídas de uma distribuição uniforme *X*, sem substituição (KUNCHEVA et al*.,* 2010, tradução nossa), ou seja, o objeto selecionado para amostra é removido do conjunto de dados original (TAN; STEINBACH; KUMAR, 2009).

Cada subconjunto de características representa um subespaço da distribuição *X* de tamanho *M* (KUNCHEVA et al., 2010, tradução nossa)*.* O pseudocódigo do método *Random Subspace* é representado pela figura 7 (SKURICHINA; DUIN, 2002, tradução nossa):

Figura 7 - pseudocódigo do método *Random Subspace.*

Repita para $b = 1,2,\ldots,B$ **faça:**
Selecione uma r-dimensional subespaço aleatório X^b do p-dimensional original do espaço X.
Construa um classificador $C^b(x)$ (com uma limite de decisão $C^b(x) = 0$ em X^b)
Fim para:
Combine classificadores $C^b(x)$, $b = 1,2,\ldots,B$, por voto majoritário a uma regra de decisão final

$$\beta(x) = argmax_{y \in \{-1,1\}} \sum_b^{\delta} sgn(C^b(x)), y$$

Onde $\delta_{i,j}$ é o símbolo Kronecker, e $y \in \{-1,1\}$ é uma decisão (rótulo de classe) do classificador.

Fonte: Adaptado de Skuricha e Duin (2002).

Seja *X* o conjunto de treinamento, onde cada objeto de treinamento $X_i = (i = 1,\ldots,n)$ é um vetor *p*-dimensional representado por $X_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$, dito por *p* características, seleciona-se aleatoriamente características *r* < *p* do conjunto *p*-dimensional *X*, obtendo-se um vetor *r*-dimensional do espaço de recursos do conjunto *p*-dimensional (SKURICHINA; DUIN, 2002, tradução nossa).

O conjunto de treinamento modificado é representado por $\tilde{X}_n^b = (\tilde{X}_1^b, \tilde{X}_2^b, \ldots, \tilde{X}_n^b)$, constituído pelo subespaço *r*-dimensional de treinamento $\tilde{X}_1^b = (X_{i1}^b, X_{i2}^b, \ldots, X_{in}^b)$ $(i = 1,\ldots,n)$ em que *r* características $X_{i1}^b (j = 1,\ldots,r)$ são selecionadas aleatoriamente de *p* características $X_{i1}^b (j = 1,\ldots,p)$ do conjunto de treinamento (SKURICHINA; DUIN, 2002, tradução nossa). As saídas dos classificadores são combinadas por meio do voto majoritário (KUNCHEVA, 2004, tradução nossa), representada pela função 7 (SKURICHINA; DUIN, 2002, tradução nossa):

$$\beta(x) = argmax_{y \in \{-1,1\}} \sum_b^{\delta} sgn(C^b(x)), y \quad (7)$$

Alguns subconjuntos aleatórios selecionados podem ter baixa capacidade de separação de classes, sendo uma desvantagem para o método *random subspace*, tendo em vista que um classificador fraco pode afetar o resultado final do conjunto (MERT; KILIÇ; BILGILI, 2016, tradução nossa).

Em dados cuja dimensionalidade é alta, *RS* transforma o conjunto em partes menores podendo resolver esse tipo de problema (XUE; DU; DU, 2013, tradução nossa), reduzindo ruído e eliminando características irrelevantes do modelo (TAN; STEINBACH; KUMAR, 2009). O *overfitting*[4] também pode ser reduzido ao diminuir o número de características por classificador (PLUMPTON, 2011, tradução nossa), assim como os diferentes subconjuntos aleatórios podem melhorar a diversidade dos classificadores de base do conjunto, consequentemente o desempenho do classificador é melhorado (APOLLONI; VALENTINI; BRAGA, 2006, tradução nossa).

Os classificadores buscam modelos que atinjam melhor precisão ou menor taxa de erro (como citado no subcapítulo 2.1.1), o desempenho ou a qualidade destes classificadores podem ser medidos de diversas maneiras aplicadas no conjunto de testes (TAN; STEINBACH; KUMAR, 2009). O subcapítulo 2.4 aborda medidas de qualidade aplicadas em classificação.

3.3 MEDIDAS DE QUALIDADE PARA CLASSIFICAÇÃO

A qualidade de um classificador pode ser avaliada em termos da sua taxa de erro ou precisão aplicado em um conjunto de testes (FREUND; SCHAPIRE, 2012, tradução nossa), em que os rótulos de classes são desconhecidos (TAN; STEINBACH; KUMAR, 2009). A taxa de erro indica a proporção em que instâncias são classificadas incorretamente, representada por (WITTEN; FRANK, 2005, tradução nossa):

$$Taxa\ de\ erro\ = \frac{número\ de\ previsões\ erradas}{número\ total\ de\ previsões} \tag{8}$$

A precisão é equivalente à taxa de erro (TAN; STEINBACH; KUMAR, 2009), contudo indica a proporção em que instâncias são classificadas corretamente (FREUND; SCHAPIRE, 2012, tradução nossa), representada pela fórmula 9 (TAN; STEINBACH; KUMAR, 2009):

[4] *Overfitting* ocorre quando o classificador está excessivamente adaptado ao conjunto de treinamento.

$$Precisão = \frac{número\ de\ previsões\ corretas}{número\ total\ de\ previsões} \quad (9)$$

Ao analisar o desempenho de um classificador por meio de um conjunto de dados conhecidos, chamado de conjunto de treinamento, obtém-se o erro de generalização, que auxilia o algoritmo a selecionar modelos menos propícios a *overfitting* (TAN; STEINBACH; KUMAR, 2009).

Para melhor desempenho do classificador é importante que os dados do conjunto de testes não tenham sido utilizados para criar o classificador no conjunto de treinamento. Dessa forma, quando a quantidade de amostras é representativa, são selecionados um conjunto de dados para o treinamento, melhorando o erro de generalização e outras amostras para o conjunto de testes, diminuindo a taxa de erro. O problema ocorre quando a quantidade de dados disponíveis para os dois conjuntos é relativamente pequena (WITTEN; FRANK, 2005, tradução nossa).

As contagens de registros previstos pelo conjunto de testes podem ser tabuladas em uma matriz, chamada de matriz de confusão (TAN; STEINBACH; KUMAR, 2009). Em problemas de classificação binária existem quatro possíveis resultados: verdadeiro positivo (VP) e verdadeiro negativo (VN), que representam o número de registros classificados corretamente; falso positivo (FP) e falso negativo (FN), que representam o número de registros classificados incorretamente (SOKOLOVA; LAPALME, 2009, tradução nossa). A figura 8 ilustra a matriz de confusão.

Figura 8 - Matriz de confusão.

		Classe Prevista	
		Positiva	Negativa
Classe Real	Positiva	VP	FN
	Negativa	FP	VN

Fonte: Adaptado de Tan, Steinbach e Kumar (2009).

Os resultados obtidos por meio da matriz de confusão são bases para calcular algumas medidas de qualidade, em problemas de classificação binária as mais utilizadas são (SOKOLOVA; LAPALME, 2009, tradução nossa):

a) **acurácia:** mede eficiência geral de um classificador:

$$Acurácia = \frac{VP+VN}{VP+FN+FP+VN} \quad (10)$$

b) **precisão:** mede a porcentagem de instâncias classificadas como positiva estão realmente correta (HAN; KAMBER; PEI, 2012, tradução nossa):

$$Precisão = \frac{VP}{VP+FP} \quad (11)$$

c) **sensibilidade**: mede a capacidade do modelo em classificar instâncias corretamente (LAROSE; LAROSE, 2015, tradução nossa):

$$Sensibilidade = \frac{VP}{VP+FN} \quad (12)$$

d) ***F-Measure:*** mede a média entre a precisão e sensibilidade (TAN; STEINBACH; KUMAR, 2009).

$$F - Measure = \frac{2\,x\,VP}{2\,x\,VP + FP + FN} \quad (13)$$

e) **especificidade:** mede a capacidade do modelo em classificar instâncias incorretamente (LAROSE; LAROSE, 2015, tradução nossa):

$$Especificidade = \frac{VN}{VN+FP} \quad (14)$$

f) ***Area Under Curve (AUC)*:** mede a capacidade de o classificador desviar de classificações falsas (SOKOLOVA; LAPALME, 2009, tradução nossa):

$$AUC = \frac{1}{2}.\left(\frac{VP}{VP+FN} + \frac{VN}{VN+FP}\right) \quad (15)$$

g) ***Coeficiente Kappa*****:** coeficiente de avaliação que representa a matriz de confusão (GORELICK; YEN, 2006):

$$k = \frac{P_o - P_a}{1 - P_a} \tag{16}$$

As medidas de qualidade apresentadas são realizadas na tarefa de classificação, podendo ser utilizadas para desempenhar comparações estatísticas entre classificadores (SOKOLOVA; LAPALME, 2009, tradução nossa).

4 EDUCAÇÃO À DISTÂCIA

Segundo Moore e Kearsley (2007) a Educação à Distância (EaD), pode ser definida como uma modalidade na qual o aprendizado ocorre de forma planejada, e durante todo o período de ensino-aprendizagem, professores e alunos estão fisicamente separados, conectados por meio de algum ambiente de interação para transmitir informações.

Em uma abordagem pedagógica, na EaD são definidas as várias relações entre professor e o aluno enquanto separados geograficamente, tais como: interações entre professor-aluno, nível de autonomia e disciplina do aluno e estrutura de programas educacionais (MOORE,1989, tradução nossa).

A EaD está presente em todos os níveis da estrutura organizacional, considerando sistemas formais e não formais de ensino (NUNES, 2009), pode-se citar exemplos como instituições com finalidade única, que dedica-se exclusivamente à EaD; instituições com finalidade dupla, que oferecem cursos tanto presenciais quanto a distância; professores individuais, que realizam cursos a distância sem que a instituição tenha uma unidade especial, entre outros sistemas de ensino (MOORE; KEARSLEY, 2007).

A diversidade de mídias digitais, a Tecnologia de Informação e Comunicação (TIC) e a expansão da Internet contribuem para modificações no campo educacional tanto presencial quanto à distância, contribuindo para a organização, criação e comunicação (LITTO; FORMIGA, 2009). A modelagem de uma aula ou conteúdo de diferentes formas e a utilização de diversas mídias melhoram o processo de ensino-aprendizagem (MATTAR, 2009).

O ambiente que permite a manipulação de conteúdo, a interação entre professor e aluno é chamado de Ambiente Virtual de Aprendizagem (AVA). De forma geral, o AVA pode ser definido como um ambiente que reproduz o espaço de aprendizagem presencial com a utilização de Tecnologias da Informação e Comunicação (TIC) (LITTO; FORMIGA, 2009).

4.1 AMBIENTE VIRTUAL DE APRENDIZAGEM

O AVA ou *Learning Management Systems* (*LMS*), são *softwares web* utilizados para auxiliar e facilitar o aprendizado, dispondo uma plataforma para geração, entrega e gestão de conteúdos informatizados (ARFFIN et al., 2014, tradução nossa), que normalmente são apresentados em forma de cursos (SILVA, 2013).

Para instituições que oferecem cursos ou disciplinas por meio de AVAs, existem opções proprietárias como: WebAula, *Blackboard* e *Pearson Learning,* e gratuitas como: AulaNet, Sakai, TelEduc, Ilias e Moodle (SILVA, 2013). Ao optar por plataformas gratuitas, apesar da instituição não necessitar da compra da licença do *software*, uma equipe local constrói o sistema de gerenciamento de aprendizado baseado na plataforma escolhida, sendo responsáveis pela manutenção do portal (LITTO, 2009), a estrutura de *datacenter* e hospedagem (SILVA, 2013), ao contratar plataformas proprietárias, esses serviços ficam na responsabilidade da empresa contratada (LITTO, 2009).

O *Modular Object-Oriented Dynamic Learning* (*Moodle*) é uma ferramenta que possui licença pública geral (GNU – General Public License) com uma comunidade que reúne 233 países vem sendo amplamente utilizado, considerado como uma plataforma de fácil utilização e instalação (SANTOS, 2016). Atualmente é o ambiente utilizado pela Universidade do Extremo Sul Catarinense (UNESC) para os cursos na modalidade a distância.

A composição de um AVA ocorre por meio de diferentes mídias com intuito de disponibilizar conteúdos e proporcionar interatividade entre grupos de pessoas, para que assim haja a construção de conhecimento (SILVA, 2013). Proporciona em uma nova dimensão o ato de ensinar e aprender, ampliando esses conceitos (LITTO; FORMIGA, 2009), permitindo a colaboração e interação entre pessoas em contextos diferentes ou distante geograficamente (ASSIS; ARAUJO; SOUSA, 2017).

As relações entre professores, alunos e ambiente em EaD por meio dos AVA são abordadas por Moore em 1989, que definiu três tipos de interações: aluno-professor, aluno-aluno e aluno-ambiente (MATTAR, 2009).

4.2 INTERAÇÕES DE MOORE

As interações existentes entre aluno, professor e conteúdo, bem como as formas de comunicação utilizadas por meio de tecnologias, são fatores que influenciam a eficácia do ensino a distância. A fim de explorar esses fatores para melhoria da EAD, foram definidas algumas interações (KEARSLEY; MOORE, 2007).

Na interação entre aluno e ambiente, ocorre o processo de desenvolvimento de compreensão do estudante sobre os conteúdos previamente existentes. O processo do conhecimento é obtido por meio da imersão pessoal no conteúdo, o qual é elaborado e apresentado por profissionais que desenvolvem o curso com intuito de facilitar a interação (MATTAR, 2009).

Os estímulos e interesse pelo conteúdo apresentado ocorrem na interação entre aluno e professor, em que professores auxiliam a aplicar o que foi aprendido e realizam testes avaliativos formais e não formais (KEARSLEY; MOORE, 2007), já os tutores são responsáveis por proporcionar o apoio e incentivo para todo o aluno, principalmente aqueles que possuem baixa autonomia e capacidade de executar as tarefas exigidas (LEITE et al., 2015).

A terceira interação ocorre entre os alunos, podendo ser por meio de grupos, os quais são criados com intuito de desenvolver responsabilidades, servir como meio motivacional para estudo e desenvolver a capacidade de trabalhar em equipe, contribuindo para o processo de aprendizagem (MATTAR, 2009).

As interações também são exploradas na área científica com intuito de investigar e auxiliar o desenvolvimento da EaD. Em *data mining* existe um campo chamado de *Educational Data Mining* ou *EDM* que visa adequar algoritmos e métodos de *data mining* já existentes para observar dados produzidos em AVA, com o intuito de compreender como os alunos aprendem, quais fatores influenciam o seu aprendizado, analisar e definir os seus perfis (BAKER; ISOTANI; CARVALHO, 2011).

4.3 EDUCATIONAL DATA MINING

A *EDM* pode ser considerada como uma área de estudo interdisciplinar, aplicando *data mining*, estatística, aprendizado de máquina e psicologia para explorar dados retirados de ambientes educacionais, gerados por professores e alunos, a fim de auxiliar em questões educacionais (DUTT; ISMAIL; HERAWAN, 2018, tradução nossa). Os objetivos principais de *EDM* é compreender como ocorre e quais fatores afetam a aprendizagem (BAKER; ISOTANI; CARVALHO, 2011), bem como identificar emoções, habilidades e comportamentos dos alunos por meio de métodos específicos (PEÑA-AYALA, 2014, tradução nossa).

Diversos métodos utilizados para modelar dados educacionais provêm originalmente da área de *data mining*. No entanto, são modificados devido as particularidades dos dados, como suas hierarquias e a falta de independência estatística (BAKER; ISOTANI; CARVALHO, 2011). Usualmente, tarefas de *data mining* como classificação, agrupamento e mineração de correlações são aplicadas em pesquisas relacionadas à educação (DUTT; ISMAIL; HERAWAN, 2018, tradução nossa).

Estudos como de Ramesh, Parkavi e Ramar (2013), utilizam classificação para analisar fatores que influenciam o desempenho dos alunos em provas finais a fim de prever suas notas. Santana, Maciel e Rodrigues (2017) baseiam-se nas interações de Moore para criar perfis dos alunos, pesquisas como de Buniyamin, Mat e Arshad (2015), preveem desempenho de alunos com objetivo de intervir e auxiliar aqueles com baixo rendimento. Outras pesquisas como as de Ramos et al. (2017), dividem alunos em grupos, por meio do método de agrupamento, para definir perfis e tipos de participação de alunos.

As pesquisas relacionadas à *EDM* possuem basicamente duas linhas principais, a primeira é desenvolver métodos para realizar suporte a aprendizagem quando alunos estudam a distância por meio de *softwares* educacionais, e a segunda, analisar e desenvolver modelos para melhorar a compreensão dos processos de aprendizagem (BAKER; ISOTANI; CARVALHO, 2011).

4.4 TRABALHOS CORRELATOS

Nos subcapítulos a seguir são apresentados três casos de uso do algoritmo *adaboost.M1* como na utilização em um sistema de previsão de desempenho de alunos utilizando técina multiagente (MALAISE; MALIBARI; ALKHOZAE, 2014, tradução nossa), aplicado para analisar desempenho de alunos com técnicas de *EDM* (AYYAPPAN; KUMAR, 2017, tradução nossa) e em classificação de cobertura de solo urbano (FRANÇA, 2016). Um caso de utilização do algoritmo *adaboost,* em que é implementado na ferramenta s*hell orion data mining engine* (SOUZA, 2016). Apresentam também outros três casos de usos do método *random subspace,* como em detecção de câmeras de fonte (LI et al., 2015, tradução nossa), aplicada na detecção de anomalias na internet (HENKE et al., 2012) e na classificação em espaços com alta dimensionalidade (PATHICAL, 2010, tradução nossa). Por fim, é abordada a utilização de classificadores e técnica de balanceamento em dados educacionais com classes desbalanceadas (GOTTARDO; KAESTNER; NORONHA, 2013).

4.4.1 Sistema de previsão de desempenho dos alunos usando a técnica de mineração de dados de vários agentes

O artigo de Abdullah AL-Malaise, Areej Malibari e Mona Alkhozae, *Students' performance prediction system using multi agent data mining technique*, foi publicado na revista *International Journal of Data Mining & Knowledge Management Process (IJDKP)* em setembro de 2014, trata de realizar uma análise do desempenho de alunos.

Neste artigo é analisado o desempenho de alunos com o intuito de identificar aqueles com baixo rendimento, para isso são utilizados os algoritmos *boosting, adaboost.M1*, *SAMME*, sendo ambos multi-classe e *logitBoost*, uma extensão binária, com o algoritmo de classificação único *C4.5*. Foram coletados dados do sistema *e-learning EMES* que possuem 155 alunos, sendo 105 selecionados para dados de treinamento e o restante para dados de testes.

Os algoritmos multi-classe *adaboost.M1* e *SAMME* obtiveram melhores desempenhos, ambos alcançaram 80% de precisão comparado ao classificador

único *C4.5* e a extensão binária *logitBoost. adaboost.M1* supera *SAMME* em termos de tempo de execução.

4.4.2 Uma nova abordagem de modelos de conjuntos usando o EDM

O artigo *A novel approach of ensemble models by using EDM*, desenvolvido pelos autores Ayyappan e Kumar foi publicado na revista *Indian Journal of Computer Science and Engineering* (*IJCSE*) em dezembro de 2017, no qual analisa dados de desempenho de acadêmicos.

Kumar e Ayyappan (2017) utilizaram os comitês de classificadores *adaboost.M1*, *bagging* e *dagging*[5] com diferentes algoritmos de base, por meio da ferramente *Waikato Environment for Knowledge Analysis (WEKA*[6]*),* para analisar registros de uma base de dados com 7435 instâncias. Os três métodos foram analisados com algoritmos de base bayesianos *bayesnet*, *compliment naivebayes* e *naivebayes*, com os comitês de classificadores *adaboost.M1*, *bagging* e *bagging*, algoritmos de regras *conjuctiveRule*, *decision table* e *JRip* e de árvores de decisão *BFTree*, *decision stump* e *J48*.

O algoritmo que obteve melhores resultados comparados aos demais foi o *adaboost.M1,* utilizando o algoritmo de base *bayesnet*, alcançando até 72,18% de precisão.

4.4.3 Comparação entre classificações de cobertura de solo urbano derivados do WV-2 quanto ao nível de legenda de classificação: estudo de caso para um setor da UNICAMP, SP

A dissertação de mestrado de David Guimarães Monteiro França, Comparação entre classificações de cobertura de solo urbano derivados do WV-2 quanto ao nível de legenda de classificação: estudo de caso para um setor da UNICAMP, SP, aprovada pelo Instituto Nacional de Pesquisas Espaciais de São

[5] Algoritmo de comitê de classificação que cria partições estratificadas a partir dos dados e incrementa cada parte deles para uma cópia do classificador base utilizado.
[6] *Weka* é uma ferramenta de *data mining, open source* disponibilizada em (https://www.cs.waikato.ac.nz/ml/weka/).

José dos Campos em 2016, realizou estudo no sensor *WorldView-2* (*WV-2*) para classificar alvos urbanos.

França (2016) utilizou o algoritmo de comitê de classificação *adaboost.M1* com algoritmos de base de árvores de decisão *C4.5*, *C5.0* e *CART*, para analisar imagens baseada em objetos geográficos (GEOBIA). No estudo, foram gerados doze cenários para uma mesma imagem utilizando dois níveis, com e sem o auxílio de um Modelo Digital de Altura (MDA).

O algoritmo *adaboost.M1* mostrou eficiência em todos os doze cenários, auxiliando os algoritmos de base a atingir a precisão de 74% no nível 1, e 72% no nível 2.

4.4.4 O método de metaclassificação pelo algoritmo adaboost na shell orion data mining engine

O trabalho de conclusão de curso realizado por Ramon Porto de Souza, O método de metaclassificação pelo algoritmo adaboost na shell orion data mining engine, aprovado em 2016 pela Universidade do Extremo Sul Catarinense (UNESC), implementou o algoritmo *adaboost* na ferramenta de *data mining Shell Orion data mining engine*.

O algoritmo desenvolvido foi aplicado em duas bases de dados, uma binária com 203 instâncias e outra de múltiplas casses com 210. Os resultados obtidos foram analisados por meio de medidas de qualidade em classificação, como acurácia e matriz de confusão.

O *adaboost* produziu resultados satisfatórios nas medidas aplicadas, alcançou percentual de acurácia de 98,55% na base binária e 93,3447% ao utilizar múltiplas classes.

4.4.5 Método de subespaço aleatório para identificação de câmera de origem

O artigo desenvolvido por Li, Kotropoulos, Li e Guan, *Random subspace method for source camera identification*, foi publicado e setembro de 2015 no *Internacional Workshop on Machine Learning for Signal Processing* em Boston da

IEEE, que tem como objetivo extrair características de ruído para identificação da câmera de origem.

Foram utilizadas 1200 imagens de 10 câmeras, retiradas da base de dados de *Dresden Image* classificando-as com o método *random subspace* a fim de comparar com método *Principal Component Analysis* (*PCA*) para melhorar os resultados quando os detalhes das imagens estão no conjunto de treinamento.

Ao aplicar o método *RS* para selecionar aleatoriamente subespaços do espaço de características obtidos pelo *PCA,* alcançaram resultados que suprem a interferência de detalhes na imagem, melhorando o desempenho na curva *ROC*, comparado ao método de extração de características utilizando apenas o *PCA*.

4.4.6 Detecção de Intrusos usando Conjunto de k-NN gerado por Subespaços Aleatórios

O artigo, Detecção de intrusos usando conjunto de k-NN gerado por subespaços aleatórios, de Márcia Henke, Celso Costa, Eulanda M. dos Santos e Eduardo Souto, publicado em 2012 no XII Simpósio em Segurança da Informação e de Sistemas Computacionais, propôs a utilização do método *RS* para detecção de anomalias na internet.

Foram utilizados os algoritmos *k-Nearest Neighbor* (*k-NN*), *RS* com algoritmo base *k-NN* e *Triangle Area based Nearest Neighbor* (*TANN*) para realizar testes de detecção e de falsos alarmes em uma base com vetor de 41 características, na qual foi dividida em 40% das características para treinamento e o restante para testes.

Os resultados apresentados indicam que o método *RS* utilizando *k-NN* como base obteve melhor desempenho em taxa de acurácia com 99,97% e detecção com 99,93% comparado aos algoritmos únicos *k-NN* e *TANN*, contudo, ao mesmo tempo obteve a menor taxa de falso alarme, com 0,01%.

4.4.7 Classificação em Espaços de Alta Dimensão através de Conjuntos de Subespaço Aleatórios

A dissertação de mestrado de Santhosh Pathical, *Classification in high dimensional feature spaces through random subspace ensembles*, realizada para

obtenção de título de mestre em Ciências Licenciatura em Engenharia na universidade de Toledo, *The University of Toledo (USA)*, em dezembro de 2010, trata de um estudo baseado no método *RS* para problemas de classificações com características de alta dimensionalidade.

Pathical (2010) simulou o estudo na ferramenta *WEKA* com um conjunto de dados de até 20.000 instâncias, retiradas do repositório *UCI Machine Learning*. Foram utilizados como classificadores bases os algoritmos *C4.5*, *naivebayes* e *k*-vizinho mais próximo.

Os resultados indicam que um conjunto de classificadores pode ser aprendido com um espaço pequeno, como 10% do espaço de características original. Apesar de ser um subconjunto pequeno, o método possui um desempenho robusto, podendo lidar com a alta dimensionalidade dos dados.

4.4.8 Aplicação de técnicas de mineração de dados para estimativa de desempenho acadêmico de estudantes em um AVA utilizando dados com classes desbalanceadas

O artigo, aplicação de técnicas de mineração de dados para estimativa de desempenho acadêmico de estudantes em uma AVA utilizando dados com classes desbalanceadas, de Ernani Gottardo, Celso Kaestner e Robinson Noronha, publicado em 2013 no ICBL *(International Confere on Interactive Computer aided Blended Learning*), propôs a utilização de técnicas de balanceamento em dados educacionais, proveniente de um AVA, nas classes em que os alunos com maior risco de reprovação estavam.

A base de dados utilizada possuía um total de 140 instâncias, nela foi aplicada a técnica de balanceamento chamada *Synthetic Minority Over-sampling Technique (SMOTE)* com percentual de sobreamostragem de 150% apenas na classe nomeada de *C*, a qual originalmente tinha 15 exemplos e passou a ter 37. Foram aplicados os algoritmos de classificação *random forest* e *multilayer perceptron* na ferramenta WEKA.

Os resultados constatam que o percentual de acurácia de *random forest* passou de 13,3% para 78,4% e do algoritmo *multilayer perceptron* passou de 46,7% para 70,3%, ambos são taxas apenas da classe *C*.

5 TRABALHO DESENVOLVIDO

A pesquisa aplica *EDM* em uma base de dados do AVA da UNESC, procurando definir padrões de participação dos alunos baseando-se nos perfis de interação estudados por Moore, a ponto de predizer quais alunos têm mais chances de aprovação ou de reprovação. Para isso, o conjunto de dados é analisado por meio da tarefa de classificação, utilizando os algoritmos de comitê de classificação *adaboost.M1* e *random subspace* para geração de modelos. Os resultados são comparados por meio de medidas de qualidade em classificação como acurácia, coeficiente Kappa, *TP-Rate* e *F-Measure*, a fim de identificar o algoritmo com melhor desempenho.

5.1 METODOLOGIA

A pesquisa é aplicada e de base tecnológica com abordagem quantitativa, tendo isso em vista, para seu desenvolvimento é necessário sua submissão ao Comitê de Ética em Pesquisa da UNESC, na qual este projeto foi submetido e aprovado, conforme parecer de número 2.857.694 (anexo A).

Para alcançar os resultados esperados levou-se em consideração a seguinte metodologia: levantamento bibliográfico envolvendo os conceitos de descoberta de conhecimento em bases de dados, *data mining*, tarefa de classificação, comitê de classificadores, *boosting,* o algoritmo *adaboost.M1, random subspace,* medidas de qualidade em classificação, educação a distância, ambiente virtual de aprendizagem, interações de Moore, *educational data mining* e trabalhos correlatos; seleção de base de dados proveniente do Moodle da UNESC de uma disciplina na modalidade a distância para aplicação dos algoritmos de comitê de classificadores, juntamente com o setor de Educação a Distância da UNESC; submissão ao Comitê de Ética e da Pesquisa da UNESC; definição das dimensões de interação de Moore (aluno-ambiente, aluno-professor e aluno-aluno); seleção dos atributos da base de dados do Moodle; pré-processamento da base; aplicação dos algoritmos *adaboost.M1* e *random subspace*; análise dos modelos obtidos por meio de medidas de qualidade em classificação

5.1.1 Base de dados

A base de dados utilizada nesta pesquisa é proveniente da plataforma Moodle da UNESC, foi selecionada a disciplina de Metodologia Científica e da Pesquisa (MCP) na modalidade a distância, sendo a primeira a ser ofertada de forma institucional a partir do primeiro semestre de 2017, tendo em vista que é autorizada, até o limite de 20% do total do curso, a inserção de disciplinas na modalidade a distância em cursos presenciais (GIACOMAZZO, 2018).

A disciplina é organizada de forma semanal, a MCP de 04 créditos possui 18 semanas no total, em que 14 são para estudos, uma para fórum de dúvidas e três de avaliações (avaliação final, casos especiais e recuperação), já a MCP I e II, de 02 créditos, possuem nove semanas cada, sendo cinco semanas de estudos, uma para fórum de dúvidas e três para avaliações (avaliação final, casos especiais e recuperação).

O Moodle possui aproximadamente 250 tabelas, contudo, para esta pesquisa foram estudadas as que contivessem dados referente a interação do aluno com ambiente, professor e com outros alunos, nas três disciplinas mencionadas anteriormente, ofertadas nos semestres 1/17, 2/17, 1/18 e 2/18 em cursos de graduação presenciais.

Para extração do conjunto de atributos selecionados foram utilizadas 18 tabelas do Moodle (tabela 1), as quais foram autorizadas pelo Setor de Educação à Distância da UNESC, respeitando o termo de confiabilidade (anexo B) do Comitê de Ética da Pesquisa, disponibilizadas pelo setor de Tecnologia da Informação da universidade, em um *script* na linguagem *Structured Query Language (SQL)*. Os dados foram disponibilizados às cegas, ou seja, sem informações pessoais ou código de usuários (alunos, professores, tutores e etc.) bem como conteúdo de mensagens, quiz, fórum ou outra atividade realizada na plataforma.

Tabela 1 - Descrição das tabelas do Moodle utilizadas.

Tabela	Descrição
mdl_user	Tabela de usuários
mdl_context	Tabela que possui o contexto dos cursos
mdl_role	Tabela de perfil dos usuários
mdl_role_assignments	Tabela de matrícula
mdl_course	Tabela de disciplinas
mdl_logstore_standard_log	Tabela com log de acessos
mdl_forum_discussions	Tabela com discussões de fóruns
mdl_forum_posts	Tabela que grava os posts dos fóruns
mdl_forum_subscriptions	Tabela com assinaturas de fóruns
mdl_message_read	Tabela com mensagens lidas
mdl_message	Tabela com mensagens não lidas
mdl_assign	Tabela com atividades/tarefas da disciplina
mdl_assign_grades	Tabela com notas de cada atividade/tarefa
mdl_assign_submission	Tabela com atividades/tarefas submetidas
mdl_quiz	Tabela de quiz
mdl_quiz_grades	Tabela com notas de quiz
mdl_grade_grades	Tabela com todas as notas do Moodle
mdl_grade_items	Tabela de todas as avaliações realizadas na plataforma

Fonte: Do autor.

A versão do Moodle utilizada pela UNESC é 3.6.2+, as tabelas iniciam com o prefixo MDL e o restante é o nome do atributo, a tabela 1 descreve cada uma das utilizadas para esta pesquisa.

5.1.2 Pré-processamento

O pré-processamento consiste em várias técnicas para preparação do conjunto de dados para os algoritmos de *data mining*. Segundo Costa et al. (2012) as tarefas de pré-processamento mais comuns em dados educacionais são: discretização, seleção de atributos para redução de dimensionalidade, tabelas de sumarização e transformação de dados. Para pré-processar o conjunto de atributos selecionados foram necessárias ferramentas como *Google Sheets*[7] e *Weka.*

[7] *Google Sheets* é uma ferramenta para criação de planilhas eletrônicas online, disponibilizada pela Google de forma gratuita.

5.1.2.1 Seleção de atributos

Os atributos foram selecionados baseando-se nas três interações de Moore, dessa forma foram extraídos das tabelas do Moodle dados que estão associados a cada uma delas. Para extração dos atributos necessários em cada interação foi utilizado como base o conjunto de dados utilizado em trabalhos como os de Gottardo, Ernani e Noronha (2012), Santana, Maciel e Rodrigues (2014) e Ramos et al. (2017).

A tabela 2 apresenta os atributos selecionados da base de dados do Moodle e a respectiva interação adotada por Moore, os quais foram escolhidos baseando-se nos estudos citados anteriormente, totalizam 39 atributos, em que seis são para identificar a base de dados e os alunos, 15 se enquadram nas interações de aluno e ambiente, 14 em aluno e professor, três estão associadas a aluno e aluno e a classe, identificada pela nota final. O conjunto de dados conta com 4320 registros.

Tabela 2- Atributos selecionados.

Dimensão	Atributo	Descrição
Identificação da base	id_disciplina	Código de identificação da disciplina no moodle
	nome_disciplina	Nome da disciplina
	semestre	Semestre da disciplina
	curso_disciplina	Curso da disciplina
	id_usuario	Código de identificação do aluno no moodle
	cidade	Cidade em que o aluno mora
Aluno-Ambiente	nunca_acessou	Identificação para alunos que nunca acessaram o moodle (0 = nunca acessou, 1= acessou)
	primeiro_acesso	Data do primeiro acesso do aluno na disciplina no moodle
	ultimo_acesso	Data do último acesso do aluno na disciplina no moodle
	data_criacao_disciplina	Data de criação da disciplina no moodle
	qtde_acessos	Quantidade de acessos na disciplina do moodle
	qtde_grupos	Quantidade de grupos que o aluno está no moodle
	qtde_posts_foruns	Quantidade de posts que o aluno fez no fórum
	qtde_discuss_forum_abertas	Quantidade de discussões abertas em fóruns
	qtde_forum_assina	Quantidade de fóruns que o aluno assina
	nota_quiz	Nota de cada quiz realizado
	status_quiz	Estado em que foi deixado o quiz (finalizado, abandonado, em progresso)
	tentativa	Quantidade de tentativas que o aluno fez
	nota_atividade	Nota de cada atividade realizada
	peso_atividade	Peso da nota da atividade
	status_atividade	Estado em que foi deixada a atividade (submetido, “lixo”)

Aluno-Professor	qtde_msgs_env_aluno_monitor_lidas	Quantidade de mensagens enviadas lidas de um aluno para monitores
	qtde_msgs_env_aluno_tutor_lidas	Quantidade de mensagens enviadas lidas de um aluno para tutores
	qtde_msgs_env_aluno_prof_lidas	Quantidade de mensagens enviadas lidas de um aluno para professores
	qtde_msgs_env_monitor_aluno_lidas	Quantidade de mensagens enviadas lidas de monitores para um aluno
	qtde_msgs_env_tutor_aluno_lidas	Quantidade de mensagens enviadas lidas de tutores para um aluno
	qtde_msgs_env_prof_aluno_lidas	Quantidade de mensagens enviadas lidas de professores para um aluno
	qtde_msgs_env_aluno_monitor_naolidas	Quantidade de mensagens enviadas não lidas de um aluno para monitores
	qtde_msgs_env_aluno_tutor_naolidas	Quantidade de mensagens enviadas não lidas de um aluno para tutores
	qtde_msgs_env_aluno_prof_naolidas	Quantidade de mensagens enviadas não lidas de um aluno para professores
	qtde_msgs_env_monitor_aluno_naolidas	Quantidade de mensagens enviadas não lidas de monitores para um aluno
	qtde_msgs_env_tutor_aluno_naolidas	Quantidade de mensagens enviadas não lidas de tutores para um aluno
	qtde_msgs_env_prof_aluno_naolidas	Quantidade de mensagens enviadas não lidas de professores para um aluno
	posts_forum_ref_post_prof	Quantidades de posts de alunos em fóruns referenciando posts de professores
	posts_forum_ref_post_tutores	Quantidades de posts de alunos em fóruns referenciando posts de tutores
Aluno-Aluno	qtde_msgs_env_aluno_aluno_lidas	Quantidade de mensagens enviadas lidas de um aluno para alunos
	qtde_msgs_env_aluno_aluno_naolidas	Quantidade de mensagens enviadas não lidas de um aluno para alunos
	posts_forum_ref_post_aluno	Quantidades de posts de alunos em fóruns referenciando posts de outros alunos
Classe	nota_final	Resultado obtido por meio da nota final do aluno. Representa se o aluno foi aprovado ou não

Fonte: Do autor.

Para identificar se os atributos selecionados em cada interação são relevantes para a disciplina estudada, Metodologia da Cientifica e da Pesquisa, foram realizadas reuniões junto ao setor de EaD da UNESC a fim de delinear a disciplina ofertada e remover atributos caso não fossem necessários para o perfil de interação.

Com isso, percebeu-se a necessidade de exclusão de alguns atributos, como: *qtde_grupos*, tendo em vista que a disciplina escolhida não agrupa alunos para realização de trabalhos; *qtde_forum_assina*, a recomendação é que não seja realizada a assinatura, pois toda a postagem relacionada ao fórum é encaminhada por e-mail para os assinantes; *tentativa*, removida pois o número de tentativas permitidas para o quiz geralmente é uma; *status_quiz* e *status_atividade* foram removidos pelo fato de que se considerada para avaliação apenas o que está submetido.

Os atributos relacionados as mensagens também foram removidos do conjunto, notou-se que os alunos que repetiram a disciplina tinham a mesma quantidade de mensagens enviadas em ambas, isso porque, quando envia a mensagem pelo Moodle é considerado o contato direto entre os usuários, sem considerar a disciplina que estão.

Na exclusão de atributos de mensagens observou-se que as interações de alunos entre professores e alunos eram pouco frequente, devido ao fato da disciplina ser na modalidade a distância e ofertada em graduações presencias, estas interações não ocorrem exclusivamente pelo AVA, tendo isso em vista, foi utilizado neste trabalho apenas a interação aluno-ambiente.

As postagens de alunos referenciando postagem de outros alunos ou professores poderiam ser referências equivocadas, ou seja, sem intenção de referenciar de fato determinada postagem, visto isso, foram selecionados apenas atributos de interação entre aluno e ambiente.

Referente à dimensão *identificação da base,* apresentada na tabela 2, apesar de não se enquadrar nos perfis de interação de Moore, auxiliam no reconhecimento de inconsistências como, alunos que estão em duas disciplinas iguais no mesmo semestre, problema identificado devido ao processo de otimização realizado pelo setor de EaD, em que são unificados cursos na mesma disciplina. Neste caso, foram removidos da tabela os alunos que estivessem em disciplinas que não tinham registros de acessos, atividades, quiz e fóruns.

As tabelas solicitadas não possuíam identificação dos alunos desistentes ou dos que estavam ou não matriculados na disciplina, dessa forma não foi possível diferenciar os alunos reprovados dos desistentes. A classe “R”, reprovado, do conjunto de dados pode possuir exemplos de alunos desistentes.

5.1.2.2 Tabelas de sumarização

O Moodle é um banco de dados relacional, constituído por diversas tabelas, a tabela de sumarização se faz necessária quando é preciso extrair o conjunto de dados selecionados, para uma única tabela.

Foram criados *scripts* para cada atributo e exportado em .csv. Esses arquivos em .csv foram adicionados em uma única tabela.

5.1.2.3 Derivação de novos atributos

A derivação de novos atributos é uma técnica utilizada a fim de potencializar o conjunto de dados existente, as novas características são baseadas nas já existentes (CECHINEL, 2018). Foram realizadas duas derivações, tais como:

a) o atributo *dias_transcorridos* foi originado da diferença entre *data_criacao* e *primeiro_acesso*, quando o primeiro acesso acontecia antes da criação do curso o atributo possuía um número negativo, logo, essas instâncias foram substituídas pelo número 9999, para representar uma variável desconhecida (WAGNER; MOTTA; DORNELLES, 2004);
b) os atributos *perc_atv_realizadas*, *perc_quiz_realizadas* e *perc_forum_realizadas* foram originados dos atributos que constava a quantidade total disponível e a quantidade total realizada de atividade, quiz e fórum, como por exemplo, *atv_realizadas* e *atv_total*, foi calculada a relação entre os dois atributos para encontrar o percentual realizado.

Após finalizar a etapa de derivação de novos atributos o conjunto de atributos original é apresentado na tabela 3, sendo utilizado nos experimentos realizados, totalizando 4269 registros. Observa-se que os atributos foram reduzidos de 39 para 10 e as dimensões de quatro para duas em relação à tabela 2, apresentada anteriormente.

Ressalta-se que o primeiro atributo *id_disciplina* não está relacionado com nenhuma das interações de Moore, no entanto, foi incluído para possibilitar informações adicionais.

Tabela 3 - Conjunto de atributos original.

Dimensão	Atributo	Descrição
Identificação da base	id_disciplina	Código de identificação da disciplina no moodle

	dias_transcorridos	Dias transcorridos entre a data de criação do curso e o primeiro acesso
	nunca_acessou	Identificação para alunos que nunca acessaram o moodle (0 = nunca acessou, 1= acessou)
Aluno-	qtde_acessos	Quantidade de acessos na disciplina do moodle
Ambiente	perc_atv_realizadas	Percentual de atividades realizadas
	perc_forum_realizadas	Percentual de fóruns realizados
	perc_quiz_realizadas	Percentual de quiz realizadas
	qtdediscussoes	Quantidade de discussões abertas em fóruns por alunos
	qtdepostagem	Quantidade de postagens feitas em fóruns por alunos
Classe	resultado	Resultado obtido por meio da nota final do aluno. Representado por “A” para aprovado e “R“ para reprovado

Fonte: Do autor.

5.1.2.4 Discretização

Segundo Cechinel (2018) a discretização adapta os dados transformando-os em tipos que os algoritmos trabalham, o *adaboost.M1* por exemplo, utiliza apenas classes discretas. Esse método foi aplicado nas classes, separando as notas dos alunos em duas categorias, utilizando-se fórmulas no *Google sheets*:

a) aprovado, representado pela por letra A, considera as notas igual ou acima de 6;
b) reprovado, representado por R, considera as notas abaixo de 6.

O procedimento de discretização também pode ser aplicado em todos os registros contínuos para reduzir a variação deles. Para isso, existem alternativas como a utilização de algoritmos de automação para realização desse método, no *Weka* está disponível como um filtro não supervisionado e foi aplicado em apenas alguns experimentos realizados.

Salienta-se que, a discretização da classe foi aplicada no conjunto de dados original e estão presentes em todos os experimentos realizados nesta pesquisa, tendo em vista os algoritmos utilizados apenas trabalham com classes discretas. No entanto, a discretização de todos os registros foi aplicada no conjunto de dados em apenas alguns experimentos.

5.1.2.5 Balanceamento de classes

O desbalanceamento das classes pode influenciar o desempenho de um modelo de classificação, pois tentem a classificar corretamente somente as classes maioritárias, esse aspecto é definido por Chawla et al. (2002) como um conjunto de dados em que as classes não estão representadas de forma igual.

O conjunto de dados utilizados possui na classe “A”, aprovada, 3385 registros e em “R”, reprovado, 884. A diferença entre os números de registros das classes é de 2501, sendo que a classe “A” é significamente maior, partindo do fato de que número de reprovados na disciplina é menor que os de aprovados.

A técnica *SMOTE* é uma alternativa para tratar classes desbalanceadas, amplamente utilizada em dados educacionais, presente em trabalhos como os de Júnior (2015) e Gottardo, Ernani e Noronha (2012). Nesta técnica são introduzidas instâncias sintéticas na classe minoritária, considerando fundamental a classificação correta da classe “R”, reprovado, por possuírem exemplos de alunos com desempenho inferior.

Aplicou-se a técnica *SMOTE* disponível no *Weka* como um filtro supervisionado, nele é possível definir parâmetros ao aplicar esse método na ferramenta, como o percentual de sobreamostragem que foram usados 100% e 282% e o número de vizinhos, utilizado o valor 5, como sugerido pela *Weka*.

Os percentuais utilizados para a sobreamostragem foram aplicados em diferentes experimentos, ao ampliar na classe “R” o percentual de 100% obtém-se o dobro de registros, ou seja, do total, metade é constituído por exemplos reais e outra metade por exemplos sintéticos, neste caso os registros passa de 884 para 1768, reduzindo a diferença entre as classes para 1617. Ressalta-se que, apesar de haver uma diferença considerável entre as classes, a quantidade de exemplos para a classe “R” respeita a realidade dos dados, ou seja, o número de reprovados é muito menor que as dos aprovados.

Ao ampliar o percentual de 282% as classes “A” e “R” obtém-se classes com número de registros muito próximos, passando de 884 para 3376 exemplos, reduzindo a diferença entre as classes para 9.

Após a finalização da etapa de pré-processamento, o conjunto de atributos foi exportado de uma planilha eletrônica para um arquivo no formato .csv, para assim poder importar na ferramenta de *data mining Weka* e empregar os algoritmos de comitê de classificação.

5.1.3 Execução do Data Mining

Para execução do *data mining* foi utilizada a ferramenta *Weka* por possuir métodos para aplicação do *data mining*, bem como os algoritmos da tarefa de classificação utilizados nesta pesquisa e ser *open source*. Para utilizá-la foi necessário um estudo prévio, de como aplicar e configurar os algoritmos *adaboost.M1* e *random subspace*.

Ao empregar técnicas de *data mining* com algoritmos de classificação é necessário que a base de dados seja dividida em dois conjuntos: treinamento e testes, os modelos são obtidos por meio do conjunto de treinamento e logo são aplicados para classificar as instâncias separadas no conjunto de teste. Para estratificação dos conjuntos foi utilizado o método chamado de *K- fold Cross-Validation*, usando 10 partições, tendo em vista que este número é adequado para obter uma estimativa de erro ideal (WITTEN; FRANK; HALL, 2011).

A aplicação de algoritmos de comitê de classificadores no *Weka* permite a seleção de algoritmos bases em sua configuração. A fim de analisar o desempenho com diferentes classificadores de base, foram utilizados seis algoritmos de árvores de decisão[8], tais como: *decision stump*, que usa apenas um atributo para divisão, ao utilizar atributos discretos a árvore consiste em apenas um único nó, já com atributos numéricos a árvore gerada pode ser mais complexa (FÜRNKRANZ, 2016); *hoeffding tree*, que utiliza um sistema de indução de árvore de decisão incremental, capaz de aprender a partir de um alto fluxo de dados (HULTEN; SPENCER; DOMINGOS, 2001); *random tree,* usa um conjunto de árvores preditoras, chamados de floresta (KALMEGH, 2015); *REP tree*, classificador de árvore de decisão que baseia-se no ganho de informação e em minimizar o erro da variação com a entropia; *J48*, gera

[8] Árvore de decisão é um modelo estatístico de classificação para predição dos dados, possui uma estrutura de árvore em que cada nó interno é um atributo de classe e cada nó-folha possui um rótulo da classe (COSTA et al.,2012).

árvores de decisão C4.5 binárias, podada ou não; e *random forest*, que utiliza árvores preditoras, as quais dependem da amostragem aleatória de vetores com a mesma distribuição para todas as árvores (DEVASENA, 2014). A escolha dos algoritmos é baseada no estudo de Walse, Dharaskar e Thakare (2016), o qual aplicou os mesmos algoritmos de base no comitê de classificador *adaboost.M1.*

Segundo Coelho (2006), para promover desempenhos superiores ao de classificadores únicos, ao utilizar a técnica de comitê de classificadores é necessário que os algoritmos bases utilizados devam apresentar bons resultados individuais. Tendo isso em vista, foram aplicados os seis classificadores apresentados anteriormente como base no *adaboost.M1* e *random subspace,* a fim de verificar os diferentes modelos gerados em cada configuração.

Após a escolha dos algoritmos bases foram aplicados os comitê de classificadores no *Weka*, os algoritmos a*daboost.M1* e o *random subspace* estão disponíveis com esse nome na ferramenta, na categoria “meta”, onde são escolhidos os algoritmos para classificação.

Ao configurar o algoritmo pode-se manipular alguns parâmetros, para este trabalho, a única modificação realizada foi dos algoritmos bases, os modelos foram gerados com diferentes classificadores de árvore de decisão, como citado anteriormente.

O *adaboost.M1* tem como classificador padrão no *Weka* o algoritmo *decision stump* e o *random subspace* tem o *REP Tree*, além disso é possível ressaltar que para ambos os algoritmos a quantidade de iterações foram 10, como a própria ferramenta sugere.

Após definir o método de estratificação e configurar os algoritmos é possível analisar os resultados dos modelos obtidos por meio das medidas de qualidade em classificação.

Para isso, foram desenvolvidos seis experimentos, os quais estão descritos na tabela 4, o objetivo é analisar o comportamento dos classificadores selecionados em relação ao conjunto de dados original, mostrado no subcapítulo 5.1.2.3, e verificar o impacto de técnicas de pré-processamento como discretização em todos os registros e balanceamento de classe nas medidas de qualidade em classificação selecionadas para analisar o desempenho dos algoritmos.

Tabela 4 - Descrição dos experimentos realizados.

Experimentos	Descrição
1	Conjunto de dados original (apenas as classes discretizadas)
2	Aplicação da técnica de discretização em todos os registros no conjunto de dados original
3	Aplicação da técnica SMOTE com 100% no conjunto de dados discretizados
4	Aplicação da técnica SMOTE com 282% no conjunto de dados discretizados
5	Aplicação da técnica SMOTE com 100% no conjunto de dados original
6	Aplicação da técnica SMOTE com 282% no conjunto de dados original

Fonte: Do autor.

O *adaboost.M1* e *random subspace* foram aplicados em cada experimento utilizando 10 iterações, os algoritmos de bases são modificados a cada finalização, nesse sentido o *adaboost.M1* executou com *decision stump*, *hoeffding tree*, *random tree, REP Tree, J48* e *random forest* uma vez em cada experimento, dessa mesma forma procedeu para *random subspace*.

Os resultados dos modelos gerados pelos algoritmos em cada experimento são expressados em medidas de qualidade para classificação, dessa forma é possível analisar o desempenho dos algoritmos em cada um deles.

5.1.4 Análise dos resultados

A análise do desempenho dos modelos gerados por algoritmos de classificação normalmente envolve a avaliação da capacidade de previsão correta das classes. Conforme Cechinel (2018) e Tan, Steinbach e Kumar (2009) algumas das principais medidas para avaliação dos modelos são acurácia, taxa de verdadeiros positivos ou *TP-Rate*, matriz de confusão, coeficiente Kappa e curva ROC.

Para avaliação do desempenho geral do modelo pode-se utilizar medidas como acurácia e Kappa, no entanto, segundo Tan, Steinbach e Kumar (2009) quando se trata de conjunto de dados com classes desbalanceadas, medidas como acurácia, pode não ser apropriada para análise do modelo, principalmente quando a classe de interesse é a minoritária. Com isso, podem ser utilizadas medidas alternativas como taxa de verdadeiros positivos, matriz de confusão, precisão, lembrança, *F-measure* e a curva ROC.

Neste sentido, nesta pesquisa foram utilizadas, para avaliação geral do desempenho dos modelos obtidos, acurácia e coeficiente Kappa, e para analisar o desempenho por classe, foram usados taxa de verdadeiros positivos, a fim de verificar o percentual de acertos e *F-measure* para examinar a média harmônica entre precisão e a lembrança.

Com o objetivo de analisar a significância estatística entre os percentuais de acurácia obtidos aplicou-se o teste estatístico conhecido como *T-test*, que é o resultado de uma comparação por pares (WITTEN; FRANK; HALL, 2011). Nesta pesquisa foi utilizada uma de suas variações, conhecida como *correct resampled t-test*, devido à utilização de estratificação *K- fold Cross-Validation.* Segundo Witten, Frank e Hall (2011) o *T-test* assume amostras independentes, já a variação *correct resampled t-test* utiliza um fator de correção para compensar a dependências das amostras.

O teste de significância está disponível no *Weka* na guia *Experimenter* como *Paired T-Tester (corrected)*, a técnica foi empregada na acurácia com nível de significância de 5%, como a ferramenta sugere.

A análise e os resultados obtidos pelos modelos em cada um desses experimentos estão descritos no subcapítulo 5.2.

5.2 RESULTADOS

Considerando-se os experimentos realizados, os quais foram apresentados no subcapítulo 5.1.3, os resultados obtidos destas análises são descritas a seguir.

Após a aplicação de *adaboost.M1* e *random susbpace* na ferramenta de *data mining Weka* os resultados foram analisados por meio dos percentuais de acurácia, coeficiente Kappa, taxas de verdadeiros positivos e *F-Measure* gerados por cada algoritmo, com intuito de identificar qual o experimento gerou melhores modelos.

Ao selecionar o experimento, foi identificado o algoritmo de base mais apropriado para *adaboost.M1* e *random subspace*, por meio dos modelos gerados utilizando as medidas de qualidade definidas.

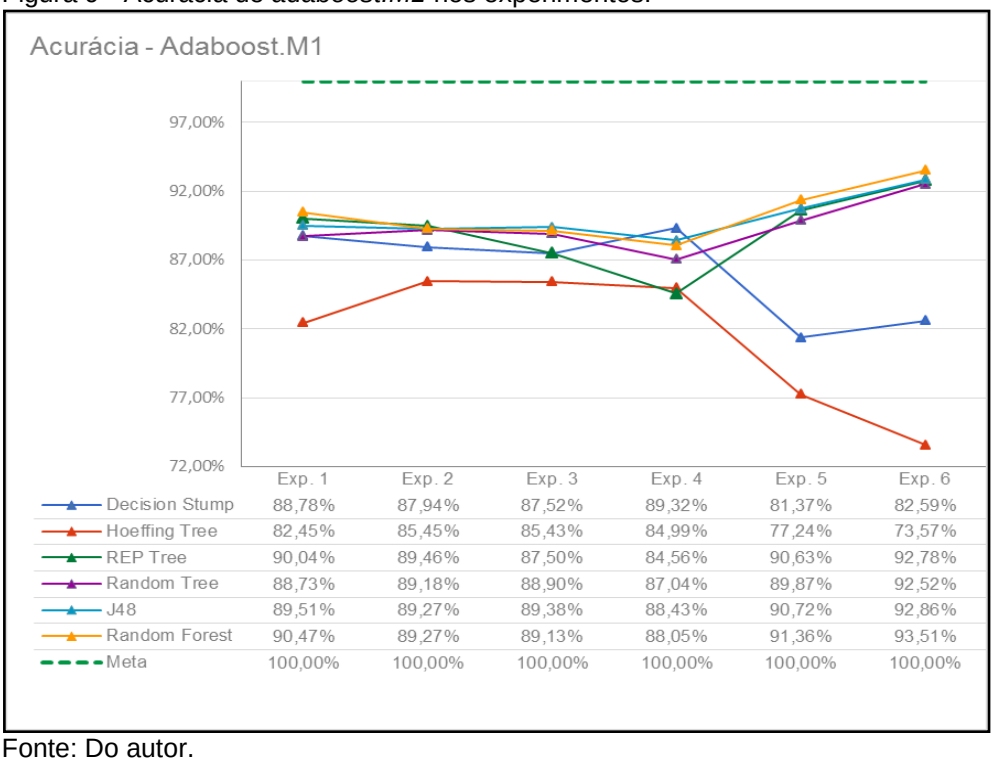
Por fim, para encontrar o modelo com melhores resultados, *adaboost.M1* e *random subspace,* utilizando o algoritmo de base mais apropriado no experimento selecionado, foram analisados por meio das medidas de qualidade definidas nesta pesquisa e pelo teste de significância aplicado no percentual de acurácia, *correct resampled t-test*.

5.2.1 Resultados gerados pela aplicação do algoritmo adaboost.M1

Os modelos obtidos pelo comitê de classificadores *adaboost.M1,* utilizando diferentes classificadores bases, foram analisados por meio de medidas de qualidade em classificação como acurácia, coeficiente Kappa, percentuais de verdadeiros e *F-measure*.

A figura 9 apresenta os percentuais de acurácia atingidos pelos modelos obtidos por *adaboost.M1*, com os diferentes algoritmos bases em todos os seis experimentos realizados. Pode-se observar a linha tracejada nomeada de meta, ela é apresentada em todos os gráficos, quanto mais à taxa de acurácia se aproxima da linha, maior é o percentual de acertos do modelo no experimento, no entanto, deve-se cuidar do *overfitting.*

Figura 9 - Acurácia de *adaboost.M1* nos experimentos.



	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5	Exp. 6
Decision Stump	88,78%	87,94%	87,52%	89,32%	81,37%	82,59%
Hoeffing Tree	82,45%	85,45%	85,43%	84,99%	77,24%	73,57%
REP Tree	90,04%	89,46%	87,50%	84,56%	90,63%	92,78%
Random Tree	88,73%	89,18%	88,90%	87,04%	89,87%	92,52%
J48	89,51%	89,27%	89,38%	88,43%	90,72%	92,86%
Random Forest	90,47%	89,27%	89,13%	88,05%	91,36%	93,51%
Meta	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%

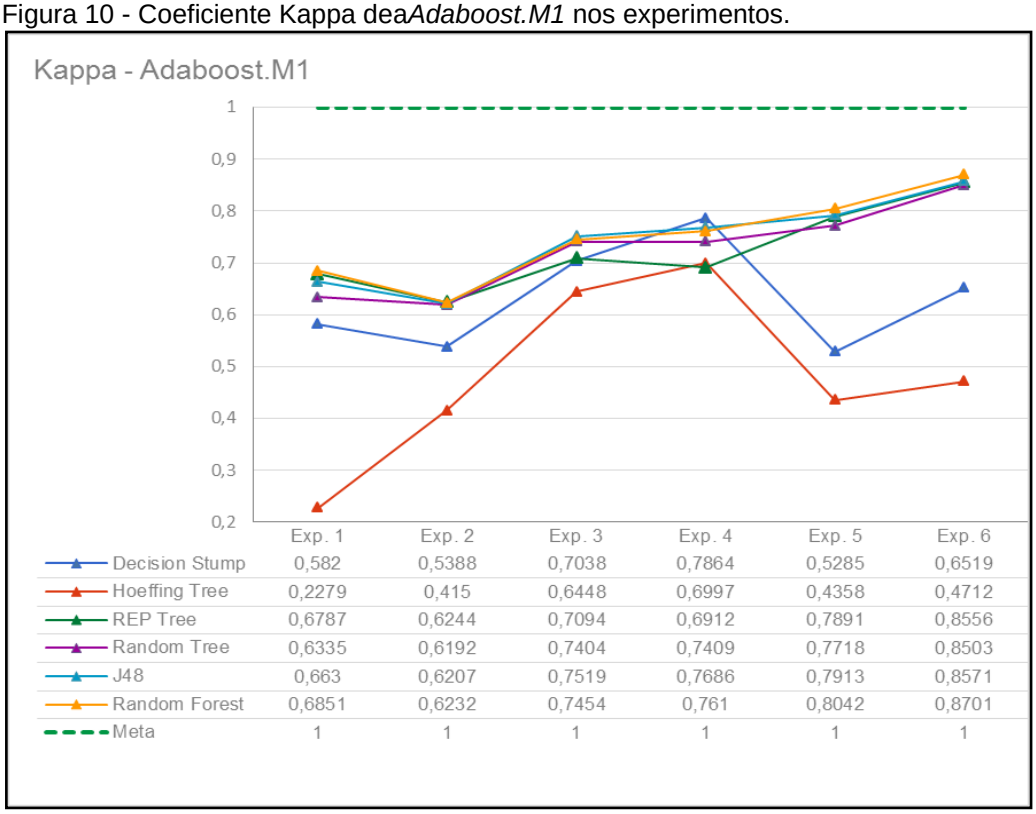
Fonte: Do autor.

Nota-se que os algoritmos que obtiveram os percentuais mais altos foram *random forest* com 93,51%, *J48* com 92,86%, *REP tree* com 92,78% e *random tree* com 92,52%. Analisando a linha de desempenho traçada nos gráficos dos algoritmos citados, todos alcançaram as taxas mais próximas da linha meta no experimento 6.

Destaca-se o algoritmo *hoeffding tree* por atingir as taxas mais baixas comparadas aos demais classificadores. Conforme a figura 10, chegou a percentuais como 77,24% e 73,57% nos experimentos 5 e 6, respectivamente. O *decision stump* também obteve percentuais menores nos mesmos experimentos, atingindo valores como 81,37% no quinto e 82,59% no sexto.

Foram analisados os coeficientes *Kappa* dos modelos gerados pelo a*daboost.M1* (figura 10), da mesma forma que na acurácia foi apresentada a linha meta, os valores mais próximos ao 1 são os modelos que obtiveram melhor desempenho nesta medida.

Figura 10 - Coeficiente Kappa dea*Adaboost.M1* nos experimentos.



	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5	Exp. 6
Decision Stump	0,582	0,5388	0,7038	0,7864	0,5285	0,6519
Hoeffing Tree	0,2279	0,415	0,6448	0,6997	0,4358	0,4712
REP Tree	0,6787	0,6244	0,7094	0,6912	0,7891	0,8556
Random Tree	0,6335	0,6192	0,7404	0,7409	0,7718	0,8503
J48	0,663	0,6207	0,7519	0,7686	0,7913	0,8571
Random Forest	0,6851	0,6232	0,7454	0,761	0,8042	0,8701
Meta	1	1	1	1	1	1

Fonte: Do autor.

Os índices mais altos foram alcançados ao utilizar *random forest*, *J48*, *REP tree* e *random tree*, com 0,8701, 0,8571, 0,8556 e 0,8503, respectivamente. Pode-se notar que as taxas mais próximas da linha meta dos algoritmos citados foram obtidas no experimento 6.

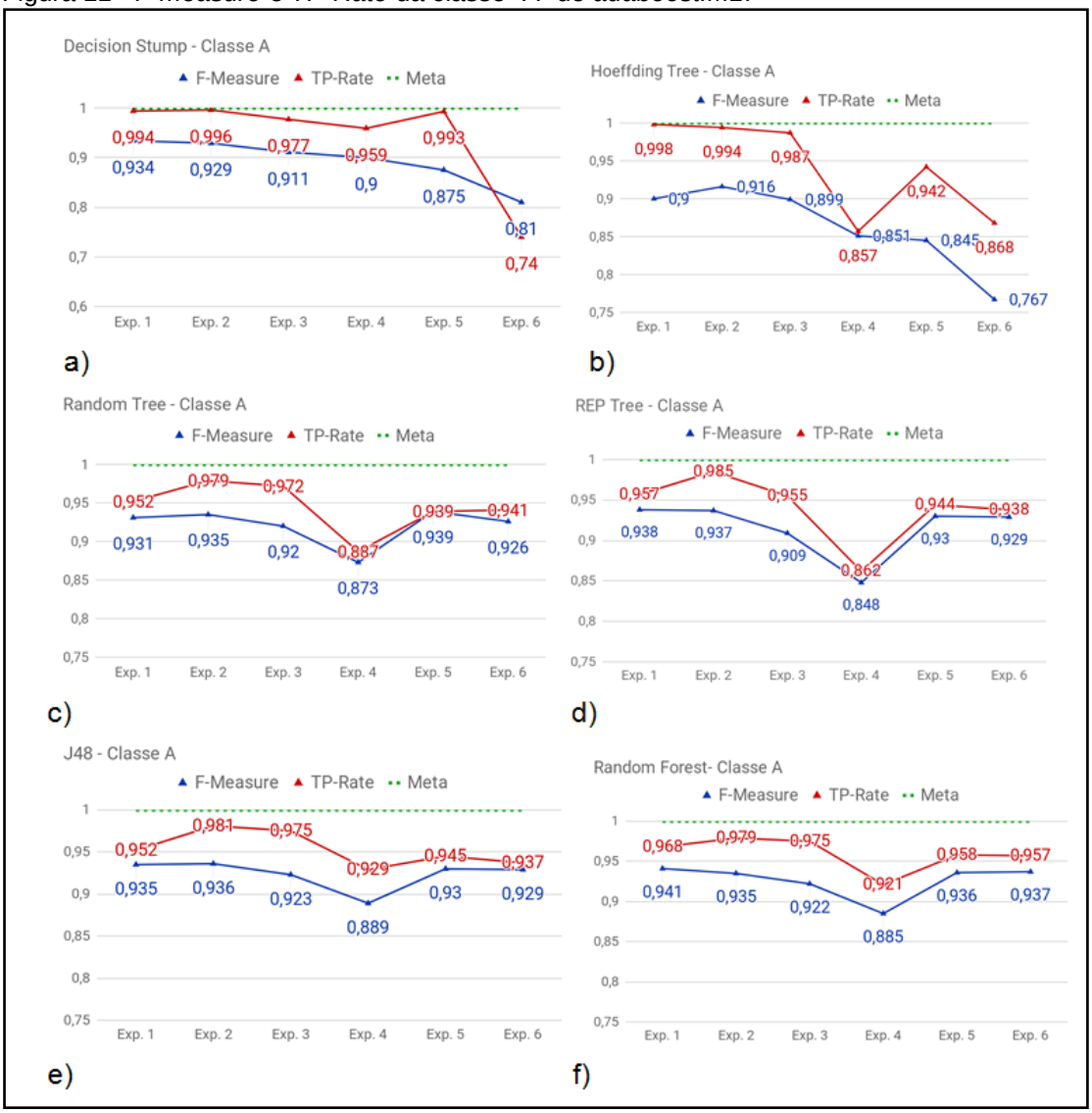
Ao utilizar *hoeffding tree*, observa-se que nos experimentos 1, 2, 5 e 6 chegou a valores como 0,2279, 0,415, 0,4358 e 0,4712, respectivamente, que são os mais distantes da linha meta comparado aos demais algoritmos.

Em relação ao desempenho geral dos modelos, considerando tanto acurácia quanto coeficiente Kappa, os melhores resultados estão concentrados nos experimentos 5 e 6, nos quais foram aplicados a técnica *SMOTE* com percentuais de 100% e 282% sobre o conjunto de dados original, exceto ao utilizar os classificadores como: *decision stump* que obteve resultados maiores no experimento 4, em que o conjunto de dados estava discretizado e com *SMOTE* 100%; e *hoeffding tree*, o qual alcançou melhor resultado em termos de acurácia, no 2, cujo conjunto de dados estava apenas discretizado e no coeficiente Kappa no experimento 4.

A fim de analisar o desempenho dos modelos gerados nas classes “A”, aprovado e “R”, reprovado, verificou-se a taxa de verdadeiros positivos e a medida *F-Measure* para cada uma delas.

As taxas de verdadeiros positivos para a classe “A” que ficaram mais próximas da linha meta são apresentados na figura 11, itens *b* e *a*, os resultados foram de *hoeffding tree* com 0,998 e 0,994 nos experimentos 1 e 2 e *decision stump* com 0,996 e 0,994 nos experimentos 2 e 1, respectivamente. No entanto, pode-se observar que da mesma forma que possuíram as taxas mais altas, os mesmos algoritmos também alcançaram os valores mais distantes da linha meta, *decision stump* chegou a 0,74 no experimento 6 e *hoeffding tree* a 0,857 no 4.

Figura 11 - *F-measure* e *TP-Rate* da classe "A" de *adaboost.M1*.



Fonte: Do autor.

Ao analisar a medida *F-Measure* para a classe “A”, pode-se perceber que os algoritmos com os valores mais altos são de *random forest* com 0,941, 0,937 nos experimentos 1 e 6 e de *random tree* com 0,939 no 5, conforme figura 12 itens *f* e *a*, respectivamente.

Observa-se que em relação aos experimentos 1 e 2, nos quais utilizou-se classes desbalanceadas, os demais experimentos, que aplicam técnica para balanceamento, tem taxas de verdadeiros positivos menores na classe “A”. O mesmo ocorre ao comparar as taxas obtidas na medida *F-Measure*, exceto ao utilizar *random tree*, que no experimento 5 obteve a maior taxa nesta medida, se comparado ao demais experimentos.

A classe “R”, por sua vez, possui os resultados com valores mais próximos da linha meta, em termos de verdadeiros positivos, ao utilizar *J48*, *REP tree* e *random forest*, chegando a valores como 0,92, 0,918 e 0,913, respectivamente, todos no experimento 6, conforme figura 12, itens *e*, *d* e *f* .

Ainda em relação a taxas de verdadeiros positivos, destaca-se o algoritmo *hoeffding tree,* apresentado no item *b* da figura 12, por possuir as taxas mais distantes da linha meta em relação aos demais algoritmos, no experimento 1 chegou a 0,16 e no 2 a 0,32, ou seja, classificou mais da metade das instâncias da classe “R” como sendo “A”.

Para a medida *F-Measure* na classe “R”, os melhores resultados são obtidos ao utilizar os algoritmos *random forest*, *J48* e *REP tree,* com valores 0,934, 0,928 e 0,927 respectivamente, todos no experimento 6 conforme apresentados nos itens *f*, *e* e *d* na figura 12.

Da mesma forma que nas taxas de verdadeiros positivos, na medida *F-Measure* ao utilizar o classificador *hoeffding tree,* os resultados atingidos possuem os valores mais distantes da linha meta comparado aos demais classificadores, chegando a 0,274 no experimento 1 e 0,477 no 2, conforme item *b* da figura 12.

Figura 12 - *F-Measure* e *TP-Rate* da classe "R" de *adaboost.M1*.

Fonte: Do autor.

Nota-se que, os melhores resultados obtidos para a classe “R” estão concentrados nos experimentos que utilizam a técnica de balanceamento, destacando o experimento 6, já os resultados mais baixos foram obtidos nos experimentos 1 e 2, em que não se aplicou técnica de balanceamento, exceto ao utilizar o classificador *decision stump* no experimento 5, o qual atingiu uma de suas taxas de verdadeiros positivos mais baixas, 0,47, conforme item *a* da figura 12.

O desempenho por classe dos modelos gerados mostra que ao balancear a classe “R” há uma queda no desempenho da classe “A”, no entanto identificar

corretamente o perfil de interação dos alunos reprovados pode ser relevante para medidas pedagógicas.

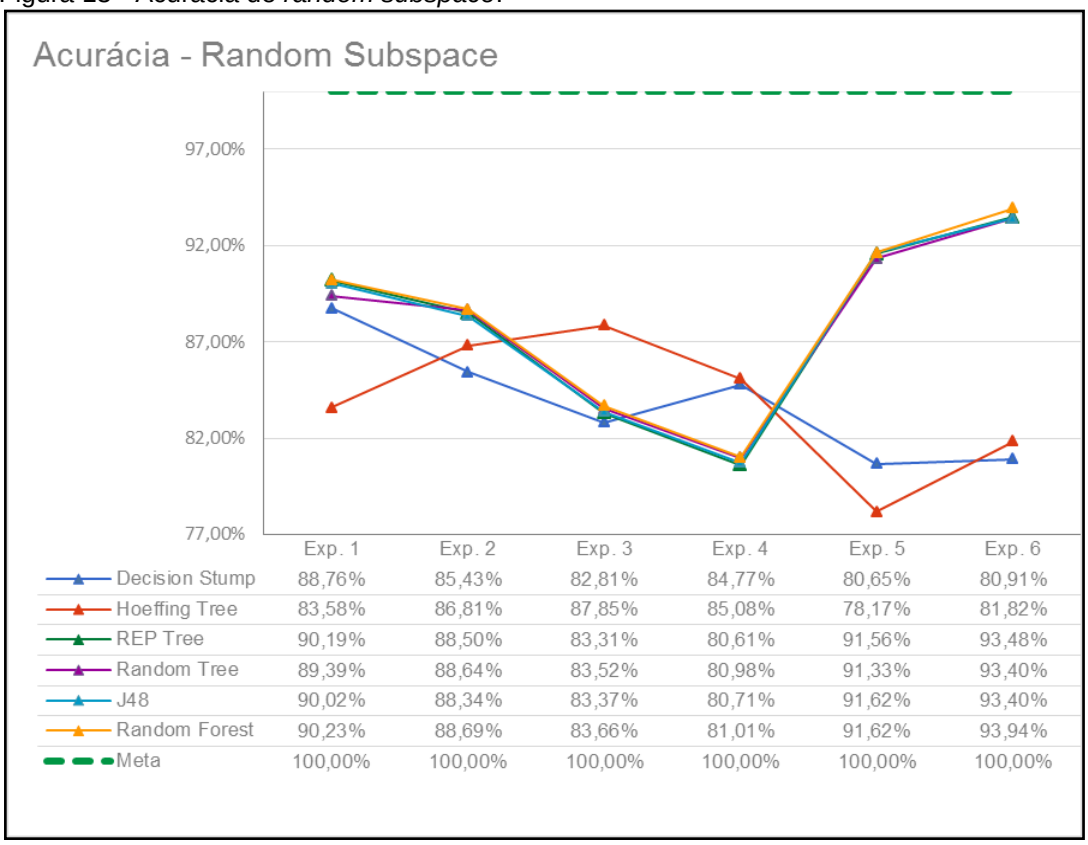
Tendo isso em vista, para o algoritmo *adaboost.M1*, utilizando diferentes classificadores bases, pode-se verificar que conforme as análises realizadas, foram selecionados os experimento 5 e 6 por terem bons resultados em relação ao desempenho geral e por classe, principalmente na reprovado, comparado aos demais experimentos.

5.2.2 Resultados gerados pela aplicação do algoritmo random subspace

Os modelos obtidos pelo comitê de classificadores *random subspace,* utilizando diferentes classificadores bases, foram analisados por meio de medidas de qualidade em classificação como acurácia, coeficiente Kappa, percentuais de verdadeiros e *F-measure*.

Conforme a figura 13 os algoritmos que mais se aproximam da linha meta são *random forest*, *J48*, *REP tree* e *random tree*, atingindo percentuais como 93,94%, 93,48%, e 93,40%, respectivamente, todos no experimento 6. Analisando a linha de desempenho traçada dos classificadores citados anteriormente, todos alcançaram as taxas mais baixas no experimento 4.

Figura 13 - Acurácia de *random subspace*.



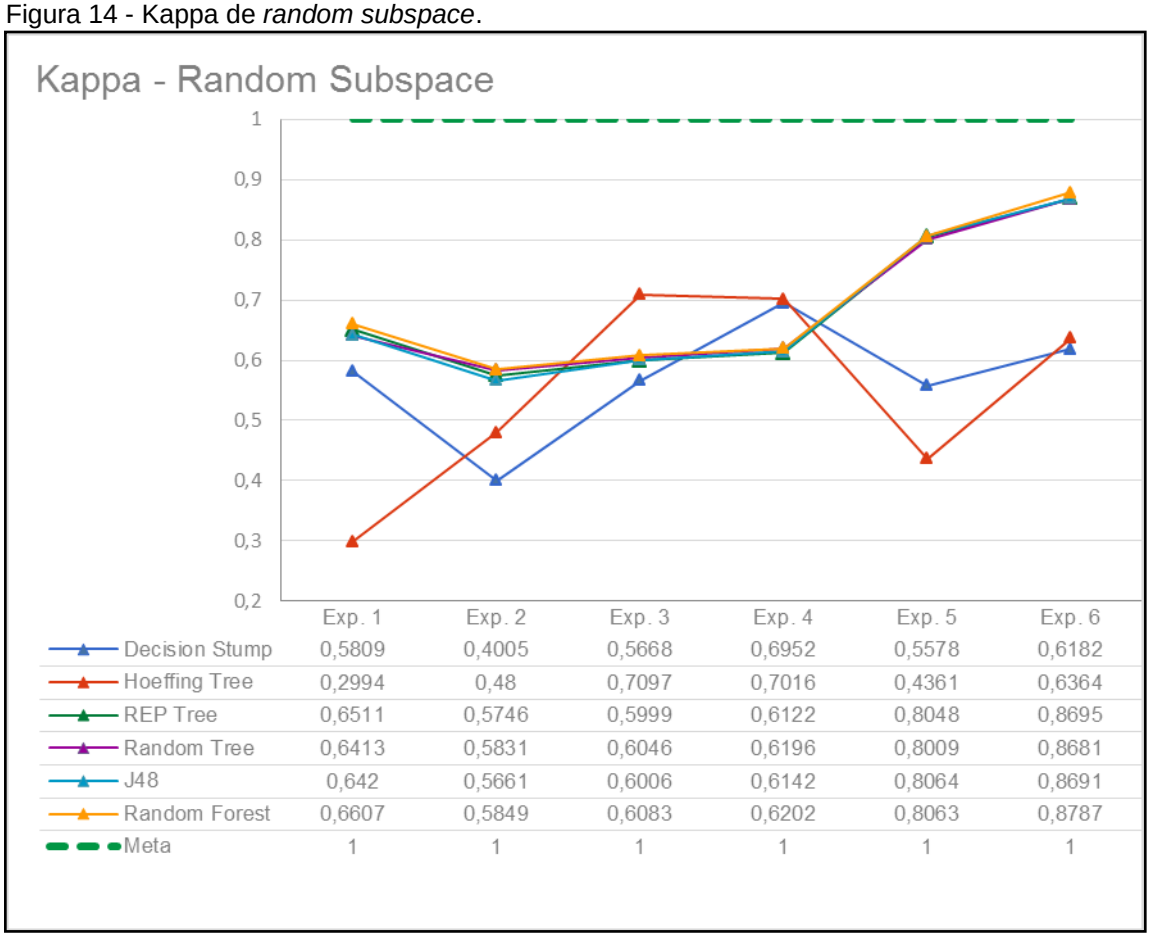
	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5	Exp. 6
Decision Stump	88,76%	85,43%	82,81%	84,77%	80,65%	80,91%
Hoeffing Tree	83,58%	86,81%	87,85%	85,08%	78,17%	81,82%
REP Tree	90,19%	88,50%	83,31%	80,61%	91,56%	93,48%
Random Tree	89,39%	88,64%	83,52%	80,98%	91,33%	93,40%
J48	90,02%	88,34%	83,37%	80,71%	91,62%	93,40%
Random Forest	90,23%	88,69%	83,66%	81,01%	91,62%	93,94%
Meta	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%

Fonte: Do autor.

Os percentuais de acurácia mais distantes da linha meta são de *hoeffding tree* com 78,17% no experimento 5, *REP tree* com 81,00% no 4 e *decision stump* no 5 com 80,65%.

Ao analisar o coeficiente Kappa, pode-se observar na figura 14 que os valores mais altos foram alcançados ao utilizar *random forest*, *REP tree*, *J48*, e *random tree*, com 0,8787, 0,8695, 0,8691 e 0,8681, respectivamente. Pode-se notar que nos algoritmos citados as taxas mais próximas da linha meta foram obtidas no experimento 6 e as mais distantes no 2.

Figura 14 - Kappa de *random subspace*.



	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5	Exp. 6
Decision Stump	0,5809	0,4005	0,5668	0,6952	0,5578	0,6182
Hoeffing Tree	0,2994	0,48	0,7097	0,7016	0,4361	0,6364
REP Tree	0,6511	0,5746	0,5999	0,6122	0,8048	0,8695
Random Tree	0,6413	0,5831	0,6046	0,6196	0,8009	0,8681
J48	0,642	0,5661	0,6006	0,6142	0,8064	0,8691
Random Forest	0,6607	0,5849	0,6083	0,6202	0,8063	0,8787
Meta	1	1	1	1	1	1

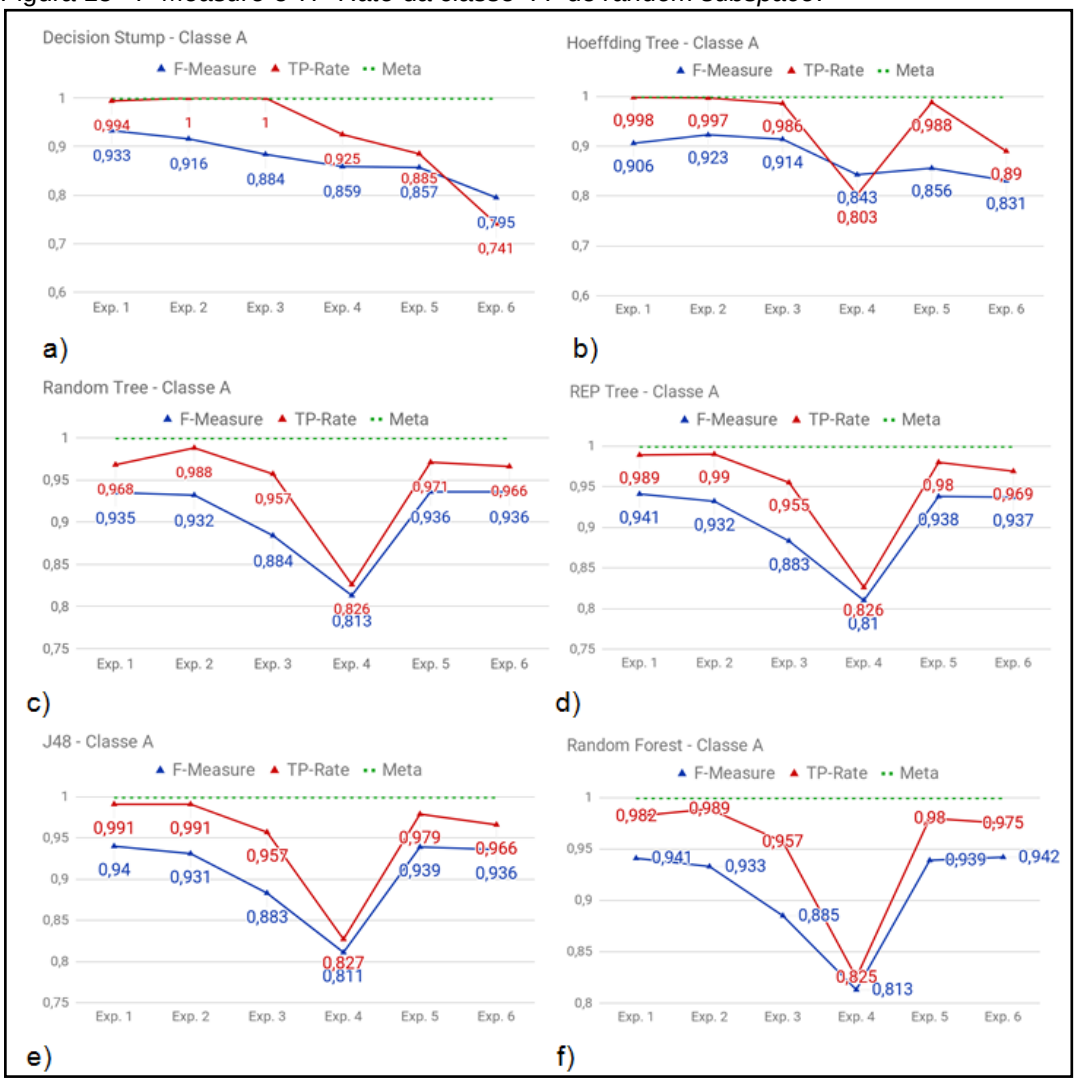
Fonte: Do autor.

Os resultados obtidos por *hoeffding tree,* são os mais distantes da linha meta, com 0,2994 no experimento 1 e 0,4361 no 5. O classificador *decision stump* também obteve valores baixos, destaca-se o mais distante, 0,4005 no experimento 2.

Pode-se observar que exceto ao utilizar os classificadores *decision stump* e *hoeffding tree*, os demais algoritmos obtiveram melhores resultados em termos de acurácia e coeficiente Kappa nos experimentos 5 e 6.

Ao analisar as taxas de verdadeiros positivos da classe “A” na figura 15, observa-se que os melhores resultados são obtidos pelo algoritmo *decision stump*, que tanto no experimento 2 quanto no 3 conseguiu classificar todos as instâncias da classe “A” corretamente, alcançando a linha meta no gráfico, ou seja, chegou a 1 (item *a* da figura 15). Ainda assim, pode-se destacar resultados como os de *hoeffding tree* que no experimento 1 chegou a 0,998 e *J48*, que atingiu 0,991 para os experimentos 1 e 2.

Figura 15 - *F-Measure* e *TP-Rate* da classe "A" de *random subspace*.



Fonte: Do autor.

O algoritmo *decision stump* atingiu a taxa de verdadeiros positivo mais distante da linha meta em relação aos demais classificadores, conforme a figura 15, chegou a taxa de 0,741 no experimento 6. Ao analisar a linha de desempenho traçada nos gráficos dos demais algoritmos, todos alcançaram as taxas mais baixas no experimento 4, conforme itens *b, c, d, e* e *f* da figura 15.

Para a medida *F-Measure* na classe “A” destaca-se os algoritmos *random forest* nos experimentos 6 e 1, *J48* e *REP tree* no experimento 1 por possuírem valores mais altos, conforme itens *f, e* e *d* da figura 15. Da mesma forma, como na taxa de verdadeiros positivos, os resultados mais distantes da linha de meta para

esta medida foram no experimento 4, exceto ao utilizar o classificador *decision stump*, que possui valor mais baixo no 6.

Percebe-se que, de modo geral, os melhores resultados em termos de taxas de verdadeiros positivos e *F-Measure* nas análises geradas para o algoritmo *random subspace* na classe “A” estão nos experimentos 1 e 2, os quais não possuem técnicas de balanceamento empregadas, já as taxas mais baixas estão no experimento 4, o qual foi aplicado a técnica *SMOTE* em 282% sobre o conjunto de dados discretizado, exceto para o *decision stump*.

Na classe “R”, os percentuais mais próximos da linha de meta, quando se trata da taxa de verdadeiros positivos, são alcançados por algoritmos como *random forest* (figura 16 item *f*), *J48* (figura 16 item *e*), *random tree* (figura 16 item *c*) e *REP tree* (figura 16 item *d*) chegando a 0,904, 0,902, 0,902 e 0,901, respectivamente, todos no experimento 6. Ao analisar as linhas mais distantes da meta, pode-se observar que, exceto para *hoeffding tree*, estão no experimento 2.

Nota-se na figura 16 que quando se trata da medida *F-Measure* os valores mais altos são obtidos ao utilizar *random forest*, chegando a 0,937 no experimento 6 (item *f*) e *decision stump* com 0,933 no experimento 1 (item *a*). Os valores mais distantes da linha meta foram obtidos por *hoeffding tree* com 0,353 no experimento 1 (item *b*) e *random tree* com 0,495 no 2 (item *c*).

Figura 16 - *F-Measure* e *TP-Rate* da classe "R" de *random subspace*.

Fonte: Do autor.

O desempenho por classe dos modelos gerados mostra que ao balancear a classe “R” há uma queda nos percentuais da classe “A”, principalmente em taxas de verdadeiro positivo, no entanto, da mesma forma que para *adaboost.M1* é relevante identificar corretamente o perfil de interação dos alunos reprovados.

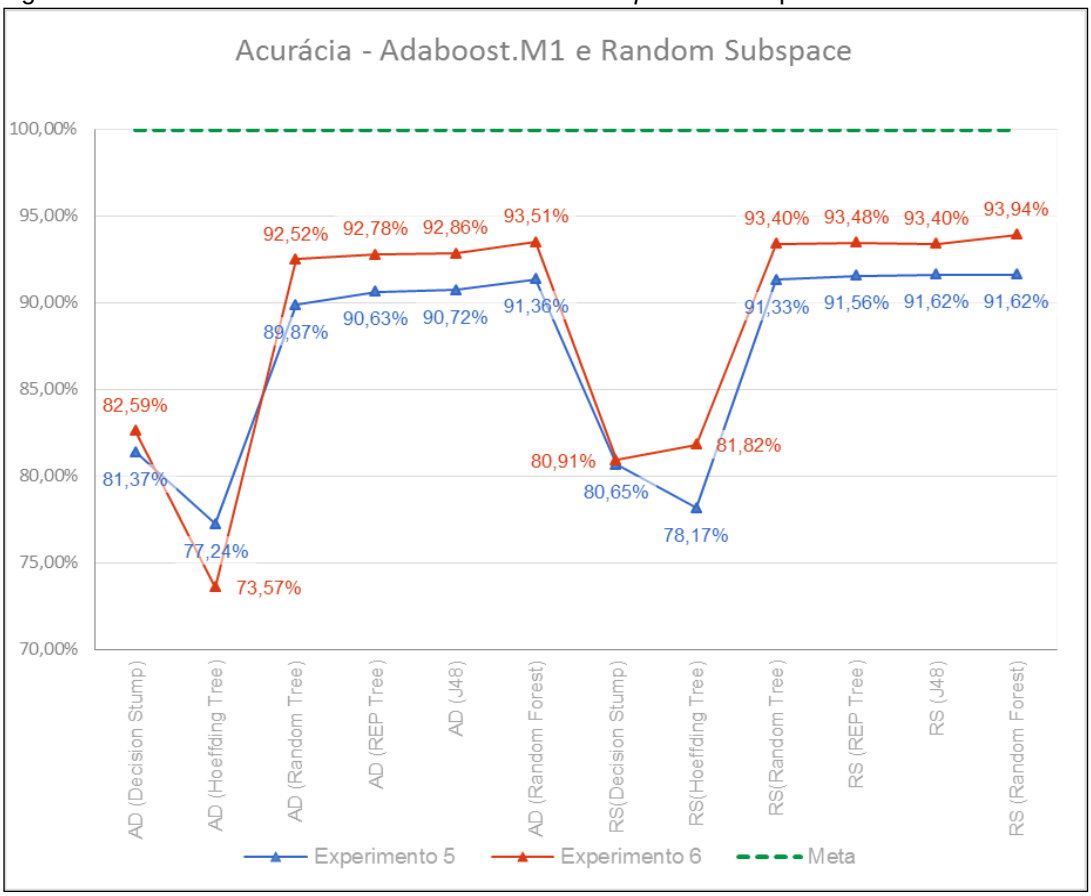
Conforme as análises realizadas, para o algoritmo *random subspace,* utilizando diferentes classificadores bases, foram selecionados os experimento 5 e 6, da mesma forma que para *adaboost.M1*, por terem bons resultados em relação ao desempenho geral e por classe, principalmente na reprovado, comparado aos demais experimentos.

5.2.3 Comparação entre os melhores modelos gerados por adaboost.M1 e random subspace nos experimentos 5 e 6

Tendo em vista que os modelos obtidos por *adaboost.M1* e *random subspace* tiveram melhores desempenhos nos experimentos 5 e 6, foram realizadas análises entre os dois experimentos utilizando acurácia, coeficiente Kappa, taxas de verdadeiro positivo e *F-Measure* a fim de encontrar o experimento mais apropriado para identificar os perfis de interação dos alunos.

Os percentuais de acurácia obtidas nos experimentos 5 e 6, por cada algoritmo são expressas na figura 17, observa-se que os algoritmos com prefixo *AD* são de *adaboost.M1* e com *RS random subspace*.

Figura 17 - Acurácias de *adaboost.m1* e *random subspace* nos experimentos 5 e 6.



Fonte: Do autor.

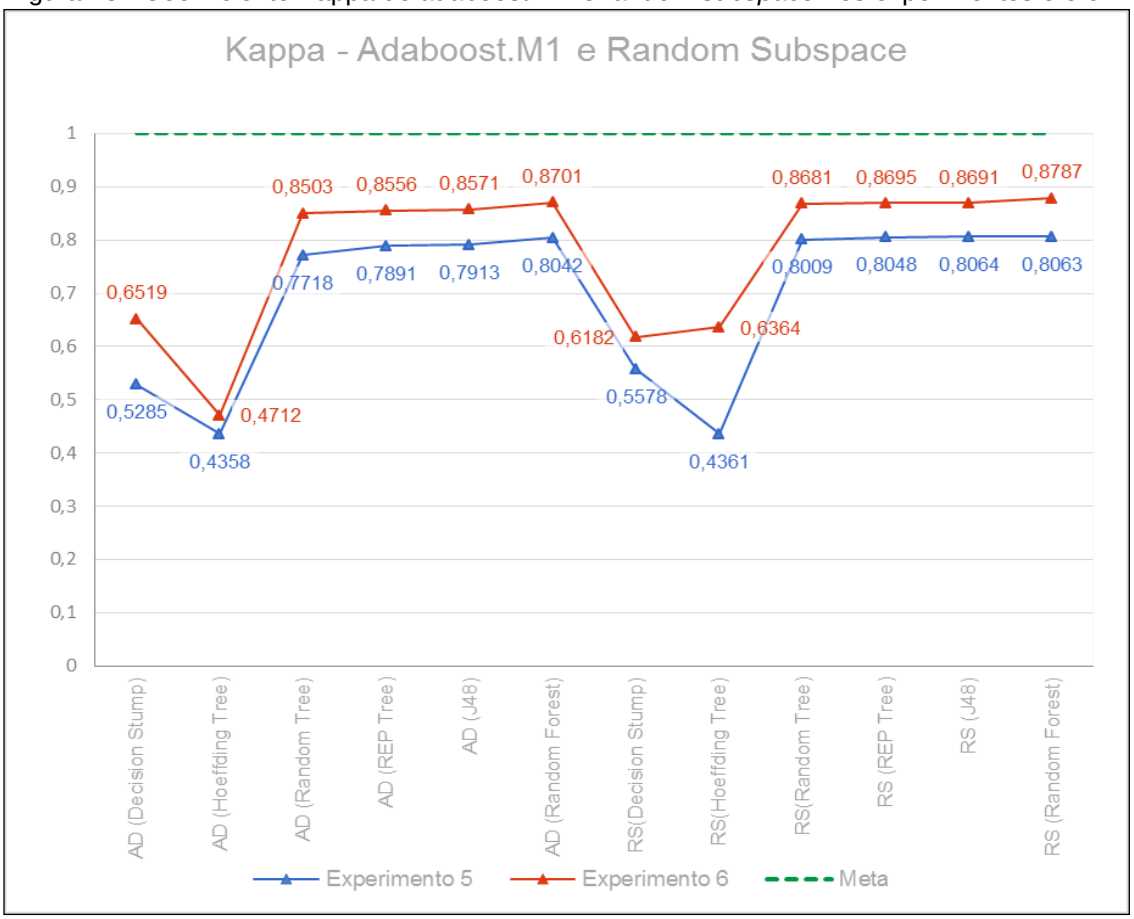
Ao analisar o desempenho dos algoritmos, exceto quando *adaboost.m1* utilizada *hoeffding tree*, os modelos obtidos no experimento 6 tiveram percentuais superiores em relação aos do 5.

No experimento 5 destaca-se o algoritmo *random subspace* utilizando *random forest* e *J48*, ambos chegaram a 91,62%, já no 6, novamente *random subspace* obteve resultados superiores, ao usar *random forest* alcançou o percentual de 93,94% e com *REP tree* 93,48%.

Da mesma forma, foi realizada a comparação entre os resultados obtidos no coeficiente Kappa de ambos os experimentos, a figura 18 mostra que, tanto para *adaboost.M1* quanto para *random subspace,* os resultados do experimento 6 estão mais próximos da linha meta, do que comparado aos do 5.

Os melhores resultados no experimento 5 são de *random subspace*, utilizando *J48* (0,8064) e *random forest* (0,8063), já no 6 são de *random susbpace* com *random forest* (0,8787) e *adaboost.m1*com *random forest* (0,8701).

Figura 18 - Coenficiente Kappa de *adaboost.m1* e *random subspace* nos experimentos 5 e 6.



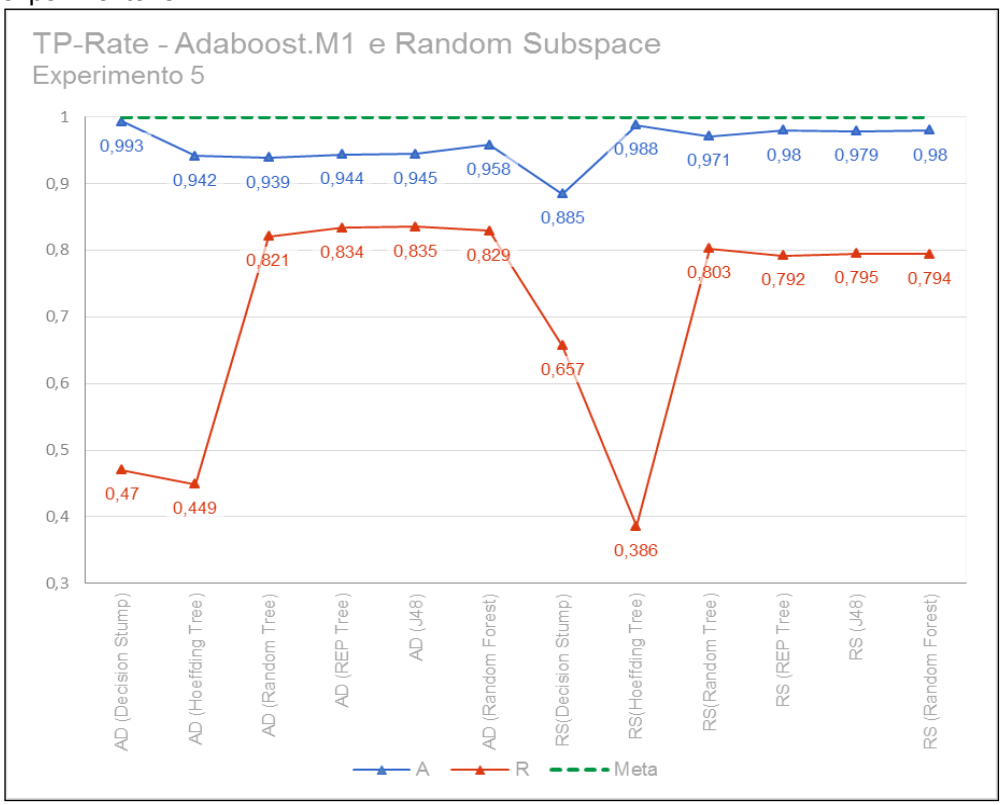
Fonte: Do autor.

As análises mostradas anteriormente dizem respeito ao desempenho geral dos classificadores, com intuito de analisar também os resultados por classe,

foram comparadas as taxas de verdadeiros positivos e *F-Measure* obtidos em cada experimento.

A figura 19 mostra as taxas de verdadeiros positivos na classe “A” e “R” no experimento 5 para *adaboost.M1* e *random subspace*. Os melhores resultados para a classe “A” são de *adaboost.m1* utilizando *decision stump* (0,993) e *random subspace* também usando como classificador base o *decision stump* (0,988). Para a classe “R” os algoritmos com resultados superiores são de *adaboost.m1* com *J48* (0,835) e com *REP tree* (0,834).

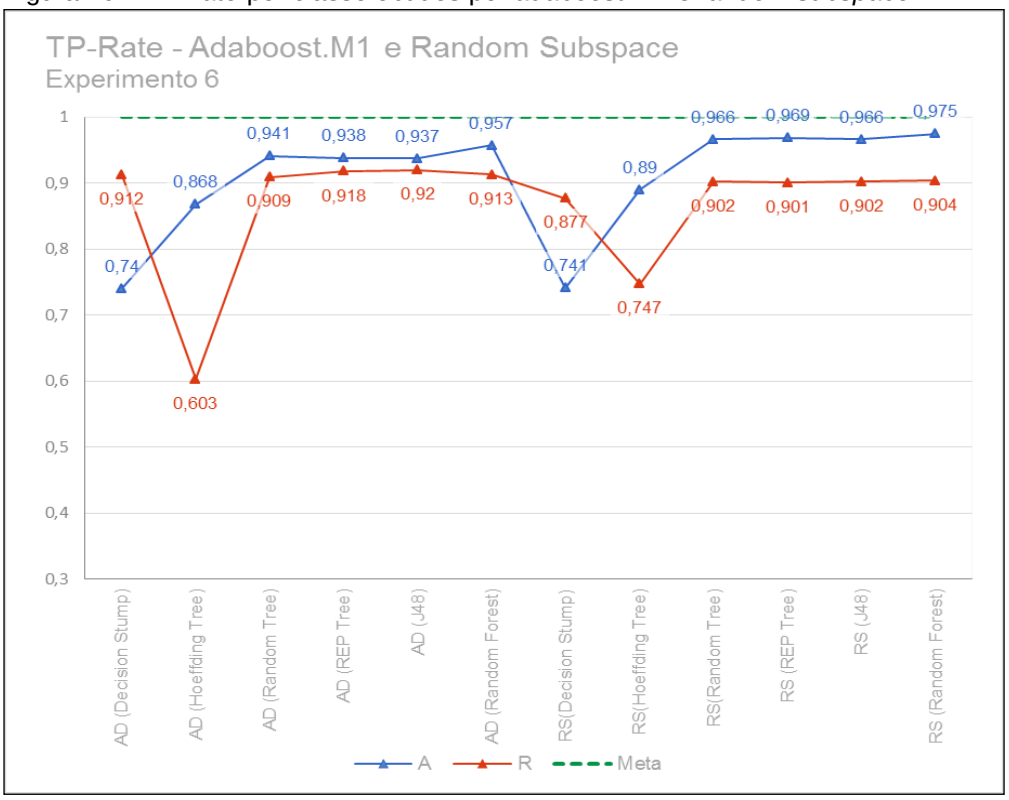
Figura 19 - *TP-Rate* por classe obtidos por *adaboost.m1* e *random subspace* no experimento 5.



Fonte: Do autor.

A figura 20 mostra as taxas de verdadeiros positivos na classe “A” e “R” no experimento 6 para *adaboost.M1* e *random subspace*. Os melhores resultados para a classe “A” são de *random subspace* utilizando *random forest* (0,975) e *random tree* (0,988), já para a classe “R” são de *adaboost.M1* utilizando *J48* (0,92) e *REP tree* (0,918).

Figura 20 - *TP-Rate* por classe obtidos por *adaboost.m1* e *random subspace*.



Fonte: Do autor.

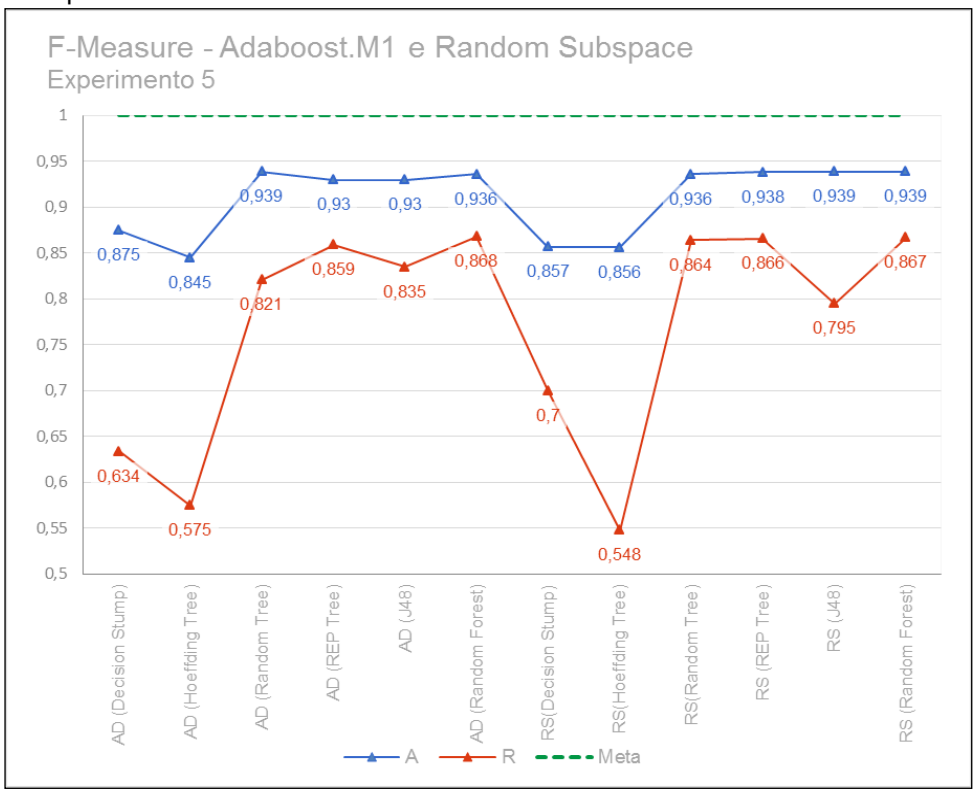
Ao comparar os resultados da figura 19 com a 20, pode-se observar que para a classe “A” o experimento 5 obteve melhores resultados, exceto ao utilizar *random tree* no *adaboost.M1* (figura 20), que somente neste caso a taxa foi maior no 6. Para a classe “R”, nota-se que tanto para *adaboost.M1* quanto para *random subspace* (figura 20) os resultados mais próximos a linha meta são do experimento 6.

Ainda no experimento 6, pode-se destacar que as linhas traçadas pela classe “A” e “R” estão mais próximas entre si do que as do 5, mostrando que, apesar do experimento 6 diminuir as taxas de verdadeiros positivos na classe “A” e maximizado a de “R”, as predições entre as classes estão muito próximas.

A figura 21 apresenta os valores obtidos na medida *F-Measure* no experimento 5 por ambos algoritmos, pode-se observar que para a classe “A” os valores mais próximos da linha meta são obtidos por *adaboost.m1* com *hoeffding tree, random subspace* utilizando *J48* e *random forest,* todos chagaram ao mesmo

valor, 0,939, já para a classe “R” destacam-se *adaboost.m1* com *random forest* (0,868) e *random subspace* com *random forest* (0,867).

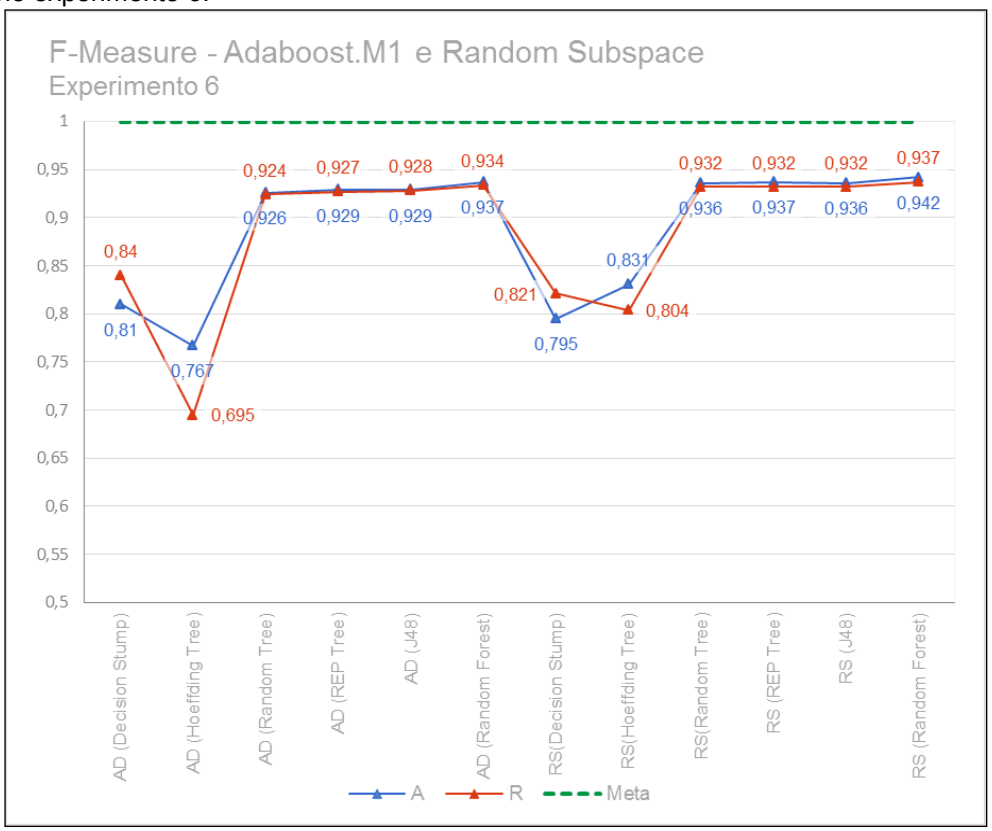
Figura 21 - *F-Measure* por classe obtidos por *adaboost.m1* e *random subspace* no experimento 5.



Fonte: Do autor.

A figura 22 apresenta os valores obtidos na medida *F-Measure* no experimento 6 por ambos algoritmos, pode-se observar que para a classe “A” os valores mais próximos da linha meta são obtidos por *random subspace* com *random forest* (0,937) e *adaboost.m1* com o mesmo classificador base (0,934), já para a classe “R” destacam-se novamente *random subspace* com *random forest* (0,942), *adaboost.m1* utilizando *random forest* e *random subspace* com *REP tree* alcançando o mesmo valor, 0,937.

Figura 22 - *F-Measure* por classe obtidos por *adaboost.m1 e random subspace* no experimento 6.



Fonte: Do autor.

Pode-se observar que tanto para *adaboost.m1* quanto para *random subspace* , quando se trata da classe “A” os valores são mais altos no experimento 5 e para a classe “R” são no 6.

Como na taxa de verdadeiros positivos, na medida *F-Measure* pode-se perceber que no experimento 6 (figura 22), as linhas traçadas pela classe “A” e “R” estão mais próximas entre si do que as do 5.

Sendo assim, analisando o desempenho geral e das classes dos modelos obtidos em cada experimento, observa-se que o experimento 6 teve bons resultados, destacando a classe “R”, que obteve resultados superiores comparados aos do 5. Tendo em vista que o objetivo em ambos os experimentos é melhorar sua classificação na classe “R”, devido as analises realizadas, determina-se que o desempenho dos classificadores será analisado por meio do experimento 6.

5.2.4 Análise do desempenho do adaboost.M1 no experimento 6

A tabela 5 mostra os resultados de acurácia e coeficiente Kappa obtidos por *adaboost.M1* utilizando diferentes algoritmos bases. Pode-se observar que as taxas mais altas são obtidas ao utilizar *random forest*, que possui o maior percentual de acurácia, chegando a 93,51% e *J48* com 92,86%, para o coeficiente Kappa os algoritmos são os mesmos, chegando a 0,8701 e 0,8571, respectivamente.

Tabela 5 - Acurácia e coenficente Kappa de *adaboost.M1*.

Medidas	**Decision Stump**	**Hoeffding Tree**	**Random Tree**	**REP Tree**	**J48**	**Random Forest**
Acurácia	82,59%	73,57%	92,52%	92,78%	92,86%	93,51%
Kappa	0,6519	0,4712	0,8503	0,8556	0,8571	0,8701
Nº total de instâncias	6761	6761	6761	6761	6761	6761

Fonte: Do autor.

A fim de analisar se as diferenças entre os resultados são estatisticamente significativas aplicou-se o teste de significância nas taxas de acurácia, conforme tabela 6. A comparação dos resultados é realizada sempre entre o primeiro algoritmo, caso o valor seja maior significativamente deve ser marcado “v” na linha nomeada de “marcação”, se for menor é expressado por “*” e nos casos que não possuem diferenças não são realizadas marcações.

A tabela 6 mostra os percentuais de acurácia, o desvio padrão e a marcação, caso haja ou não diferenças significativas, ao executar 100 vezes.

Tabela 6 - Teste de significância estatística da acurácia de *Adaboost.M1*.

Exp. 6	**Random Forest**	**J48**	**Hoeffding Tree**	**Decision Stump**	**REP Tree**	**Random Tree**
Acurácia	93,51	92,93	71,51	82,53	92,68	92,45
Desvio Padrão	0,89	0,93	7,99	1,31	0,88	0,94
Marcação		*	*	*	*	*

Fonte: Do autor.

Pode-se observar que *random forest* possui o percentual de acurácia maior, com 93,51%, a tabela 6 mostra que há uma diferença maior estatisticamente significativa para esse algoritmo, considerando o nível de significância de 5%.

Com o intuito de analisar o desempenho das classes, tendo em vista que foi utilizada a técnica para balanceamento da classe “R”, a tabela 7 mostra os valores que cada algoritmo obteve por classe, em termos de taxa de verdadeiro positivo e *F-Measure*.

Tabela 7 - Valores de *F-Measure* e *TP-Rate* por classe de *adaboost.M1*.

Medida	**Classe**	**Decision Stump**	**Hoeffding Tree**	**Random Tree**	**REP Tree**	**J48**	**Random Forest**
TP-Rate	A	0,74	0,868	0,941	0,938	0,937	0,957
	R	0,912	0,603	0,909	0,918	0,92	0,913
	média	*0,826*	*0,736*	*0,925*	*0,928*	*0,929*	*0,935*
F-Measure	A	0,81	0,767	0,926	0,929	0,929	0,937
	R	0,84	0,695	0,924	0,927	0,928	0,934
	média	*0,825*	*0,731*	*0,925*	*0,928*	*0,929*	*0,935*

Fonte: Do autor.

Os resultados mais expressivos para a classe “A”, em termos de taxas de verdadeiros positivos, são de *random forest* com 0,957 e *random tree* com 0,941. Na medida *F-Measure* o *random forest* também atinge o valor mais alto, com 0,937, *J48* e *REP tree* alcançam 0,929.

Na classe “R”, pode-se observar que, em termos de taxas de verdadeiros positivos, *J48* possui 0,92 e *REP tree* 0,918, já na medida *F-Measure* destacam-se *random forest* com 0,934 e *J48* com 0,928.

Considerando o teste de significância aplicado no percentual de acurácia (tabela 6) e o coeficiente Kappa (tabela 5), pode-se observar que, *random forest* possui o modelo com o melhor desempenho geral, com diferença estatisticamente maior comparado aos demais. Ao analisar os resultados por classe, iniciando por “A”, em números absolutos, *random forest* possui os melhores resultados tanto para taxas de verdadeiros positivos quanto para *F-Measure*. Quanto à classe “R” apenas a taxa de verdadeiros positivos não é a melhor, no entanto possui um bom resultado, alcançando 0,913.

5.2.5 Análise do desempenho de random subspace no experimento 6

A tabela 8 mostra os resultados de acurácia e coeficiente Kappa obtidos por *random subspace* utilizando diferentes algoritmos bases. Pode-se observar que os classificadores com taxas mais altas são de *random forest*, que possui o maior percentual de acurácia, chegando a 93,94% e *REP tree* com 93,48%. Já para o coeficiente Kappa os algoritmos são os mesmos, chegando a 0,8787 e 0,8695, respectivamente.

Tabela 8 - Acurácia e Índice Kappa de *Random Subspace*.

Medidas	**Decision Stump**	**Hoeffding Tree**	**Random Tree**	**REP Tree**	**J48**	**Random Forest**
Acurácia	80,90%	81,82%	93,40%	93,48%	93,40%	93,94%
Kappa	0,6182	0,6364	0,8681	0,8695	0,8691	0,8787
Nº total de instâncias	6761	6761	6761	6761	6761	6761

Fonte: Do autor.

Pode-se observar que *random forest* possui o percentual de acurácia maior e ao comparar sua taxa com os demais algoritmos utilizados, percebe-se que só não há diferenças estaticamente significativa comparado ao percentual obtido por *REP tree*, conforme mostra a tabela 9. No entanto, em termos de números absolutos, *random forest* ainda possui o maior percentual.

Tabela 9 - Teste de significância estatística da acurácia *Random Subspace*.

Exp. 6	**Random Forest**	**J48**	**Hoeffding Tree**	**Decision Stump**	**REP Tree**	**Random Tree**
Acurácia	93,77	93,35	81,80	80,90	93,47	93,23
Desvio Padrão	0,89	0,96	4,76	2,7	0,92	0,97
Marcação		*	*	*		*

Fonte: Do autor.

Com o intuito de analisar o desempenho das classes, tendo em vista que foi utilizada a técnica para balanceamento da classe “R”, a tabela 10 mostra os

valores que cada algoritmo obteve por classe, em termos de taxa de verdadeiro positivo e *F-Measure*.

Tabela 10 - Valores de *F-Measure* e *TP-Rate* por classe *random subspace*.

Medida	**Classe**	**Decision Stump**	**Hoeffding Tree**	**Random Tree**	**REP Tree**	**J48**	**Random Forest**
TP-Rate	A	0,741	0,89	0,966	0,969	0,966	0,975
	R	0,877	0,747	0,902	0,901	0,902	0,904
	média	*0,809*	*0,818*	*0,934*	*0,935*	*0,934*	*0,939*
F-Measure	A	0,795	0,831	0,936	0,937	0,936	0,942
	R	0,821	0,804	0,932	0,932	0,932	0,937
	média	*0,808*	*0,817*	*0,934*	*0,935*	*0,934*	*0,939*

Fonte: Do autor.

Os resultados mais expressivos para a classe “A”, em termos de taxas de verdadeiros positivos são de *random forest* com 0,975 e *REP tree* com 0,969. Na medida *F-Measure* o *random forest* também atinge o valor mais alto, com 0,935 e *REP tree* com 0,937.

Na classe “R”, pode-se observar que, em termos de taxas de verdadeiros positivos, *random forest* possui 0,904, *REP tree* e *random tree* possuem 0,902, já na medida *F-Measure* destacam-se *random forest* com 0,937, *J48*, *REP tree* e *random tree* com 0,932.

Ao analisar o teste de significância aplicado no percentual de acurácia (tabela 9) pode-se observar que *random forest* não possui diferença estatisticamente significativa comparado ao *REP tree*, no entanto, considerando apenas números absolutos, *random forest* ainda possui o maior percentual de acurácia. Para o coeficiente Kappa, *random forest* também possui o melhor resultado, sendo assim, pode-se perceber que ao utiliza-lo como algoritmo base, seus modelos possuem o melhor desempenho geral.

Considerando os resultados por classe (tabela 10), tanto para a classe “A” quanto para “R”, em números absolutos, *random forest* possui os melhores resultados tanto para taxas de verdadeiros positivos quanto *para F-Measure*.

5.2.6 Comparação entre adaboost.M1 e random Subspace

A comparação dos modelos obtidos por *adaboost.M1 e random subspace* foi realizada utilizando o algoritmo de base *random forest* no experimento 6*,* tendo em vista que possuiu os melhores modelos, a fim de verificar qual dos dois possui melhor desempenho para o conjunto de atributos selecionados.

O teste de significância foi aplicado nos percentuais de acurácia, a tabela 11 mostra que não há diferenças estatisticamente significativa entre os resultados, apesar de que, em números absolutos, *random subspace* possui taxa superior ao *adaboost.M1*, chegando a 93,77%.

Tabela 11 - Teste de significância estatística da acurácia de a*daboost.M1* e *random subspace*.

Exp. 6	**Adaboost.M1 (Random Forest)**	**Random Subspace (Random Forest)**
Acurácia	93,51	93,77
Desvio Padrão	0,89	0,89
Marcação		

Fonte: Do autor.

A tabela 12 apresenta o desempenho por classe de cada algoritmo, quando se trata de taxas de verdadeiros positivos, *random subspace* possui percentual maior na classe “A”, já para a classe “R”, *adaboost.M1* obteve resultados superiores.

Tabela 12 - Valores de *TP-Rate* e *F-Measure* por classe de *adaboost.M1* e *random subspace*.

Medida	**Classe**	**Adaboost.M1 (Random Forest)**	**Random Subspace (Random Forest)**
TP-Rate	A	0,957	**0,975**
	R	**0,913**	0,904
	média	*0,935*	*0,939*
F-Measure	A	0,937	**0,942**
	R	0,934	**0,937**
	média	*0,935*	*0,939*

Fonte: Do autor.

Pode-se observar que na medida *F-Measure,* em ambas as classes, o *random subspace* atingiu resultados superiores comparados aos de *adaboost.M1*.

Apesar dos dois algoritmos não possuírem diferenças estatisticamente significativa nos percentuais de acurácia, o modelo obtido por *random subspace* utilizando *random forest* (93,77%) possui, em números absolutos, taxa superior a de *adaboost.M1* utilizando *random forest* (93,51%).

O modelo que apresentou melhores resultados foi obtido por *random subspace* utilizando *random forest* como algoritmo de base, vale ressaltar que neste algoritmo são selecionados, a cada iteração, um subconjunto de atributos do total disponível, de forma aleatória para realizar o treinamento. Os atributos selecionados são submetidos ao algoritmo de base *random forest* por mais 10 iterações, ao final é realizado o voto majoritário.

Em termos de interação com o ambiente, os atributos *nunca_acessou*, *dias_trans_criacao_primeiroacesso*, *qtdepostagem*, *perc_atv_realizadas* e *perc_quiz_realizado* foram os mais relevantes para identificar os alunos aprovados e reprovados neste modelo.

5.3 DISCUSSÃO DOS RESULTADOS

Em relação aos experimentos realizados, pode-se observar que os conjuntos de dados que não utilizam a técnica de balanceamento, mais precisamente 1 e 2, possuem altas taxas de verdadeiros positivos para classe “A” variando de 0,925 a 1, porém, quando se trata da classe “R”, são os experimentos que possuem as taxas menores, os resultados ficam entre 0,683 e 0,16, grande parte dos modelos gerados classificam, mais da metade, a classe “R” sendo “A”, neste sentido, a base com as classes desbalanceadas são melhores para classificar apenas o perfil de interação dos alunos aprovados.

Considerando que a classificação correta da classe “R” é importante, pois representam os alunos com desempenho inferiores, observou-se que, o experimento 6, utilizando a técnica *SMOTE* com percentual de sobreamostragem de 282%, ou seja, onde o número de instâncias das classes “A” e “R” possuem uma diferença de 9 registros, sobre o conjunto de dados original, notou-se que, as taxas de

verdadeiros positivos da classe “R” possuem resultados entre 0,603 e 0,92, muito superiores ao dos experimentos que não utilizam a técnica.

O estudo de Gottardo (2012) realiza experimentos em dados educacionais, utiliza a técnica *SMOTE* para balancear a classe minoritária, nos quais os alunos com desempenho inferior estavam em maior concentração. Foram aplicados os algoritmos *random forest* e *multilayer perceptron,* em termos de acurácia global, no experimento em que o conjunto de dados é original, o primeiro algoritmo obteve uma taxa de 77,4% e o segundo 80,1%, já ao aplica-los no conjunto de dados balanceados, *random forest* chegou a 78,4% e *multilayer perceptron* 77,1%.

Neste sentido, observa-se que comparando o resultado obtido por *random forest* no trabalho de Gottardo (2012) de 78,4%, com *random subspace* utilizando *random forest* com taxa de 93,77%, alcançados no experimento 6 deste trabalho, as diferenças são altas e o resultado obtido nesta pesquisa supera aos encontrados no trabalho em questão, validando a utilização da técnica SMOTE para dados educacionais em que as classes estão desbalanceadas.

Além disso, pode-se comparar o percentual de acurácia obtido por *random subspace* ao utilizar *random forest* no experimento 6, tendo em vista que foi a configuração e o experimento mais adequado para este algoritmo no conjunto de dados utilizados, de 93,77% com percentuais obtidos em trabalhos como de Malaise, Malibari e Alkhozae (2014) que chegou a 80% utilizando *adaboost.M1* em dados educacionais para previsão de desempenho; de Ayyappan e Kumar (2017), em que atingiu 72,18%, também aplicando *adaboost.M1* em dados educacionais; de Souza (2016), que obteve um percentual de 98,55% com *adaboost.M1* em uma base de dados com classe binária.

No trabalho de Santana, Maciel e Rodrigues (2014), foram utilizados sete algoritmos de classificação, *random forest*, *multilayer perceptron*, *naive bayes*, *SVM*, *KNN*, *J48* e *RBF* em dados educacionais, considerando atributos relacionados a perfil de uso do AVA. O método de estratificação utilizado foi *K-fold cross-validation* com 10 partições, assim como nesta pesquisa. Foram analisados os resultados da matriz de confusão e acurácia, os melhores resultados encontrados na pesquisa

foram ao utilizar duas classes, aprovado e reprovado, com o classificador *J48*, chegando ao percentual de acurácia de 74,68%.

Observa-se que ao comparar os percentuais obtidos em cada estudo, em números absolutos, o percentual atingido nesta pesquisa só não supera ao encontrado no trabalho de Souza (2016), no entanto, a utilização do algoritmo *random subspace* com *random forest*

6 CONCLUSÃO

O acompanhamento do desempenho de estudantes em cursos e disciplinas ofertados na modalidade a distância tem sido amplamente explorado pela comunidade cientifica, a fim de auxiliar e buscar soluções que facilitem a compressão de diversos problemas pedagógicos.

A *EDM* é uma subárea de *data mining* que possui técnicas para realizar inferências em dados educacionais, a classificação é uma de suas principais tarefas, muito utilizada para verificar predições de desempenho e perfis de interações de alunos em ambientes virtuais.

A fim de se obter dados mais precisos, foram utilizados os algoritmos *adaboost.m1* e *random subspace*, que tem como principio o método de comitê de classificadores.

Para aplicação dos algoritmos foi necessário investigar como os dados estão armazenados na plataforma Moodle da UNESC, juntamente com o delineamento da disciplina utilizada na pesquisa, Metodologia Cientifica e da Pesquisa, para verificar quais atributos poderiam ser utilizados nos três perfis de interações de Moore (aluno-ambiente, aluno-professor e aluno-aluno).

Nesta etapa foram encontradas dificuldades como: entendimento do banco de dados do Moodle, devido ao fato de possuir muitas tabelas e terem poucos estudos que relatam de forma minuciosa a extração do conjunto de atributos de um banco de dado relacional, nesta situação, foram gerados diversos *scripts* para capturar diferentes atributos e realizada reuniões com o setor de EaD com intuito de melhor entender como os dados estavam estruturados e quais atributos seriam relevante; pela disciplina ser ofertada em cursos presenciais, as interações entre alunos e professores não ocorrem exclusivamente no ambiente virtual, essas dificuldades foram resolvidas deixando apenas a interação entre aluno-ambiente.

O pré-processamento é uma etapa extensa e requer bastante atenção, pois interfere diretamente na qualidade dos modelos encontrados, foram encontradas dificuldades relacionadas à quais técnicas empregar, sendo resolvido estudando de forma mais detalhada a natureza dos dados e trabalhos que estivessem relacionados a esta pesquisa.

Apesar das dificuldades, os resultados encontrados são satisfatórios e atingem os objetivos da pesquisa, possibilitando a identificação do algoritmo de comitê de classificação que possui melhor desempenho usando medidas de qualidade em classificação.

Após serem realizadas diversas análises comparativas entre os experimentos utilizados nesta pesquisa, o conjunto de dados mais apropriado para classificação correta tanto de alunos aprovados quanto reprovados, foram os dados originais, com apenas as classes discretizadas e a técnica SMOTE com percentual de sobreamostragem de 282%, denominado como experimento 6, nesse sentido, o algoritmo que obteve melhor desempenho neste conjunto de dados foi *random subspace* utilizando *random forest* como classificador de base, alcançando um percentual de acurácia de 93,77%.

Considerando os resultados obtidos nesta pesquisa, destacam-se algumas sugestões para trabalhos futuros:

a) aplicar o mesmo conjunto de atributos em classificadores únicos a fim de comparar os resultados com os obtidos por algoritmos de comitê de classificação;

b) empregar em outras disciplinas da modalidade a distância que possuem outras organizações para analisar os modelos gerados;

c) adotar as três interações por Moore em disciplinas que possuem interações de aluno-aluno e aluno-professor em ambientes virtuais;

e) utilizar outras medidas de qualidade para analisar o desempenho por classe como matriz de confusão e curva ROC.

7 REFERÊNCIAS

ABED - Associação Brasileira de Educação a Distância. **Censo EAD Brasil 2016. 2017**. Disponível em<http://www.abed.org.br/site/pt/midiateca/censo_ead/1449/2017/09/censoead.br_-_2016/2017>. Acesso em: 20 set. 2017.

ABED - Associação Brasileira de Educação a Distância. **Censo EAD Brasil 2017. 2018**. Disponível em<http://www.abed.org.br/site/pt/midiateca/censo_ead/1554/2018/10/censoeadbr_-_2017/2018>. Acesso em: 10 jun. 2019.

APOLLONI, B.; VALENTINI, G.; BREGA, A. Bica and random subspace ensembles for dna microarray-based diagnosis.In: INTERNATIONAL FLINS CONFERENCE, 2006, Itália. **Applied Artificial Intelligence.** Disponível em<**https://doi.org/10.1142/9789812774118_0088**>. Acesso em: 09 jun. 2018.

ARFFIN, H. M. Nor et al. Assessment of the students' utilization of a learning management system in a malaysian higher education. In: **IEEE conference on e-learning, e-management and e-services** , 2015, Hawthorn. Disponível em <https://ieeexplore.ieee.org/document/7081235>. Acesso em: 16 jun. 2018.

ASSIS, Izabela Mendonça de; ARAÚJO, Fernando Costa; SOUSA, Walter Lopes de. Educação à distância no Brasil: um estudo sobre perspectivas e desafios do ensino em ambiente virtual. Revista EM FOCO - Fundação Esperança/IESPES, [S.l.], v. 1, n. 27, p. 88-102, fev. 2018. ISSN 2319-037x. Disponível em: <http://revistaemfoco.iespes.edu.br/index.php/Foco/article/view/173>. Acesso em: 08 Jul. 2019.

AYYAPPAN, G.; KUMAR, S. K. A novel approach of ensemble models using edm. **Indian Journal of Computer Science and Engineering,** vol. 8, no. 6, 2017. Disponível em< **http://www.ijcse.com/ijcse-issue.html?issue=20170806**>. Acesso em: 23 jun. 2018.

BAKER, R.S.J.D.; ISOTANI, S; CARVALHO, A.M.J.B.D. Mineração de dados educacionais: oportunidades para o brasil. **Revista Brasileira de Informática na Educação**, vol. 19, no. 2, p. 2-3, 2011.

BUNIYAMIN, N; MAT, B. U; ARCHAD, M. P**.** Educational data mining for prediction and classification of engineering students achievement. In: **IEEE International conference on engineering education**, 7., 2015, Kanazawa. Disponível em< **https://ieeexplore.ieee.org/document/7451491/**>. Acesso em: 24 jun. 2018.

C, Lakshmi Devasena. Comparative Analysis of Random Forest, REP Tree and J48 Classifiers for Credit Risk Prediction. International Journal Of Computer Applications. India, p. 30-36. mar. 2014. Disponível em: <https://www.ijcaonline.org/proceedings/icccmit2014/number3/19785-7033>. Acesso em: 01 jun. 2019.

CECHINEL, Cristian; CAMARGO, Sandro da Silva. Mineração de dados educacionais: avaliação e interpretação de modelos de classificação. In: JAQUES, Patrícia Augustin; PIMENTEL, Mariano; SIQUEIRA; Sean; BITTENCOURT, Ig. (Org.) Metodologia de Pesquisa em Informática na Educação: Abordagem Quantitativa de Pesquisa. Porto Alegre: SBC, 2019. (Série Metodologia de Pesquisa em Informática na Educação, v. 2) Disponível em: <http://metodologia.ceie-br.org/livro-2>. Acesso em: 30/04/2019

CHANDOLA, Varun; KUMAR, Vipin. Summarization compressing data into an informative representation. In: IEEE INTERNATIONAL CONFERENCE IN DATA MINING, 5., 2005, Houston. **Confêrencia**. Disponível em <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1565667&isnumber=33217> Acesso em: 13 abr. 2018.

CHAWLA, N. V. et al (2002). SMOTE: Synthetic minority over-sampling technique. J. Artif. Intell. Res. (JAIR), 16, 321–357. Chawla, N. V., Japkowicz, N., e Kotcz, A. (2002). Editorial: special issue on learning from imbalanced data sets. SIGKDD Explorations, 6(1), 1–6.

COELHO, Guilherme Palermo. **Geração, Seleção e Combinação de Componentes para ializaçãoEnsembles de Redes Neurais Aplicadas a Problemas de Classificação.** 2006. 115 f. Monografia (Especialização) - Curso de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas Faculdade de Engenharia Elétrica e de Computação, Campinas, 2006.

COSTA, Evandro et al. Mineração de Dados Educacionais: Conceitos, Técnicas, Ferramentas e Aplicações. Jornada de Atualização em Informática na Educação (jaie 2012), Porto Alegre, v. 1, n. 1, p.1-29, nov. 2012. Disponível em: <http://www.br-ie.org/pub/index.php/pie/article/view/2341/2096>. Acesso em: 01 abr. 2019.

DUDA, O. Richard; HART, E. Peter; STORK, G. David. **Pattern classification**. 2. ed. New York: John Wiley & Sons. 2001.

DIETTERIC, G. Thomas. Ensemble methods in machine learning. In: INTERNATIONAL WORKSHOP ON MULTIPLE CLASSIFIER SYSTEMS. 2000, Itália. Multiple Classifier Systems.Disponível em: <https://doi.org/10.1007/3-540-45014-9_1> Acesso em: 21 fev. 2018.

DIETTERIC, G. Thomas; BAKIRI, Ghulum. Solving multiclass learning problems via error-correcting output codes. **Journal of Artificial Intelligence Research**. jan. 1994. vol 2. Disponível em: <https://jair.org/index.php/jair/article/view/10127> Acesso em: 14 mai. 2018.

DEEPASHRI, K. S; KAMATH, A. Survey on techniques of data mining and its applications. INTERNATIONAL CONFERENCE ON EMERGING TREND IN ENGINEERING, 2., 2017, India. **Conference Held at Hotel Magaji Orchid,** Disponível em<

https://pdfs.semanticscholar.org/b738/3df4705133a132f58104b514b80555fe78cb.pdf >. Acesso em: 23 jun. 2018.

DUTT, A.; ISMAIL, A. M.; HERAWAN, T. A systematic review on educational data mining. **IEEE Acess,** vol. 5, p. 15991-16005, 2017. Disponível em <https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=6287639>. Acesso em:03 jun. 2018.

EIBL, Günther; PFEIFFER, P. Karl. How to make adaboost.m1 work for weak base classifiers by changing only one line of the code. **Lecture Notes in Computer Science**, Heidelberg, set. 2002, vol 2430. Disponível em <https://link.springer.com/chapter/10.1007/3-540-36755-1_7> Acesso em: 20 mai. 2018.

FAYYAD, Usama; UTHURUSAMY, Ramasamy. Evolving data mining into solutions for insights**. Comunications of the ACM.** 2002, vol 45. Disponível em < https://dl.acm.org/citation.cfm?id=545174 > Acesso em: 30 mar. 2018.

FAYYAD, Usama; PIATETSKY-SHAPIRO, Gregory; SMYTH, Padhraic. The kdd process for extracting useful knowledge from volumes of data**. ACM.** New York, nov. 1996. vol 39. Disponível em <http://dx.doi.org/10.1145/240455.240464> Acesso em: 30 mar. 2018.

FRANÇA, M. G. D. **Comparação entre classificações de cobertura de solo urbano derivados do wv-2 quanto ao nível de legenda de classificação: estudo de caso para um setor da unicamp, sp**. 2017, 192 f. Dissertação de mestrado (Curso de pós-graduação em sensoriamento remoto),Instituto nacional de pesquisas espaciais. Disponível em <http://urlib.net/8JMKD3MGP3W34P/3MAPN7E>. Acesso em: 23 jun. 2018.

FRANK, Eibe et al. **Data Mining: Pratical Machine Learning Tools and Techniques**. 4 ed. San Francisco: Elsevier. 2017.

FREUND, Yoav. The strength of weak learnability. **Kluwer Academic Publishers-Plenum Publishers**. Boston, jun. 1990. vol 5. Disponível em <https://doi.org/10.1023/A:1022648800760> Acesso em: 30 mar. 2018.

FREUND, Yoav.; SCHAPIRE, E. Robert. Experiments with a New Boosting Algorithm. **Machine Learning: Proceedings of the Thirteenth International Conference**, São Francisco, 1996. Morgan Kaufmann Publishers Inc. p. 148-156.

FREUND, Yoav.; SCHAPIRE, E. Robert. **Boosting:** foundations and algorithms. United States of America: MIT Press. 2012.

FÜRNKRANZ, Johannes. Decision Stump. Encyclopedia Of Machine Learning And Data Mining, [s.l.], p.1-1, 2016. Springer US. http://dx.doi.org/10.1007/978-1-4899-7502-7_285-1.

GIACOMAZZO, Graziela Fatima et al. Disciplina institucional a distância: processo de implantação numa universidade comunitária. Intersaberes, São Paulo, v. 13, n. 29, p.240-250, 2018. Disponível em: <https://www.uninter.com/intersaberes/index.php/revista/article/view/1415>. Acesso
em: 11 jun. 2019.

GOLDSCHMIDT, Ronaldo; Emmanuel, PASSOS; BEZERRA, Eduardo. **Data mining: conceitos, técnicas, algoritmos, orientações e aplicações.** 2. ed. Rio de Janeiro: Elsevier. 2015.

GOTTARDO, Ernani. **ESTIMATIVA DE DESEMPENHO ACADÊMICO DE ESTUDANTES EM UM AVA UTILIZANDO TÉCNICAS DE MINERAÇÃO DE DADOS.** 2012. 84 f. Dissertação (Mestrado) - Curso de Programa de pos-graduacapo em Computacao Aplicada, Universidade Tecnologica Federal do Parana, Curitiba, 2012

GOTTARDO, E.; KAESTNER, C.; NORONHA, R. V. A**plicação de Técnicas de Mineração de Dados para Estimativa de Desempenho Acadêmico de Estudantes em um AVA Utilizando Dados com Classes Desbalanceadas**. Em: ICBL2013–International Conference on Interactive Computer aided Blended Learning. 2013.

GORELICK, M. H.; YEN, K. The kappa statistic was representative of empirically observed inter-rater agreement for physical findings. Journal of Clinical Epidemiology, v. 59, n. 8, p. 859-861, aug. 2006. Disponível em: < http://www.jclinepi.com/article/S0895- 4356%2806%2900024-2/abstract>. Acesso em: 04. Jun, 2019.

HAN, Jiawei; KAMBER, Micheline; PEI, Jian. **Data mining:** concepts and techniques. 3. ed. United States of America: Elsevier. 2012.

HASTIE, Trevor; TIBSHINARI, Robert; FRIEDMAN, Jerome. **The elements of statistical learning:** data mining, inference, and prediction. 2. ed. New York: Springer. 2009.

HENKE, et al. Detecção de intrusos usando conjunto de k-nn gerado por subespaços aleatórios. In: **Simpósio brasileiro de segurança da informação**,6 , 2012

HULTEN, Geoff; SPENCER, Laurie; DOMINGOS, Pedro. Mining time-changing data streams. Proceedings Of The Seventh Acm Sigkdd International Conference On Knowledge Discovery And Data Mining - Kdd '01, [s.l.], p.1-10, 2001. ACM Press. http://dx.doi.org/10.1145/502512.502529.

INEP - Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira. **Censo da Educação Superior 2016**. 2017. Disponível em<

http://download.inep.gov.br/educacao_superior/censo_superior/documentos/2016/notas_sobre_o_censo_da_educacao_superior_2016.pdf>. Acesso em: 20 set. 2017.

KALMEGH, Sushilkumar Rameshpant. Comparative Analysis of WEKA Data Mining Algorithm RandomForest, RandomTree and LADTree for Classification of Indigenous News Data. International Journal Of Emerging Technology And Advanced Engineering. Madhya Pradesh, p. 507-517. jan. 2015. Disponível em: <https://pdfs.semanticscholar.org/d334/ec05c5bafed96ff40bb6ca219253e0f7040d.pdf>. Acesso em: 09 jun. 2019.

KANTARDIZIC, Mehmed. **Data mining: concepts, models, methods, and algorithms.** 2. ed. Hoboken: Wiley & Sons, 2011.

KEARSLEY, Greg; MOORE, Michael. **Educação a distância: Uma visão integrada**. São Paulo: Cengage learning, 2007.

KUMAR, Vipin; TAN, Pang-Ning; STEINBACH, Michael. **Introdução a Data Mining: mineração de dados**. Rio de Janeiro: Ciência Moderna, 2009.

KUNCHEVA, I. Ludimila. **Combining Pattern Classifiers: Methods and Algorithms**. Hoboken: Wiley & Sons, 2004.

KUNCHEVA, et al. **Ransom subspace ensembles for fMRI classification**.Em: IEEE Trans Med Imaging. 2010.

LAROSE,T. Daniel; LAROSE, D. Chantal. **Discovering knowledge in data: An introduction to data mining**. Wiley & Sons, Inc., Hoboken, New Jersey. 2014. p. 10-15.

LAROSE, T. Daniel; LAROSE, D. Chantal. **Data mining and predictive analytics**. Hoboken: Wiley & Sons, Inc.. 2015.

LI, et al. Random subspace method for source camera identification. In:IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, 2015, BOSTON. **Workshop**. Disponível em < http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7324339&isnumber=7324166>. Acesso em: 24 jun. 2018.

LITTO, M. Fredric. O atual. In: LITTO, Fredric M.; FORMIGA, Marcos (Org.). **Educação a Distância o estado da arte**. São Paulo: Pearson, 2009. Cap. 13. p. 15-20. Disponível em: <http://www.abed.org.br/arquivos/Estado_da_Arte_1.pdf>. Acesso em: 24 jun. 2018.

LITTO, M. F.; FORMIGA, M. **Educação a distancia: o estado da arte**. São Paulo: Pearson. 2009.

MADNI, A. Hussain; ANWAR, Zahid; SHAH, A. Munam. Data mining techniques and applications – a decade review**. 23rd International Conference on Automation and**

Computing, Huddersfield, set. 2017. Disponível em <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8082090&isnumber=8081956> Acesso em: 17 mar. 2018.

MALAISE, A. A.; MALIBARI, A.; ALKHOZAE, M. Student's performance prediction system using multi agent data mining technique. In: INTERNATIONAL JOURNAL OF DATA MINING & KNOWLEDGE MANAGEMENT PROCESS, 5., 2014. **Journal IJDKP**.

MATTAR, João. Interatividade e aprendizagem. In: LITTO, Fredric M.; FORMIGA, Marcos (Org.). **Educação a Distância o estado da arte**. São Paulo: Pearson, 2009. Cap. 16. p. 112-120. Disponível em: <http://www.abed.org.br/arquivos/Estado_da_Arte_1.pdf>. Acesso em: 24 jun. 2018.

MERT, A.; KILIÇ, N.; BILGILI, E. Subspace method with class separability weighting **The Journal of Knowledge Engineering**, vol. 33, no 3, 2016, p.275-285,New York. Disponível em<https://onlinelibrary.wiley.com/doi/pdf/10.1111/exsy.12149>. Acesso em: 26 mai. 2018.

LEITE, Eliana Alves Moreira et al. FORMAÇÃO DE TUTORES NA DEAD/IFCE: COMPETÊNCIAS E HABILIDADES NECESSÁRIAS. **Conexões - Ciência e Tecnologia**, [S.l.], v. 10, n. 2, p. 53-60, nov. 2015. ISSN 2176-0144. Disponível em: <http://conexoes.ifce.edu.br/index.php/conexoes/article/view/725/763>. Acesso em: 07 june 2019. doi: https://doi.org/10.21439/conexoes.v10i2.725.

MOORE, G. M. Three types of interaction. **American Journal of Distance Education,** vol. 3, no. 2, p. 1-7, 1989. Disponível em < https://www.researchgate.net/publication/237404371_Three_Types_of_Interaction?enrichId=rgreq-d2a5121a7ec939af51c5ce55805d5481-XXX&enrichSource=Y292ZXJQYWdlOzIzNzQwNDM3MTtBUzoxMDY1ODUwMDMxMzQ5NzhAMTQwMjQyMzI1Mzc3Ng%3D%3D&el=1_x_3&_esc=publicationCoverPdf >. Acesso em: 21 jun. 2018.

MORAIS, M. Alana; FECHINE, Joseana. **Mineração de Dados Educacionais no Apoio ao Processo de Tomada de Decisão do Docente**. Em: Congresso da Sociedade Brasileira de Computação. 2013. p. 23-26.

NUNES, B. Ivônio. A história da ead no mundo. In: LITTO, Fredric M.; FORMIGA, Marcos (Org.). **Educação a Distância o estado da arte**. São Paulo: Pearson, 2009. Cap. 1. p. 2-8. Disponível em: <http://www.abed.org.br/arquivos/Estado_da_Arte_1.pdf>. Acesso em: 24 jun. 2018.

JÚNIOR, O. G. José. IDENTIFICAÇÃO DE PADRÕES PARA A ANÁLISE DA EVASÃO EM CURSOS DE GRADUAÇÃO USANDO MINERAÇÃO DE DADOS EDUCACIONAIS. 2015. 86 f. Dissertação (Mestrado) - Curso de Programa de PÓs-graduaÇÃo em ComputaÇÃo Aplicada, Universidade TecnolÓgica Federal do ParanÁ, Curitiba, 2015. Disponível em: <http://repositorio.utfpr.edu.br/jspui/bitstream/1/1995/1/CT_PPGCA_M_Oliveira%20J

unior%2C%20Jos%C3%A9%20Gon%C3%A7alves_2015.pdf>. Acesso em: 10 jun. 2019.

PATHICAL, S. **Classification in High Dimensional Feature Spaces through Random Subspace Ensembles**. 2010, 233 f. Tese (Mestrado em Ciências licenciatura em engenharia), The University of Toleto.

PEÑA-AYALA. **Educational Data Mining: Applications and Trends**. Springer. 2013.

POLIKAR, Robi. Ensemble based systems in decision machine. **IEEE circuits and systems magazine**, set. 2006, vol 6. Disponível em < http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1688199&isnumber=35619> Acesso em 10 fev. 2018.

RABELO, Humberto et al. **Utilização de técnicas de mineração de dados educacionais para predição de desempenho de alunos de EAD em ambientes virtuais de aprendizagem**. Em: Simpósio Brasileiro de Informática na Educação-SBIE. 2017. p. 1527.

RAMESH, V.; PARKAVI, P.; RAMAR, K. Predicting student performance: a statistical and data mining approach. In: INTERNATIONAL JOURNAL OF COMPUTER APPLICATIONS, 8., 2013. **IJCA Journal.** Disponível em < https://www.ijcaonline.org/archives/volume63/number8/10489-5242>. Acesso em: 02 jun. 2018.

RAMOS, T, et al. Uso de mineração de dados educacionais para identificação de perfis e padrões de participação dos estudantes de cursos a distância. In:**Iberian Conference on Information Systems and Technologies**, 12., 2017, Lisboa. Disponível em <https://ieeexplore.ieee.org/document/7975960/ >. Acesso em: 15 abr. 2018.

REFAAT, Mamdouh. **Data preparation for data mining using sas**. Morgan Kaufmann: São Francisco. 2006.

SANTANA, C. Leandro; MACIEL, MA. Alexandre; RODRIGUES, L. Rodrigo. **avaliação do perfil de uso no ambiente moodle utilizando técnicas de mineração de dados**. Em: Simpósio Brasileiro de Informática na Educação-SBIE. 2014. p. 269-277.

SANTANA, C. Leandro; MACIEL, M. Alexandre; RODRIGUES,L. Rodrigo. **Integração de um mecanismo de Mineração de Dados Educacionais ao Moodle**. Em: Anais dos Workshops do Congresso Brasileiro de Informática na Educação. 2017. p. 664.

SILVA, M. J., Noeme; FERREIRA, A., Diogo. O poder judiciário e a ead. In: LITTO, Fredric M.; FORMIGA, Marcos (Org.). **Educação a Distância o estado da arte**. São

Paulo: Pearson, 2012. Cap. 28. p. 234-240. Disponível em: <http://www.abed.org.br/arquivos/Estado_da_Arte_2.pdf>. Acesso em: 23 jun. 2018.

SILVA, S. Robson. **Gestão de ead: educação a distância na era digital**. São Paulo: Novatec. 2013.

SKURICHINA, Marina. DUIN, W. P., Robert. Bagging, boosting and the random subspace method for linear classifiers. Anal PPA. 2002. Springer. Acesso em <https://link.springer.com/article/10.1007/s100440200011 11> Acesso em 12 mai. 2018.

SOKOLOVA, M.; LAPALME, G. A systematic analysis of performance measures for classification tasks. **Information Processing & Management,** 2009, vol. 45, no. 4, p. 427-437.Disponível em < https://www.sciencedirect.com/science/article/pii/S0306457309000259>. Acesso em: 04 jun. 2018.

ZAQUI, J. Mohammed. Parallel and distributed data mining: an introduction**. Lecture Notes in Computer Scienc**, Heidelberg, mai. 2002, vol 1759. Disponível em <https://doi.org/10.1007/3-540-46502-2_1> Acesso em: 17 fev. 2018.

ZHANG, Cha; MA, Yunqian. **Ensemble Machine Learning: methods and applications.** New York: Springer. 2012.

WALSE, Kishor H; DHARASKAR, Rajiv V; THAKARE, Vilas M. A STUDY OF HUMAN ACTIVITY RECOGNITION USING ADABOOST CLASSIFIERS ON WISDM DATASET. Iioab Journal. India, p. 68-76. 68 jan. 2016. Disponível em: <https://www.researchgate.net/publication/324262009_A_study_of_human_activity_recognition_using_adaboost_classifiers_on_WISDM_dataset>. Acesso em: 02 jun. 2019.

WAGNER, Mario B; MOTTA, Valter T.; DONELLES, Cristina. SPSS passo a passo: statisticlal package for the social sciences. Caxias do Sul: Educs, 2004. 172 p.

WITTEN, H. Ian; FRANK, Eibe. **Data mining: practical machine learning tools and techniques**. 2. ed. San Francisco: Elsevier. 2005.

APÊNDICE(S)

APÊNDICE A – Artigo

Modelo de predição utilizando comitê de classificadores para identificação de perfis de interação no ambiente virtual de aprendizagem

Laíne Dimer[1], Merisandra Cortês de Mattos Garcia [1]

[1]Curso de Bacharelado em Ciência da Computação – Universidade do Extremo Sul Catarinense (UNESC) – Criciúma – SC – Brazil

Laainediimer@gmail.com, mem @unesc.net

Abstract. *The purpose of this research is to apply Educational Data Mining in a database of the virtual learning environment, seeking to define the profiles of students based on the profiles of interaction studied by Moore. For this, the data will be analyzed by means of the classification task using the algorithms of classification of the type of boosting, Adaboost.M1 and Random Subspace, comparing them by means of measures of quality in classification in order to identify the model that presents improved results.*

Resumo. *Esta pesquisa consiste na aplicação do Educational Data Mining em uma base de dados de um ambiente virtual de aprendizagem, procurando definir os perfis dos alunos baseando-se nos perfis de interação estudados por Moore. Para isso, os dados serão analisados por meio da tarefa de classificação utilizando os algoritmos de comitê de classificação do tipo boosting, Adaboost.M1 e Random Subspace, comparando-os por meio de medidas de qualidade em classificação a fim de identificar o modelo que apresenta melhores resultados.*

1. Introdução

A descoberta de conhecimento em base de dados, do inglês *Knowledge Discovery in Database* (KDD), é constituída por etapas para identificação de modelos válidos e padrões. A etapa de *data mining* destaca-se por ser capaz de transformar grandes volumes de dados em conhecimentos específicos para sociedade. Esses conhecimentos adquiridos são utilizados para análise de mercado, produção, negócios, entre outros [Han e Kamber 2001].

Com o aumento da procura por padrões e perfis dos aprendizes em ambientes virtuais de aprendizagem, surgiu a mineração de dados educacionais, do inglês *Educational Data Mining* (EDM), em que algoritmos de *data mining* são adaptados para melhores resultados na busca por conhecimentos nas bases de dados educacionais [Baker, Isotani e Carvalho 2011].

Diversas técnicas são utilizadas em EDM, e uma delas é a predição [Baker, Isotani e Carvalho 2011]. A predição consiste na classificação dos dados, expressada por uma variável alvo, em que no processo de *data mining* os registros serão examinados, e que cada um deles contém informações sobre a variável alvo, aprendendo-se a relação entre os registros observados [Larose e Larose 2014].

A fim de se obter resultados mais precisos, ao invés de utilizar apenas um classificador, utiliza-se a combinação deles. Esse método é chamado de comitê de classificadores, meta classificadores ou *ensembles*, o qual divide os dados em partes menores e mais fáceis de aprender, cada classificador fica responsável por uma partição específica, e ao final é calculada a média ou votação das saídas de cada um [Polikar 2006].

O comitê de classificadores pode ser do tipo *boosting*, o qual atribui pesos a todos os exemplos de treinamento. Dentre os algoritmos do tipo *boosting*, tem-se o *adaboost.M1*, que refaz a reamostragem se o classificador base não puder lidar com as instâncias ponderadas. Outro método é o *random subspace*, que cria um conjunto de classificadores, em que cada um é treinado usando um subconjunto de características selecionadas aleatoriamente do espaço de recurso disponível [Frank et al 2017].

2. Materiais e Métodos

O software escolhido para a execução de data mining foi Weka por ser uma ferramenta que possui diversos métodos de data mining, bem como medidas de qualidade em classificação e por ser um software livre. Para isso, foi aplicado os algoritmos *adaboost.M1* e *random susbpace* e analisado por meio das medidas de qualidades escolhidas para identificar o modelo com melhores resultados**.**

2.1. Conjunto de dados

Para a realização do processo data mining sobre os perfis de interação dos alunos é necessário selecionar um conjunto de atributos que represente os alunos em um Ambiente Virtual de Aprendizagem (AVA). Neste trabalho utilizaram-se dados baseados nas interações estudadas por Moore, tais como: Aluno-ambiente, aluno-aluno e aluno-professor.

A base de dados utilizada nesta pesquisa é proveniente da plataforma Moodle da UNESC, foi selecionada a disciplina de Metodologia Científica e da Pesquisa (MCP) na modalidade a distância, sendo a primeira a ser ofertada de forma institucional a partir do primeiro semestre de 2017, os atributos selecionados foram escolhidos baseando-se nos estudos de Gottardo, Ernani e Noronha (2013), Santana, Maciel e Rodrigues (2014) e Ramos et al. (2017), totalizam 39 atributos, em que seis são para identificar a base de dados e os alunos, 15 se enquadram nas interações de aluno e ambiente, 14 em aluno e professor, três estão associadas a aluno e aluno e a classe, identificada pela nota final. O conjunto de dados conta com 4320 registros.

Para extração do conjunto de atributos selecionados foram utilizadas 18 tabelas do Moodle, as quais foram autorizadas pelo Setor de Educação à Distância da UNESC, disponibilizadas pelo setor de Tecnologia da Informação da universidade, em um script na linguagem *Structured Query Language* (*SQL*). Os dados foram disponibilizados às cegas, ou seja, sem informações pessoais ou código de usuários (alunos, professores, tutores e etc.) bem como conteúdo de mensagens, quiz, fórum ou outra atividade realizada na plataforma.

2.2. Pré-processamento

O pré-processamento consiste em várias técnicas para preparação do conjunto de dados para os algoritmos de data mining, nesta pesquisa foram utilizadas as seguintes técnicas: discretização, seleção de atributos para redução de dimensionalidade, tabelas de sumarização e transformação de dados. Para pré-processar o conjunto de atributos selecionados foram necessárias ferramentas como *Google Sheets*[9] e Weka.

[9] *Google Sheets* é uma ferramenta para criação de planilhas eletrônicas online, disponibilizada pela Google de forma gratuita.

2.3. Seleção de atributos

Para identificar se os atributos selecionados em cada interação são relevantes para a disciplina estudada, Metodologia da Cientifica e da Pesquisa, foram realizadas reuniões junto ao setor de Educação a Distância (EaD) da UNESC a fim de delinear a disciplina ofertada e remover atributos caso não fossem necessários para o perfil de interação.

Com isso, percebeu-se a necessidade de exclusão de alguns atributos, como: qtde_grupos, tendo em vista que a disciplina escolhida não agrupa alunos para realização de trabalhos; qtde_forum_assina, a recomendação é que não seja realizada a assinatura, pois toda a postagem relacionada ao fórum é encaminhada por e-mail para os assinantes; tentativa, removida, pois o número de tentativas permitidas para o quiz geralmente é uma; status_quiz e status_atividade foram removidos pelo fato de que se considerada para avaliação apenas o que está submetido.

Os atributos relacionados as mensagens também foram removidos do conjunto, notou-se que os alunos que repetiram a disciplina tinham a mesma quantidade de mensagens enviadas em ambas, isso porque, quando envia a mensagem pelo Moodle é considerado o contato direto entre os usuários, sem considerar a disciplina que estão.

Na exclusão de atributos de mensagens observou-se que as interações de alunos entre professores e alunos eram pouco frequente, devido ao fato da disciplina ser na modalidade a distância e ofertada em graduações presencias estas interações não ocorrem exclusivamente pelo AVA, tendo isso em vista, foi utilizado neste trabalho apenas a interação aluno-ambiente.

2.4. Tabelas de sumarização

O Moodle é um banco de dados relacional, constituído por diversas tabelas, a tabela de sumarização se faz necessária quando é preciso extrair o conjunto de dados selecionados, para uma única tabela. Foram criados scripts para cada atributo e exportado em .csv. Esses arquivos em .csv foram adicionados em uma única tabela.

2.5. Dervivação de novos atributos

A derivação de novos atributos é uma técnica utilizada a fim de potencializar o conjunto de dados existente, as novas características são baseadas nas já existentes. Foram realizadas duas derivações, tais como: dias_transcorridos foi originado da diferença entre data_criacao e primeiro_acesso, quando o primeiro acesso acontecia antes da criação do curso; e perc_atv_realizadas, perc_quiz_realizadas e perc_forum_realizadas foram originados dos atributos que constava a quantidade total disponível e a quantidade total realizada de atividade, quiz e fórum, como por exemplo, atv_realizadas e atv_total, foi calculada a relação entre os dois atributos para encontrar o percentual realizado.

Após finalizar a etapa de derivação de novos atributos o conjunto de atributos final é apresentado na tabela 1, sendo utilizado nos experimentos realizados, totalizando 4269 registros. Observa-se que os atributos foram reduzidos de 39 para 10 e as dimensões de quatro para duas em relação à tabela 1, apresentada anteriormente.

Dimensão	Atributo	Descrição
Identificação da base	id_disciplina	Código de identificação da disciplina no moodle
Aluno-Ambiente	dias_transcorridos	Dias transcorridos entre a data de criação do curso e o primeiro acesso
	nunca_acessou	Identificação para alunos que nunca acessaram o moodle (0 = nunca acessou, 1= acessou)
	qtde_acessos	Quantidade de acessos na disciplina do moodle
	perc_atv_realizadas	Percentual de atividades realizadas
	perc_forum_realizadas	Percentual de fóruns realizados
	perc_quiz_realizadas	Percentual de quiz realizadas
	qtdediscussoes	Quantidade de discussões abertas em fóruns por alunos
	qtdepostagem	Quantidade de postagens feitas em fóruns por alunos
Classe	resultado	Resultado obtido por meio da nota final do aluno. Representado por "A" para aprovado e "R" para reprovado

Tabela 1 - Conjunto de atributos final

2.6. Discretização

Para possibilitar a aplicação dos algoritmos, foi realizada a técnica de discretização na classe, os alunos foram divididos em duas categorias, a classe "A" representa os alunos aprovados, considera as notas igual ou acima de 6, já a "R" representa alunos reprovados e considera as notas abaixo de 6.

O procedimento de discretização também pode ser aplicado em todos os registros contínuos para reduzir a variação deles. Para isso, existem alternativas como a utilização de algoritmos de automação para realização desse método, no Weka está disponível como um filtro não supervisionado, nesta pesquisa foi aplicado esta técnica em apenas alguns experimentos realizados.

2.7. Balanceamento de classes

O desbalanceamento das classes pode influenciar o desempenho de um modelo de classificação, pois tentem a classificar corretamente somente as classes maioritárias, esse aspecto é definido por Chawla et al. (2002) como um conjunto de dados em que as classes não estão representadas de forma igual. O conjunto de dados utilizados possui na classe "A", aprovada, 3385 registros e em "R", reprovado, 884. A diferença entre os números de registros das classes é de 2501, sendo que a classe "A" é significamente maior, partindo do fato de que número de reprovados na disciplina é menor que os de aprovados. A técnica Synthetic Minority Over-sampling Technique (SMOTE) é uma alternativa para tratar classes desbalanceadas, amplamente utilizada em dados educacionais, presente em trabalhos como os de Júnior (2015) e Gottardo, Ernani e Noronha (2013).

Aplicou-se a técnica SMOTE disponível no Weka como um filtro supervisionado, nele é possível definir parâmetros ao aplicar esse método na ferramenta, como o percentual de sobreamostragem que foram usados 100% e 282% e o número de vizinhos, utilizado o valor 5, como sugerido pela Weka. Os percentuais foram aplicados em diferentes experimentos, ao ampliar na classe "R" o percentual de 100% obtém-se o dobro de registros, ou seja, do total, metade é constituído por exemplos reais e outra metade por exemplos sintéticos, neste caso os registros passa de 884 para 1768, reduzindo a diferença entre as classes para 1617. Ao ampliar

o percentual de 282% as classes “A” e “R” obtém-se classes com número de registros muito próximos, passando de 884 para 3376 exemplos, reduzindo a diferença entre as classes para 9.

2.8. Execução do Data Mining

Para execução do data mining foi utilizada a ferramenta Weka por possuir métodos para aplicação do data mining, bem como os algoritmos da tarefa de classificação utilizados nesta pesquisa e ser open source.

Ao empregar técnicas de data mining com algoritmos de classificação é necessário que a base de dados seja dividida em dois conjuntos: treinamento e testes, os modelos são obtidos por meio do conjunto de treinamento e logo são aplicados para classificar as instâncias separadas no conjunto de teste. Para estratificação dos conjuntos foi utilizado o método chamado de K- fold Cross-Validation, usando 10 partições.

A aplicação de algoritmos de comitê de classificadores no Weka permite a seleção de algoritmos bases em sua configuração. A fim de analisar o desempenho com diferentes classificadores de base, foram utilizados seis algoritmos de árvores de decisão, tais como: decision stump, hoeffding tree, random tree, REP tree, J48 e random forest. A escolha dos algoritmos é baseada no estudo de Walse, Dharaskar e Thakare (2016) e Coelho (2016).

Foram desenvolvidos seis experimentos, os quais estão descritos na tabela 2, o objetivo é analisar o comportamento dos classificadores selecionados em relação ao conjunto de dados final e verificar o impacto de técnicas de pré-processamento como discretização em todos os registros e balanceamento de classe nas medidas de qualidade em classificação selecionadas para analisar o desempenho dos algoritmos.

Experimentos	Descrição
1	Conjunto de dados final (apenas as classes discretizadas)
2	Aplicação da técnica de discretização em todos os registros no conjunto de dados final
3	Aplicação da técnica SMOTE com 100% no conjunto de dados discretizados
4	Aplicação da técnica SMOTE com 282% no conjunto de dados discretizados
5	Aplicação da técnica SMOTE com 100% no conjunto de dados final
6	Aplicação da técnica SMOTE com 282% no conjunto de dados final

Tabela 2 – Descrição dos experimentos realizados.

3. Resultados

Após a aplicação de adaboost.M1 e random susbpace na ferramenta de data mining Weka os resultados foram analisados por meio dos percentuais de acurácia, coeficiente Kappa, taxas de verdadeiros positivos e F-Measure gerados por cada algoritmo, com intuito de identificar qual o experimento gerou melhores modelos. Ao selecionar o experimento, foi identificado o algoritmo de base mais apropriado para adaboost.M1 e random subspace, por meio dos modelos gerados utilizando as medidas de qualidade definidas. Por fim, para encontrar o modelo com melhores resultados, adaboost.M1 e random subspace, utilizando o algoritmo de base mais apropriado no experimento selecionado, foram analisados por meio das medidas de qualidade definidas nesta pesquisa e pelo teste de significância aplicado no percentual de acurácia, correct resampled t-test.

O experimento que obteve melhores resultados foi o seis, em que se aplicou a técnica SMOTE com 282% no conjunto de dados final, chegando a percentuais de acurácia como 93,51% e 93,94%, obtidos por adaboost.M1 com random forest e por random subspace com

random forest, respectivamente e coeficiente Kappa como 0,8701 e 0,8787 obtidos também por adaboost.M1 com random forest e por random subspace com random forest, respectivamente.

Ao selecionar o experimento é possível analisar quais os classificadores de base são mais apropriados para adaboost.M1 e random subspace. Os resultados de acurácia e coeficiente Kappa obtidos por adaboost.M1, utilizando diferentes algoritmos bases, que obtiveram as taxas mais altas foi alcançado ao utilizar random forest, que possui o maior percentual de acurácia, chegando a 93,51% e para o coeficiente Kappa o algoritmo base é o mesmo, chegando a 0,8701. Para random subspace o algoritmo de base com melhor desempenho também é random forest, com 93,94% de acurácia e 0,8787 de coeficiente Kappa.

A fim de se identificar o modelo com melhores resultados foi realizada a comparação entre adaboost.M1 e random subspace utilizando o algoritmo de base random forest no experimento 6, para isso foi realizado teste de significância nos percentuais de acurácia, a tabela 3 mostra que não há diferenças estatisticamente significativa entre os resultados, apesar de que, em números absolutos, random subspace possui taxa superior ao adaboost.M1, chegando a 93,77%.

Exp. 6	**Adaboost.M1 (Random Forest)**	**Random Subspace (Random Forest)**
Acurácia	93,51	**93,77**
Desvio Padrão	0,89	0,89

Tabela 3 – Resultado do teste de significância.

A tabela 4 apresenta os resultados de verdadeiros positivos e F-Measure obtidos por adaboost.M1 e random subspace. O desempenho por classe de cada algoritmo, quando se trata de taxas de verdadeiros positivos, random subspace possui percentual maior na classe "A" com 0,975, já para a classe "R", adaboost.M1 obteve resultados superiores, com 0,913. Na medida F-Measure, em ambas as classes, o random subspace atingiu resultados superiores comparados aos de adaboost.M1, chegando a 0,942 e 0,937 na classe "A" e "R", respectivamente.

Medida	**Classe**	**Adaboost.M1 (Random Forest)**	**Random Subspace (Random Forest)**
TP-Rate	A	0,957	**0,975**
	R	**0,913**	0,904
	média	*0,935*	*0,939*
F-Measure	A	0,937	**0,942**
	R	0,934	**0,937**
	média	*0,935*	*0,939*

Tabela 4 – Valores de TP-Rate e F-Measure

4. Discussão

Em relação aos experimentos realizados, pode-se observar que os conjuntos de dados que não utilizam a técnica de balanceamento, mais precisamente 1 e 2, possuem altas taxas de verdadeiros positivos para classe “A” variando de 0,925 a 1, porém, quando se trata da classe “R”, são os experimentos que possuem as taxas menores, os resultados ficam entre 0,683 e 0,16, grande parte dos modelos gerados classificam, mais da metade, a classe “R” sendo “A”, neste sentido, a base com as classes desbalanceadas são melhores para classificar apenas o perfil de interação dos alunos aprovados.

Considerando que a classificação correta da classe “R” é importante, pois representam os alunos com desempenho inferiores, observou-se que, o experimento 6, utilizando a técnica SMOTE com percentual de sobreamostragem de 282%, ou seja, onde o número de instâncias das classes “A” e “R” possuem uma diferença de 9 registros, sobre o conjunto de dados original, notou-se que, as taxas de verdadeiros positivos da classe “R” possuem resultados entre 0,603 e 0,92, muito superiores ao dos experimentos que não utilizam a técnica.

O estudo de Gottardo (2012) realiza experimentos em dados educacionais, utiliza a técnica SMOTE para balancear a classe minoritária, nos quais os alunos com desempenho inferior estavam em maior concentração. Foram aplicados os algoritmos random forest e multilayer perceptron, em termos de acurácia global, no experimento em que o conjunto de dados é original, o primeiro algoritmo obteve uma taxa de 77,4% e o segundo 80,1%, já ao aplica-los no conjunto de dados balanceados, random forest chegou a 78,4% e multilayer perceptron 77,1%.

Neste sentido, observa-se que comparando o resultado obtido por random forest no trabalho de Gottardo (2012) de 78,4%, com random subspace utilizando random forest com taxa de 93,77%, alcançados no experimento 6 deste trabalho, as diferenças são altas e o resultado obtido nesta pesquisa supera aos encontrados no trabalho em questão, validando a utilização da técnica SMOTE para dados educacionais em que as classes estão desbalanceadas.

Além disso, pode-se comparar o percentual de acurácia obtido por random subspace ao utilizar random forest no experimento 6, tendo em vista que foi a configuração e o experimento mais adequado para este algoritmo no conjunto de dados utilizados, de 93,77% com percentuais obtidos em trabalhos como de Malaise, Malibari e Alkhozae (2014) que chegou a 80% utilizando adaboost.M1 em dados educacionais para previsão de desempenho; de Ayyappan e Kumar (2017), em que atingiu 72,18%, também aplicando adaboost.M1 em dados educacionais; de Souza (2016), que obteve um percentual de 98,55% com adaboost.M1 em uma base de dados com classe binária.

No trabalho de Santana, Maciel e Rodrigues (2014), foram utilizados sete algoritmos de classificação, random forest, multilayer perceptron, naive bayes, SVM, KNN, J48 e RBF em dados educacionais, considerando atributos relacionados a perfil de uso do AVA. O método de estratificação utilizado foi K-fold cross-validation com 10 partições, assim como nesta pesquisa. Foram analisados os resultados da matriz de confusão e acurácia, os melhores resultados encontrados na pesquisa foram ao utilizar duas classes, aprovado e reprovado, com o classificador J48, chegando ao percentual de acurácia de 74,68%.

Observa-se que ao comparar os percentuais obtidos em cada estudo, em números absolutos, o percentual atingido nesta pesquisa só não supera ao encontrado no trabalho de Souza (2016), no entanto, a utilização do algoritmo random subspace com random forest.

5. Conclusão

O acompanhamento do desempenho de estudantes em cursos e disciplinas ofertados na modalidade a distância tem sido amplamente explorado pela comunidade cientifica, a fim de auxiliar e buscar soluções que facilitem a compressão de diversos problemas pedagógicos.

A EDM é uma subárea de data mining que possui técnicas para realizar inferências em dados educacionais, a classificação é uma de suas principais tarefas, muito utilizada para verificar predições de desempenho e perfis de interações de alunos em ambientes virtuais. A fim de se obter dados mais precisos, foram utilizados os algoritmos adaboost.m1 e random subspace, que tem como principio o método de comitê de classificadores.

Para aplicação dos algoritmos foi necessário investigar como os dados estão armazenados na plataforma Moodle da UNESC, juntamente com o delineamento da disciplina utilizada na pesquisa, Metodologia Cientifica e da Pesquisa, para verificar quais atributos poderiam ser utilizados nos três perfis de interações de Moore (aluno-ambiente, aluno-professor e aluno-aluno).

Nesta etapa foram encontradas dificuldades como: entendimento do banco de dados do Moodle, devido ao fato de possuir muitas tabelas e terem poucos estudos que relatam de forma minuciosa a extração do conjunto de atributos de um banco de dado relacional, nesta situação, foram gerados diversos scripts para capturar diferentes atributos e realizada reuniões com o setor de EaD com intuito de melhor entender como os dados estavam estruturados e quais atributos seriam relevante; pela disciplina ser ofertada em cursos presenciais, as interações entre alunos e professores não ocorrem exclusivamente no ambiente virtual, essas dificuldades foram resolvidas deixando apenas a interação entre aluno-ambiente.

O pré-processamento é uma etapa extensa e requer bastante atenção, pois interfere diretamente na qualidade dos modelos encontrados, foram encontradas dificuldades relacionadas à quais técnicas empregar, sendo resolvido estudando de forma mais detalhada a natureza dos dados e trabalhos que estivessem relacionados a esta pesquisa.

Apesar das dificuldades, os resultados encontrados são satisfatórios e atingem os objetivos da pesquisa, possibilitando a identificação do algoritmo de comitê de classificação que possui melhor desempenho usando medidas de qualidade em classificação.

Após serem realizadas diversas análises comparativas entre os experimentos utilizados nesta pesquisa, o conjunto de dados mais apropriado para classificação correta tanto de alunos aprovados quanto reprovados, foram os dados originais, com apenas as classes discretizadas e a técnica SMOTE com percentual de sobreamostragem de 282%, denominado como experimento 6, nesse sentido, o algoritmo que obteve melhor desempenho neste conjunto de dados foi random subspace utilizando random forest como classificador de base, alcançando um percentual de acurácia de 93,77%.

Considerando os resultados obtidos nesta pesquisa, destacam-se algumas sugestões para trabalhos futuros: aplicar o mesmo conjunto de atributos em classificadores únicos a fim de comparar os resultados com os obtidos por algoritmos de comitê de classificação; empregar em outras disciplinas da modalidade a distância que possuem outras organizações para analisar os modelos gerados; adotar as três interações por Moore em disciplinas que possuem interações de aluno-aluno e aluno-professor em ambientes virtuais; e utilizar outras medidas de qualidade para analisar o desempenho por classe como matriz de confusão e curva ROC.

Referencias

Ayyappan, G; Kumar, S. K. A novel approach of ensemble models using edm. Indian Journal of Computer Science and Engineering, vol. 8, no. 6, 2017. Disponível em< http://www.ijcse.com/ijcse-issue.html?issue=20170806>. Acesso em: 23 jun. 2018.

Baker, R.S.J.D.; Isotani, S; Carvalho, A.M.J.B.D. Mineração de dados educacionais: oportunidades para o brasil. Revista Brasileira de Informática na Educação, vol. 19, no. 2, p. 2-3, 2011.

Chawla, N. V. et al (2002). SMOTE: Synthetic minority over-sampling technique. J. Artif. Intell. Res. (JAIR), 16, 321–357. Chawla, N. V., Japkowicz, N., e Kotcz, A. (2002). Editorial: special issue on learning from imbalanced data sets. SIGKDD Explorations, 6(1), 1–6.

Coelho, Guilherme Palermo. Geração, Seleção e Combinação de Componentes para ializaçãoEnsembles de Redes Neurais Aplicadas a Problemas de Classificação. 2006. 115 f. Monografia (Especialização) - Curso de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas Faculdade de Engenharia Elétrica e de Computação, Campinas, 2006.

Frank, Eibe et al. Data Mining: Pratical Machine Learning Tools and Techniques. 4 ed. San Francisco: Elsevier. 2017.

Gottardo, Ernani. Estimativa de desempenho acadêmico de estudantes em uma ava utilizando técnias de mineração de dados. 2012. 84 f. Dissertação (Mestrado) - Curso de Programa de pos-graduacapo em Computacao Aplicada, Universidade Tecnologica Federal do Parana, Curitiba, 2012.

Gottardo, E.; Kaestner, C.; Noronha, R. V. Aplicação de Técnicas de Mineração de Dados para Estimativa de Desempenho Acadêmico de Estudantes em um AVA Utilizando Dados com Classes Desbalanceadas. Em: ICBL2013–International Conference on Interactive Computer aided Blended Learning. 2013.

Han, J.; Kamber, M.; Pei, J.. Data mining: concepts and techniques. 3. ed. United States of America: Elsevier. 2012.

Júnior, O. G. José. Identificação de padrões para análise da evasão em cursos de graduação usando mineração de dados. 2015. 86 f. Dissertação (Mestrado) - Curso de Programa de Pós-graduação em Computação Aplicada, Universidade Tecnológica Federal do Paraná, Curitiba, 2015. Disponível em: <http://repositorio.utfpr.edu.br/jspui/bitstream/1/1995/1/CT_PPGCA_M_Oliveira%20Junior%2C%20Jos%C3%A9%20Gon%C3%A7alves_2015.pdf>. Acesso em: 10 jun. 2019.

Larose,T. D.; Larose, D. C.. Discovering knowledge in data: An introduction to data mining. Wiley & Sons, Inc., Hoboken, New Jersey. 2014. p. 10-15.

Malaise, A. A.; Malibari, A.; Alkhozae, M. Student's performance prediction system using multi agent data mining technique. In: INTERNATIONAL JOURNAL OF DATA MINING & KNOWLEDGE MANAGEMENT PROCESS, 5., 2014. Journal IJDKP.

Polikar, R.. Ensemble based systems in decision machine. In: IEEE circuits and systems magazine, set. 2006, vol 6.

Ramos, T, et al. Uso de mineração de dados educacionais para identificação de perfis e padrões de participação dos estudantes de cursos a distância. In:Iberian Conference on Information Systems and Technologies, 12., 2017, Lisboa. Disponível em <https://ieeexplore.ieee.org/document/7975960/ >. Acesso em: 15 abr. 2018.

Santana, C. Leandro; Maciel, MA. Alexandre; Rodrigues, L. Rodrigo. avaliação do perfil de uso no ambiente moodle utilizando técnicas de mineração de dados. Em: Simpósio Brasileiro de Informática na Educação-SBIE. 2014. p. 269-277.

Walse, Kishor H; Dharaskar, Rajiv V; Thakare, Vilas M. A study of human activitt recognition using adaboost classifiers on wisdm dataset. Iioab Journal. India, p. 68-76. 68 jan. 2016. Disponível em:

<https://www.researchgate.net/publication/324262009_A_study_of_human_activity_recognition_using_adaboost_classifiers_on_WISDM_dataset>. Acesso em: 02 jun. 2019.

ANEXO(S)

ANEXO A – Carta de aprovação

CEP
COMITÊ DE ÉTICA EM PESQUISA
DE SERES HUMANOS

RESOLUÇÃO

O Comitê de Ética em Pesquisa UNESC, reconhecido pela Comissão Nacional de Ética em Pesquisa (CONEP) / Ministério da Saúde analisou o projeto abaixo.

Parecer nº: 2.857.694 **CAAE:** 96550518.5.0000.0119

Pesquisador (a) Responsável: MERISANDRA CÔRTES DE MATTOS GARCIA

Pesquisador (a): ALINI MARANGONI EYNG
ANA CLAUDIA FONTANA MEDEIROS
LAINE DIMER
STEFANY MENDES DO NASCIMENTO

Título: "EDUCATIONAL BIG DATA EM EDUCAÇÃO A DISTÂNCIA".

Este projeto foi **Aprovado** em seus aspectos éticos e metodológicos, de acordo com as Diretrizes e Normas Internacionais e Nacionais. Toda e qualquer alteração do Projeto deverá ser comunicada ao CEP. Os membros do CEP não participaram do processo de avaliação dos projetos onde constam como pesquisadores.

Criciúma, 30 de agosto de 2018.

Renan Antônio Ceretta
Coordenador do CEP

Av. Universitária, 1.105 – Bairro Universitário – CEP: 88.806-000 – Criciúma / SC
Bloco Administrativo – Sala 31 | Fone (48) 3431 2606 | cetica@unesc.net | www.unesc.net/cep
Horário de funcionamento do CEP: de segunda a sexta-feira, das 08h às 12h e das 13h às 17h.

ANEXO B – Termo de confiabilidade

CEP
COMITÊ DE ÉTICA EM PESQUISA DE SERES HUMANOS

Termo de Confidencialidade

Título da Pesquisa: EDUCATIONAL BIG DATA EM EDUCAÇÃO A DISTÂNCIA

Objetivo: Identificar modelos por meio das tarefas de Educational Data Mining, de classificação e agrupamento, para identificação de perfis dos alunos nas disciplinas da UNESC de Introdução a Engenharia de Segurança do Trabalho e Metodologia Científica e da Pesquisa que acontecem na modalidade a distância.

Período da coleta de dados: 15/09/2018 a 10/10/2018

Local da coleta: Setor de Tecnologia da Informação (TI) / Setor de Educação a Distância (SEAD) – Universidade do Extremo Sul Catarinense

Pesquisador/Orientador: Merisandra Côrtes de Mattos Garcia	**Telefone:** (48) 9 9611-1277
Pesquisador/Acadêmico: Alini Marangoni Eyng	**Telefone:** (48) 9 9137-9951
Pesquisador/Acadêmico: Ana Claudia Fontana Medeiros	**Telefone:** (48) 9 9636-5037
Pesquisador/Acadêmico: Laíne Dimer	**Telefone:** (48) 9 9646-0892
Pesquisador/Acadêmico: Stefany Mendes do Nascimento	**Telefone:** (48) 9 9851-9446

Os pesquisadores (abaixo assinados) se comprometem a preservar a privacidade e o anonimato dos sujeitos com relação a toda documentação e toda informação obtidas nas atividades e pesquisas a serem coletados em bases de dados das disciplinas ofertadas a distância do local informado acima.

Concordam, igualmente, em:

- Manter o sigilo das informações de qualquer pessoa física ou jurídica vinculada de alguma forma a este projeto;
- Não divulgar a terceiros a natureza e o conteúdo de qualquer informação que componha ou tenha resultado de atividades técnicas do projeto de pesquisa;
- Não permitir a terceiros o manuseio de qualquer documentação que componha ou tenha resultado de atividades do projeto de pesquisa;
- Não explorar, em benefício próprio, informações e documentos adquiridos através da participação em atividades do projeto de pesquisa;

Av. Universitária, 1.105 – Bairro Universitário – CEP: 88.806-000 – Criciúma / SC
Bloco Administrativo – Sala 31 | Fone (48) 3431 2606 | cetica@unesc.net | www.unesc.net/cep
Horário de funcionamento do CEP: de segunda a sexta-feira, das 08h às 12h e das 13h às 17h.

CEP
COMITÊ DE ÉTICA EM PESQUISA DE SERES HUMANOS

Termo de Confidencialidade

- Não permitir o uso por outrem de informações e documentos adquiridos através da participação em atividades do projeto de pesquisa.
- Manter as informações em poder do pesquisador Merisandra Côrtes de Mattos Garcia por um período de 5 anos. Após este período, os dados serão destruídos.

Por fim, declaram ter conhecimento de que as informações e os documentos pertinentes às atividades técnicas da execução da pesquisa somente podem ser acessados por aqueles que assinaram o Termo de Confidencialidade, excetuando-se os casos em que a quebra de confidencialidade é inerente à atividade ou em que a informação e/ou documentação já for de domínio público.

ASSINATURAS	
Orientador(a) **Assinatura** **Nome:** Merisandra Côrtes de Mattos Garcia **CPF:** 947.749.430-53	**Pesquisador(a)** **Assinatura** **Nome:** Alini Marangoni Eyng **CPF:** 086.364.469-45
Pesquisador(a) **Assinatura** **Nome:**Ana Claudia Fontana Medeiros **CPF:** 089.430.629-40	**Pesquisador(a)** **Assinatura** **Nome:** Laíne Dimer **CPF:** 095.230.959-94

Av. Universitária, 1.105 – Bairro Universitário – CEP: 88.806-000 – Criciúma / SC
Bloco Administrativo – Sala 31 | Fone (48) 3431 2606 | cetica@unesc.net | www.unesc.net/cep
Horário de funcionamento do CEP: de segunda a sexta-feira, das 08h às 12h e das 13h às 17h.