

Comparação de diferentes configurações de bases de dados para a identificação precoce do risco de reprovação: o caso de uma disciplina semipresencial de Algoritmos e Programação

Matheus F. B. Machado¹, Cristian Cechinel¹, Vinicius F. C. Ramos¹

¹Centro de Ciências, Tecnologias e Saúde (CTS)
Universidade Federal de Santa Catarina (UFSC)
Araranguá – SC – Brasil

matheusmachadoufsc@gmail.com, contato@crisiancechinel.pro.br, v.ramos@ufsc.br

Abstract. *This paper presents a comparison of the use of different datasets to generate models for predicting at-risk students. Four different datasets were tested together with three distinct classifiers in the context of an Introductory Programming Course. Initial results indicate that the use of a dataset composed by the counting of students interactions inside the Learning Management System during the period of the course was the one that allowed the models to achieve the best performances.*

Resumo. *O presente trabalho apresenta uma comparação do uso de diferentes configurações de bases de dados para a geração de modelos de predição de estudantes em risco de reprovação. Foram testadas 4 diferentes configurações de bases de dados e utilizados 3 algoritmos de classificação para geração de modelos de predição no contexto de uma disciplina semipresencial de Algoritmos e Programação. Os resultados apontaram que o uso de uma base de dados com a contagem de interações dos estudantes no Ambiente Virtual de Aprendizagem ao longo das semanas foi a que permitiu que os modelos de predição alcançassem os melhores resultados.*

1. Introdução

O ensino de programação é uma das inúmeras tarefas desafiadoras em todos os níveis de educação. A alta taxa de reprovação e evasão em disciplinas de programação atinge números altíssimos em todo o mundo. Há estudos que mostram percentuais variando de 30 a 80% dessas taxas [Vihavainen et al. 2014, Bennedsen and Caspersen 2007, Watson and Li 2014, Ramos et al. 2015]. Na literatura, existem inúmeros autores que apontam os diversos fatores para essa elevada taxa de reprovação. Dentre estes fatores, os que se destacam são: a dificuldade de abstração necessária para o desenvolvimento de algoritmos e a dificuldade em resolução de problemas [Jenkins 2002, Dunican 2002]. Jenkins [Jenkins 2002], por exemplo, acredita que as disciplinas de programação são oferecidas em um estágio muito precoce na vida dos estudantes de graduação. Na maioria dos casos, a disciplina de programação é ofertada no primeiro semestre do curso. Para o autor, os alunos calouros estão em um processo transitório entre o ensino médio e a graduação, em um momento muito peculiar e de grande instabilidade, experimentando várias mudanças e dificuldades. Os alunos adentram em um ambiente com muito menos restrição com currículo e disciplinas diferentes do que eles estão acostumados.

Para reduzir as altas taxas de reprovação, alguns autores sugerem diferentes práticas pedagógicas [Mow 2006, Gomes 2010]. Muitas dessas práticas têm a característica de envolver o aluno nas diversas atividades de ensino e evidenciam bons resultados quando há *feedback* contínuo por parte do professor. Ao compreendermos que o *feedback* do professor é um fator importante e efetivo ao aprendizado do aluno, também compreendemos que a identificação precoce das dificuldades de aprendizagem do aluno é um outro fator importante para a redução das taxas de reprovação.

Diferentes trabalhos abordam o problema da predição de estudantes em risco utilizando diferentes fontes de dados. É comum encontrar trabalhos de investigação na área utilizando dados provenientes dos sistemas acadêmicos, de interações recolhidas por meio dos ambientes virtuais de aprendizagem, das avaliações obtidas pelos acadêmicos ao longo do curso/semestre e também de informações não diretamente acadêmicas, como o recolhimento de questionários. O objetivo desse trabalho é o de contrastar os desempenhos de diferentes modelos de predição em uma disciplina semipresencial de algoritmos e programação utilizando diferentes configurações de bases de dados de entrada para os modelos. Em outras palavras, espera-se identificar quais os conjuntos de dados mais adequados para serem utilizados nesse tipo de predição, e de uma forma que a mesma possa acontecer de maneira precoce dentro da disciplina (em tempo de permitir que o professor possa adotar algum tipo de intervenção).

O presente trabalho está organizado da seguinte maneira: a Seção 2 apresenta alguns trabalhos relacionados na predição de estudantes em risco de reprovação. A Seção 3 explica a metodologia utilizada no presente trabalho, incluindo uma descrição das bases de dados utilizadas nos experimentos. A Seção 4 apresenta uma análise dos resultados obtidos e a Seção 5 encerra o artigo com as considerações finais e possibilidades de trabalhos futuros.

2. Trabalhos Relacionados

Técnicas de mineração de dados educacionais *EDM*, do inglês *Educational Data Mining* são constantemente utilizadas no desenvolvimento de modelos preditivos e exploratórios com o objetivo de compreender melhor o desempenho dos alunos e intervir de maneira precoce em situações de risco. Diversos estudos têm mencionado métodos de EDM para prever alunos em risco devido à necessidade de melhorar a qualidade dos processos de ensino-aprendizagem e também de compreender a situação do rendimento dos estudantes [Khobragade 2015]. Entre os exemplos de trabalhos de identificação precoce de estudantes em risco, podemos citar os experimentos de [Detoni et al. 2016] e [Queiroga et al. 2017] que buscam criar, analisar e avaliar modelos de predição para evitar fracassos de alunos no decorrer do semestre utilizando diferentes algoritmos de classificação aplicados a dados coletados das interações dos acadêmicos em um AVA (Ambiente Virtual de Aprendizagem). Nesses trabalhos, os autores defendem o uso da contagem das interações dos estudantes e professores dentro do AVA, além da utilização de atributos derivados dessas contagens como informações de entrada para os modelos de predição.

Ainda, [Brito et al. 2014] propõem a criação de modelos para prever estudantes que necessitam de apoio em matérias da área de exatas no primeiro semestre por meio das notas de ingresso no vestibular. Os autores encontraram modelos com acurácias gerais

de identificação de estudantes nessa situação a 70%. [Khobragade 2015] buscou prever o fracasso dos alunos precocemente utilizando dados de pesquisa que foram realizadas por meio de formulários (dados demográficos, culturais, socioeconômico, entre outros) preenchidos pelos estudantes. Um dos maiores desafios enfrentados foi a seleção dos atributos além do grande desbalanceamento dos dados (muitos estudantes aprovados e poucos reprovados).

[Costa et al. 2017] faz um comparativo de quatro técnicas de EDM em dois conjuntos de dados para avaliar a eficácia das mesmas em prever os alunos em situações de risco nas primeiras semanas, enquanto [Ahmad et al. 2015] propõem métodos para prever o desempenho acadêmico dos estudantes logo no primeiro ano, realizando uma análise comparativa de três técnicas de classificação e utilizando a ferramenta WEKA. De acordo com [C. Santana et al. 2014], os atributos que podem ser analisados dentro de um AVA são diversos. Essa variedade abre a possibilidade de experimentações com diferentes configurações dos atributos coletados.

O presente trabalho diferencia-se dos trabalhos anteriormente citados por ter como objetivo principal contrastar diferentes combinações de conjuntos de dados para geração de modelos preditivos no contexto específico de uma disciplina semipresencial de Algoritmos e Programação.

3. Metodologia

O presente estudo realizou experimentos com dados de uma disciplina semipresencial de Algoritmos e Programação, combinando diferentes conjuntos desses dados. Seguem as perguntas de pesquisa que norteiam este trabalho: 1) Quais são os conjuntos de dados mais adequados para prever alunos em risco de reprovação? e 2) O uso de diferentes conjuntos de dados relacionados a um mesmo curso leva a diferenças na eficácia dos modelos preditivos?

3.1. Descrição dos dados

Foram coletados dados dos *logs* das interações realizadas no AVA Moodle pelos estudantes da disciplina de Algoritmos e Programação ofertada no curso de tecnologias da informação e comunicação da Universidade Federal de Santa Catarina em quatro semestres diferentes (nos anos de 2016 e 2017). O curso é noturno e a disciplina em questão possui 108 horas semestrais ministradas ao longo de 18 semanas (4 horas/aula presenciais e 2 semipresenciais por semana), sendo a última semana reservada para avaliações de recuperação ou de substituição. Para as atividades semipresenciais são disponibilizados diversos recursos no Moodle, tais como: vídeo-aulas, textos de apoio, exercícios de programação utilizando o VPL (Virtual Programming Lab ¹), questionários, listas resolvidas, entre outros. Juntamente com os dados dos logs, também foram coletados dados de um questionário aplicado aos alunos. O objetivo do questionário, segundo o professor responsável pela disciplina, é "compreender melhor os interesses, hábitos e conhecimentos prévios, relacionados à computação e programação, dos alunos. Desta forma, é possível identificar o perfil do aluno que recém ingressou na instituição.". A análise deste questionário pelo professor é essencial para a identificação de potenciais alunos com dificuldades prévias, tais como o desafio do deslocamento (e conseqüentemente a chegada

¹https://moodle.org/plugins/mod_vpl

com atraso no horário da aula), a mudança de cidade (para aqueles que mudaram de cidade para poder estudar na instituição), ou a quantidade de horas de trabalho diárias. O questionário é composto por 28 questões (entre fechadas e abertas)². Esse são alguns exemplos de informações coletadas: 1) a quantidade de horas por dia, em média, que o estudante despense para trabalhar e para estudar, 2) se o curso em que está matriculado foi sua primeira opção no vestibular, 3) quais linguagens de programação que o estudante possui algum conhecimento e o seu nível de conhecimento, e 4) se o estudante possui algum trabalho remunerado e de quantas horas.

A partir dos logs das interações e do questionário foram construídas quatro bases de dados utilizadas para a disciplina, conforme apresentado na Tabela 1. A base de dados 1 (BD1) consiste na contagem simples das interações semanais dos estudantes na sala do Moodle da disciplina, seguindo uma metodologia proposta por [Detoni et al. 2016] e [Queiroga et al. 2017]. A base de dados 2 (BD2) é uma variação da BD1 em que, além da contagem simples das interações semanais, outros atributos derivados dessas interações são inseridos na base de dados. Precisamente, os atributos derivados incluídos na BD2 foram: 1) a média das interações na semana, 2) a mediana do conjunto das interações na semana e 3) o número absoluto de interações na semana. A base de dados 3 (BD3) é formada pela contagem das interações mais as respostas do questionário. Por último, a base de dados 4 (BD4) é uma combinação de todos os dados recolhidos (contagem de interações, atributos derivados e respostas do questionário).

Tabela 1. Conjuntos de Dados utilizadas

Base de Dados	Descrição
BD1	Contagem de interações
BD2	Contagem de interações + atributos derivados
BD3	Contagem de interações + questionário
BD4	Contagem de interações + atributos derivados + questionário

3.2. Pré-processamento

O pré-processamento dos dados foi realizado por meio de *scripts* em Python. Estes são responsáveis por gerar um arquivo contendo a identificação (em *hash*) dos estudantes e a quantidade de interações realizadas ao longo das semanas no AVA. Além das quantidades de interações, foi gerado um conjunto de atributos derivados conforme mencionado anteriormente. Os cálculos dos atributos derivados foram realizados utilizando as bibliotecas *pandas* e *numpy*. Todos os atributos derivados e a contagem das interações foram calculados para cada uma das 17 semanas de aula (lembrando que no total são 18 semanas, mas a última foi excluída por se tratar de avaliações de recuperação ou substitutivas). Para selecionar as respostas das 28 perguntas do questionário que deveriam ser incluídas nos BD3 e BD4, utilizamos o algoritmo de seleção de características denominado *Recursive Feature Elimination*. Esse algoritmo é capaz de identificar os atributos mais irrelevantes e redundantes e, conseqüentemente, podem ser desconsiderados.

²Disponibilizado para consulta em <http://bit.do/eo4BA>

3.3. Histórico e estágio atual dos experimentos

Em nosso trabalho anterior [Machado et al. 2018], apresentamos resultados iniciais sobre a predição precoce de reprovação de alunos de uma disciplina de algoritmos e programação ao longo de 4 semestres consecutivos (2016.01, 2016.02, 2017.01 e 2017.02), sendo ministrada pelo mesmo professor.

No referido trabalho, os autores utilizaram a contagem de interações de alunos no ambiente Moodle. A Tabela 2 mostra o conjunto de dados usados no referido estudo. Os melhores resultados apresentaram 95% de acurácia para a identificação de alunos reprovados logo no início do semestre (até a sexta semana). Foram utilizadas as técnicas de aprendizagem de máquina para realizar a predição, sendo que os algoritmos Naive Bayes e Random Forest tiveram maior acurácia. Ainda neste trabalho, foi possível identificar que os melhores resultados foram obtidos com os dados de interações dos alunos no último semestre analisado.

Um ponto importante a ser considerado no estudo anterior é o desenho instrucional do Moodle. No último semestre analisado (2017.02), a estrutura pedagógica, tanto de orientações aos alunos quanto de atividades e recursos, era muito mais sólida do que nos semestres anteriores. As tabelas 2 e 3 mostram a evolução do ambiente virtual da disciplina.

A Tabela 2 apresenta a quantidade de interações feitas pelos alunos matriculados na disciplina, assim como a quantidade de alunos e a quantidade de aprovados e reprovados. Não há distinção no número de reprovados por faltas ou por nota. A única ressalva que fazemos é na quantidade de interações média por alunos que é um pouco maior na edição de 2017.01, sendo 1.239. Acreditamos que um dos motivos é que um dos alunos de 2017.2 foi inscrito 1 mês depois do início das aulas, além disso, também verificamos que: primeiramente, foram 14 desistências em 2017.01, sendo 7 abandonos após a primeira avaliação, 3 abandonos após a segunda avaliação e 3 alunos que nunca cursaram e foram 17 desistências na turma de 2017.02, sendo 8 abandonos logo após a primeira avaliação, 5 após a segunda avaliação e 4 nunca cursaram; e, segundo, o professor exigia que os alunos realizassem um exercício no Moodle em todas as aulas presenciais de 2017.01, aumentando o número de interações em sala de aula. A Tabela 3 mostra a quantidade de atividades online disponibilizadas no ambiente Moodle, é possível verificar a evolução dessa estrutura a partir da quantidade apresentada. Foram 8 novos laboratórios online (VPL), 5 novas atividades e 3 novos materiais de apoio de 2017.01 para 2017.02. No ano anterior, o crescimento do número de atividades é muito significativo. Neste caso, compreendemos que o desenho instrucional pode ter tido papel fundamental nos modelos de predição utilizados, pois ambas as tabelas corroboram para este indício.

Portanto, o presente artigo é uma continuação do referido trabalho, apresentando novos experimentos com a turma (2017.02) que usufruiu de um desenho instrucional mais completo.

3.4. Geração dos modelos de predição

Foram testados três diferentes modelos de predição (Random Forest, Naive Bayes e J48) utilizando a biblioteca scikit-learn. Considerando que a quantidade de casos da base de dados utilizada é pequena, utilizou-se a técnica de particionamento chamada *Leave-one-out crossvalidation* (LOOC) em que são utilizados $n - 1$ casos para treino e o caso restante

Tabela 2. Conjunto de dados das interações

Semestre	Qtd. de interações	Alunos	Aprovados	Reprovados
2016-1	12.775	26	7	19
2016-2	15.118	24	6	18
2017-1	34.684	28	10	18
2017-2	37.400	33	12	21
Totais	99.977	121	35	86

Tabela 3. Atividades online

Semestre	Lab. Online	Atividades Online	Materiais de Apoio	Links Referências
2016.01	52	3	16	3
2016.02	53	5	27	3
2017.01	73	7	32	6
2017.02	81	12	35	6

para teste. O valor do desempenho final do modelo é calculado pela média dos desempenhos das n iterações. A medida de avaliação utilizada é a ROC (do inglês *Receiver Operating Characteristic*). A ROC é um gráfico de sensibilidade (taxa de verdadeiro positivo-TVP) no eixo y pela especificidade (taxa de falso positivo-TFP) no eixo x . Cada modelo possui um valor ROC que, em nossa análise, é plotado nos gráficos para cada uma das semanas do semestre. Essa medida permite uma análise geral e panorâmica dos modelos, pois não privilegia a classificação correta de nenhuma das classes. Considerando que estamos trabalhando com dados desbalanceados, é necessário tomar a precaução em não avaliar somente a acurácia geral dos modelos. Segundo [El Khouli et al. 2009], bons classificadores apresentam valores ROC acima de 0.7.

4. Análise de resultados

A Figura 1 apresenta os resultados ROC em cada semana, para cada um dos experimentos e utilizando três diferentes algoritmos para predição (Naive Bayes, J48 e Random Forest). O modelo J48 mostrou que a BD1 e a BD2 obtiveram os melhores resultados, conseguindo uma ROC maior que 0.75 a partir da semana 4. Por outro lado, o algoritmo Random Forest mostrou um desempenho mais estável das bases, sendo que todas as 4 bases utilizadas obtiveram valores ROC acima de 0.75 a partir da semana 6. O modelo Naive Bayes permite verificar um melhor desempenho na predição, sendo obtido quando a BD1 é utilizada e a partir da semana 6.

A Figura 2 apresenta os gráficos com as taxas de Verdadeiros Positivos (VP) e Verdadeiros Negativos (VN) para a BD1 e BD2 e utilizando os algoritmos Random Forest e J48. Essa visualização permite observar o comportamento dos modelos para a classificação de cada uma das categorias de estudantes (aprovados e reprovados). Como é possível observar, os modelos de predição possuem mais facilidade para classificar os alunos aprovados (Verdadeiros negativos). Isso se deve ao desbalanceamento do conjunto que possui uma maior quantidade desses alunos (63.6%) em comparação com os reprovados (36.4%). O modelo Random Forest obteve taxas de VP e VN acima de 75% a partir da semana 7 para o BD2 e acima de 75% a partir da semana 4 para o BD1. Ainda, o modelo gerado com o algoritmo J48 obteve taxas de VP e VN também acima de 75%

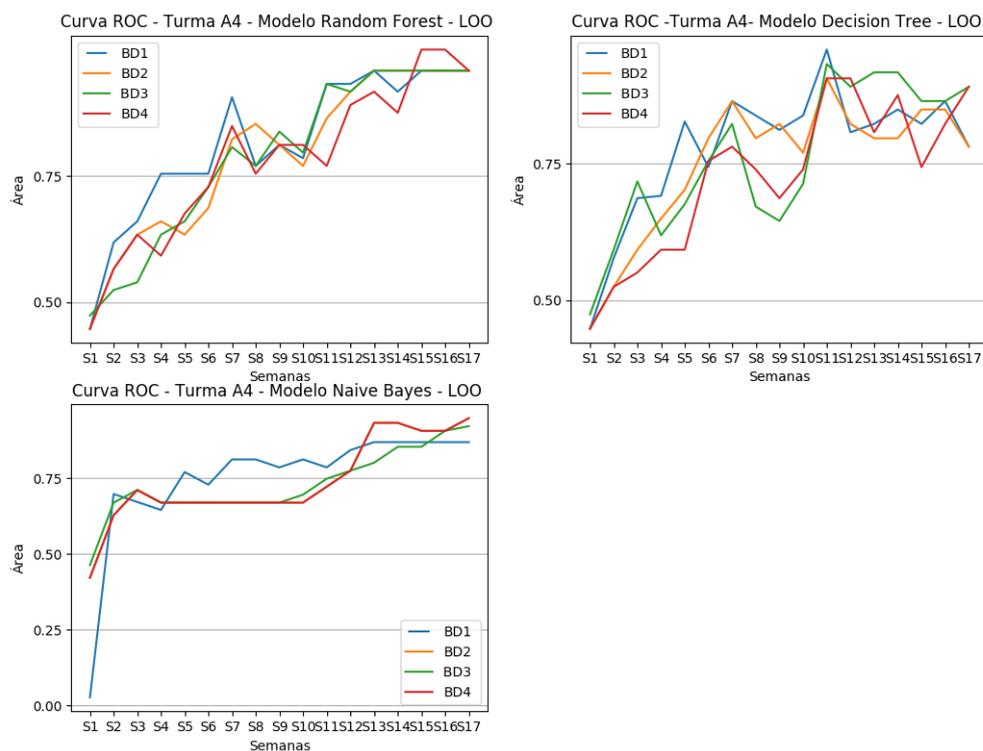


Figura 1. Área sob a curva ROC para as diferentes bases de dados ao longo das semanas

para ambas as bases de dados, porém a partir das semanas 6 e 7.

A primeira questão de pesquisa tem o objetivo de identificar os conjuntos de dados mais adequados para prever alunos em risco de reprovação. Os experimentos nos mostram que os melhores resultados foram obtidos por meio da BD1 (utilizando apenas a contagem de interações), seguido pela BD2 (utilizando a contagem de interações e atributos derivados).

Também podemos inferir pelos resultados dos experimentos, que a segunda pergunta de pesquisa também é verdadeira, uma vez que o uso de diferentes conjuntos de dados relacionados a um mesmo curso efetivamente levou a diferenças na eficácia dos modelos preditivos.

Cabe ressaltar que os resultados encontrados divergem levemente da literatura já publicada [Queiroga et al. 2017, Detoni et al. 2016] quando levamos em consideração especificamente a utilização de atributos derivados na predição de estudantes em risco utilizando a contagem de interações. Nesses estudos, os autores mencionam que o uso de atributos derivados auxilia na melhora do desempenho dos modelos de predição já nas primeiras semanas dos cursos. Com base nisso, a expectativa inicial do nosso trabalho era de encontrar desempenhos melhores com as bases de dados BD2, BD3 e BD4, porém os resultados encontrados contrariaram essa ideia inicial. Isso pode estar relacionado com a própria natureza dos dados utilizados no presente trabalho. Enquanto os estudos an-

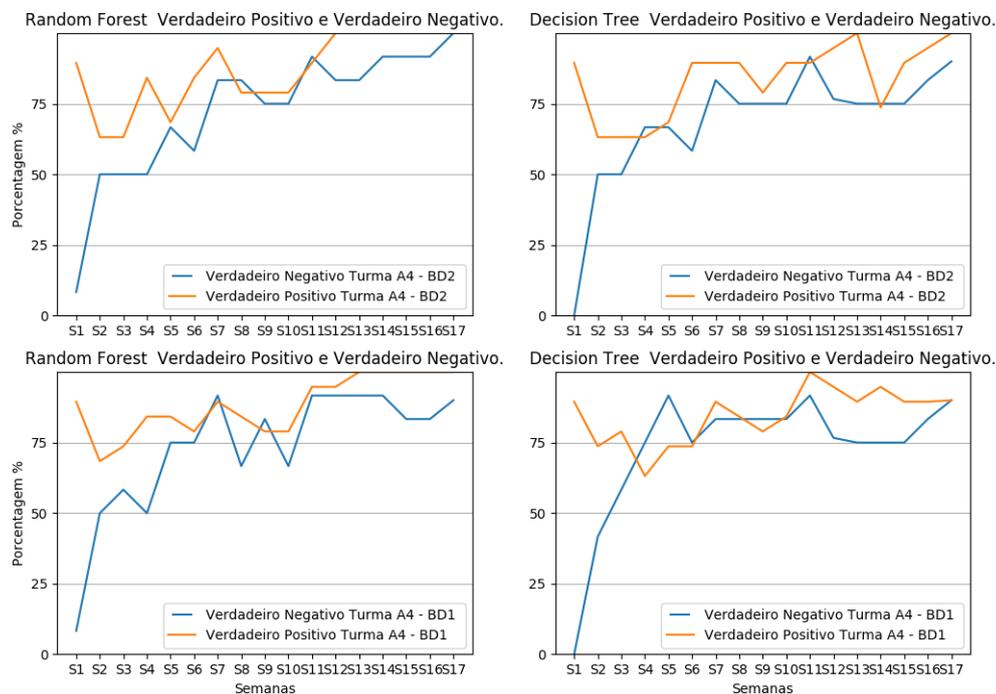


Figura 2. Verdadeiros Negativos e Verdadeiros Positivos para BD1 e BD2

teriormente mencionados utilizavam dados de cursos totalmente a distância e em cursos de Licenciatura e Técnicos, no presente estudo foram utilizados dados de uma disciplina semipresencial, noturna e de introdução aos tópicos de algoritmos e programação.

5. Considerações finais e trabalhos futuros

A predição precoce dos estudantes em risco de reprovação pode auxiliar o professor a tomar decisões sobre o método de ensino que está sendo aplicado na disciplina, podendo mudar o formato para colaborar com os alunos que apresentam dificuldades e possuem maior risco de reprovação. Neste trabalho, foram gerados modelos capazes de prever estudantes com risco de reprovação em uma disciplina semipresencial de Algoritmos e Programação e utilizando quatro configurações diferentes de bases de dados. A partir dos resultados obtidos, foi possível observar que a base de dados que utiliza apenas a contagem das interações dos estudantes no AVA ao longo da disciplina (BD1) foi a que permitiu aos modelos as melhores taxas de acerto. Nos melhores casos, os experimentos encontraram modelos capazes de prever estudantes em risco de reprovação com taxas de acerto, em ambas as categorias, acima de 75% já a partir da semana 4.

Esses resultados contrariam levemente a literatura já publicada sobre a utilização de atributos derivados aliados à contagem de interações para a predição de estudantes em risco. Na literatura existente, a utilização de atributos derivados leva a um melhor desempenho dos modelos, o que não ocorreu no presente estudo. Essa diferença pode estar relacionada à natureza diferente dos dados utilizados nos experimentos, uma vez que, nesse estudo estamos trabalhando com o cenário de uma disciplina semipresencial

de algoritmos e programação de computadores.

Como trabalho futuro pretende-se testar novamente os modelos de predição utilizando bases de dados processadas por meio de outras técnicas, como a normalização de alguns atributos e o balanceamento das classes alvo. Outra possibilidade é a contagem dos tipos das interações realizadas dentro do AVA. No presente trabalho, todas as interações são contabilizadas em uma mesma variável, sem qualquer identificação de qual ação foi realizada ou de qual atividade AVA foi acessada ou realizada.

Agradecimentos

Esse trabalho foi financiado pelo CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), por meio do Edital Universal 01/2016 processo 404369/2016-2 projeto: “Comparação entre Diferentes Abordagens na Modelagem e Identificação de Acadêmicos em Risco em Cursos de Educação a Distância” e pelo Programa Institucional de Iniciação Científica e Tecnológica – PIICT (2017/2018) da UFSC com bolsa também concedida pelo CNPq.

Referências

- Ahmad, F., Ismail, N. H., and Aziz, A. A. (2015). The prediction of students' academic performance using classification data mining techniques. *Applied Mathematical Sciences*, 9(129):6415–6426.
- Bennedsen, J. and Caspersen, M. E. (2007). Failure Rates in Introductory Programming. *SIGCSE Bull.*, 39(2):32–36.
- Brito, D. M. d., Júnior, I. A. d. A., Queiroga, E. V., and do Rêgo, T. G. (2014). Predição de desempenho de alunos do primeiro período baseado nas notas de ingresso utilizando métodos de aprendizagem de máquina.
- C. Santana, L., M. A. Maciel, A., and L. Rodrigues, R. (2014). Avaliação do Perfil de Uso no Ambiente Moodle Utilizando Técnicas de Mineração de Dados. (Cbie):269.
- Costa, E. B., Fonseca, B., Santana, M. A., de Araújo, F. F., and Rego, J. (2017). Evaluating the effectiveness of educational data mining techniques for early prediction of students' academic failure in introductory programming courses. *Computers in Human Behavior*, 73:247–256.
- Detoni, D., Cechinel, C., Matsumura, R. A., and Brauner, D. F. (2016). Learning to Identify At-Risk Students in Distance Education Using Interaction Counts. *Revista de Informática Teórica e Aplicada*, 23(2):124–140.
- Dunican, E. (2002). Making the analogy: Alternative delivery techniques for first year programming courses. In *Proceedings from the 14th workshop of the psychology of programming interest group, Brunel University*, pages 89–99.
- El Khouli, R. H., Macura, K. J., Barker, P. B., Habba, M. R., Jacobs, M. A., and Bluemke, D. A. (2009). Relationship of temporal resolution to diagnostic performance for dynamic contrast enhanced MRI of the breast. *Journal of Magnetic Resonance Imaging*, 30(5):999–1004.
- Gomes, A. (2010). *Dificuldades de aprendizagem de programação de computadores: contributos para a sua compreensão e resolução*. PhD thesis, Faculdade de Ciências e Tecnologia da Universidade.

- Jenkins, T. (2002). On the difficulty of learning to program. In *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences*, volume 4, pages 53–58. Citeseer.
- Khobragade, L. P. (2015). Predicting Students' Academic Failure Using Data Mining Techniques. *International Journal of Advance Research in Computer Science and Management Studies*, 3(5):2321–7782.
- Machado, M. F. B., Cechinel, C., and Ramos, V. (2018). Identificação precoce de estudantes em risco de reprovação em disciplinas semipresenciais de programação de computadores. (2012).
- Mow, I. T. V. C. (2006). *The effectiveness of a cognitive apprenticeship-based learning environment (cable) in teaching computer programming*. PhD thesis, University of South Australia.
- Queiroga, E., Cechinel, C., and Araújo, R. (2017). Predição de estudantes com risco de evasão em cursos técnicos a distância. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 28, page 1547.
- Ramos, V., Wazlawick, R., Galimberti, M., Freitas, M., and Mariani, A. C. (2015). A comparação da realidade mundial do ensino de programação para iniciantes com a realidade nacional: Revisão sistemática da literatura em eventos brasileiros. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 26, page 318.
- Vihavainen, A., Airaksinen, J., and Watson, C. (2014). A Systematic Review of Approaches for Teaching Introductory Programming and Their Influence on Success. In *Proceedings of the Tenth Annual Conference on International Computing Education Research*, ICER '14, pages 19–26, New York, NY, USA. ACM.
- Watson, C. and Li, F. W. (2014). Failure Rates in Introductory Programming Revisited. In *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education*, ITiCSE '14, pages 39–44, New York, NY, USA. ACM.