UNIVERSIDADE FEDERAL
DE SANTA CATARINA

Comparação entre a utilização de diferentes conjuntos de dados para a modelagem e identificação de acadêmicos em risco

---

Programa Institucional de Iniciação Científica e Tecnológica – PIICT Bolsas PIBIC/CNPq – PIBIC-Af/CNPq – BIPI/UFSC 2017/2018

Edital PROPESQ 01/2017

Bolsista:

Matheus Francisco Batista Machado – matheusmachadoufsc@gmail.com

Orientador:

Cristian Cechinel – cristian.cechinel@ufsc.br

30 de agosto de 2018

1 Resumo

As altas taxas de reprovação e evasão de estudantes são frequentemente apontadas como os principais obstáculos enfrentados na implementação e consolidação de cursos a Distância. A identificação com antecedência de estudantes em risco pode auxiliar de maneira decisiva o trabalho de professores e tutores. O projeto tem como objetivo gerar e testar diferentes modelos de predição para identificação de estudantes em risco em cursos de educação a distância. Para isso, foi coletados dados de acesso aos ambientes virtuais de aprendizagem de cursos de graduação semi-presenciais da Universidade Federal de Santa Catarina. Os dados serão pré-processados de maneira a gerar sete diferentes tipos de bases de dados e permitindo contrastar características específicas da presença dos acadêmicos nos cursos. Os modelos de predição com as melhores acurácias serão então disponibilizados para uso dos professores em uma ferramenta de visualização de logs de acesso ao Moodle que está atualmente em desenvolvimento. Espera-se que a disponibilização de uma ferramenta livre para predição da situação de acadêmicos em risco possa auxiliar o trabalho de professores e tutores e ajudar a diminuir as atuais taxas de evasão e reprovação dos cursos.

Sumário

1 Resumo **1**

2 Introdução **3**

2.1 Problemática 4

2.2 Objetivos 5

2.2.1 Objetivo Geral 5

2.2.2 Objetivos Específicos 5

3 Fundamentação teórica **6**

3.1 Mineração de dados educacionias 6

4 Metodologia e desenvolvimento **7**

4.1 Fases do desenvolvimento do projeto 9

4.2 Desenvolvimento e detalhamento dos experimentos 12

4.3 Fase 0: Coleta de dados 13

4.4 Fase 1: Pré-processamento e geração de base de dados 13

4.5 Fase 2: Geração e testes de modelos de predição 14

5 Análise de resultados **16**

6 Atividades do Bolsista **20**

7 Avaliação do Bolsista em Relação ao PIBIC **23**

8 Considerações Finais **23**

Apêndice **28**

Apêndice A Questionário **28**

2 Introdução

No 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 [Jenkins, 2002, Dunican, 2002] e a dificuldade em resoluções 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 aluno 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 menos restrição com o currículo e disciplinas diferentes do que eles estão acostumados.

Com a inserção das Tecnologias de Informação e Comunicação (TICs) na área educacional levou as instituições de ensino a utilizarem sistemas baseado em web como apoio ao processo de ensino e aprendizagem, pois não necessita de um local físico ou de plataforma computacional, assim o estudante tem acesso facilitado aos recursos de apoio [Brusilovsky and Peylo, 2003]. Esses Ambientes Virtual de Aprendizagem (AVA) geram grandes quantidades de dados com informações sobre os alunos como atividades, provas, exercícios e até comentários realizados. Com esses dados consegue-se analisar o comportamento dos alunos durante o semestre e verificar o motivo que levou o aluno a evadir ou reprovar.

As instituições de ensinos que utilizam o AVA como apoio no ensino, tem a possibilidade de realizar a predição final do rendimento dos alunos, por meio da análise dos dados gerados utilizando técnicas Mineração de Dados Educacionais (do inglês, (Educational Data Mining) - EDM) que é uma área que explora os dados educacionais através de métodos e algoritmos para extrair informações significativas, cujo o objetivo é compreender e prever o comportamento dos estudantes, permitindo adaptar o processo de aprendizagem de acordo com cada situação[Costa et al., 2012]. Neste contexto,[Einhardt et al., 2016] apontam para a importância da predição de estudantes em risco, pois esta possibilita que o professor resgate o aluno antes que ocorra uma reprovação.

2.1 Problemática

A Educação a Distância (EAD) no Brasil tem se consolidado com diversos estudantes optando por essa modalidade de ensino para ampliar suas formações e realização profissional. Em 2012, segundo dados do Censo EAD 2014 [ABE, 2014], foram ofertados 9.376 cursos. Um total de 5.772.466 matrículas foram registradas, com um 52,5% de aumento em relação a 2011. Já o número de conclusões, em 2012, foi de 1.589.374. Esses dados mostram a evolução da Educação a Distância e como ela está se tornando uma ferramenta muito importante na formação dos cidadãos brasileiros. Contudo, a EAD ainda enfrenta alguns obstáculos a serem ultrapassados. Resistência de educandos e educadores, desafios organizacionais e custos de produção são alguns deles, mas, sem dúvida, um dos maiores obstáculos é a evasão de alunos dos cursos e instituições. De acordo com o Censo EAD 2013 [ABE, 2014], a evasão em cursos a distância autorizados pelo Ministério da Educação chegou a atingir um pico de 20,5% em 2011, tendo sido de 18,6% em 2010 e de 11,74% em 2012. Um dos principais diferenciais de cursos EAD é a grande quantidade de dados gerada pelas interações no ambiente educacional, o que abre novas possibilidades para estudar e compreender estas interações [Hämäläinen and Vinni, 2010] [Mishra et al., 2014]. Algumas áreas de pesquisas surgiram nos últimos anos com intuito de auxiliar em questões como essas. A Mineração de Dados Educacionais (do inglês Educational Data Mining - EDM) é uma área de pesquisa interdisciplinar que lida com o desenvolvimento de métodos para explorar dados originados no contexto educacional [Romero and Ventura, 2010]. Juntamente com a EDM temos a Analítica de Aprendizagem (do inglês Learning Analytics - LA) que busca medir, coletar, analisar e reportar dados sobre estudantes e seus contextos, com o propósito de entender e otimizar seu aprendizado e o ambiente onde ele ocorre (Brown, 2012). As interações dos alunos e professores com os ambientes virtuais de aprendizagem (AVAs) proveem os dados que alimentam as pesquisas nessas áreas e possibilitam a descoberta de novos conhecimentos. Um dos grandes desafios dos pesquisadores é desenvolver métodos capazes de prever o comportamento dos estudantes, de modo a possibilitar a intervenção de professores/tutores, ou demais envolvidos, visando identificar estudantes em risco e intervir pedagogicamente antes que os mesmos reprovem [Macfadyen and Dawson, 2010]. A possibilidade de identificação de estudantes com alto risco de evasão e reprovação a partir de modelos automáticos foi apontada por Baker et al. [Baker et al., 2011] como um dos potenciais problemas a serem atacados pela comunidade brasileira que atua na área de mineração de dados educacionais. O presente projeto de pesquisa tem por objetivo desenvolver e testar técnicas e ferramentas para a modelagem e identificação de acadêmicos em risco em cursos de educação a distância. Se por um lado, o campo de pesquisa em mineração de dados educacionais já vem explorando de maneira exaustiva o tema de identificação de acadêmicosem risco, é certo também que os resultados dessas pesquisas ainda não foram traduzidos em ferramentas concretas que possam ser insta-

ladas e utilizadas pelos profissionais da educação. Os resultados publicados normalmente são restritos a experimentos realizados com conjuntos de dados específicos e cujos modelos não podem ser extrapolados para outros contextos. Nosso objetivo aqui é avaliar a possibilidade de gerar modelos extrapoláveis a diversos contextos, além de comparar os desempenhos de tais modelos com outros que utilizem informações mais específicas e que eventualmente não estarão disponíveis em qualquer cenário. Ainda, os modelos desenvolvidos serão disponibilizados para teste e avaliação por parte de professores de cursos de educação a distância por meio de um plugin para navegador. A presente proposta é uma continuação dos seguintes trabalhos já desenvolvidos e em desenvolvimento pela equipe executora na área de predição de estudantes em risco e Learning Analytics: [Cechinel et al., 2015][Detoni et al., 2014] [Queiroga et al., 2015][Einhardt et al., 2016][Rodriguez, 2015]. O presente projeto também pretende contrastar abordagens diferentes para a modelagem e identificação de acadêmicos em risco em cursos de educação a distância. Precisamente, pretende-se comparar o desempenho de modelos de predição gerados a partir dos diferentes tipos de interações em AVAs de acordo com a definição de presença proposta por [Swan, 2003], sendo elas: 1) interações dos acadêmicos com o curso (presença cognitiva), e 2) interações sociais dos acadêmicos no AVA (presença social e pedagógica.

2.2 Objetivos

2.2.1 Objetivo Geral

- Comparar a utilização de diferentes conjuntos de dados para a modelagem e identificação de acadêmicos em risco em cursos de programação

2.2.2 Objetivos Específicos

- Gerar e testar modelos para a predição de acadêmicos em risco em cursos de programação semipresenciais com diferentes bases de dados disponíveis
- Avaliar se modelos gerados a partir da simples contagem de interações dentro de AVAs são extrapoláveis aos diferentes contextos das bases de dados disponíveis
- Comparar modelos gerados a partir da simples contagem de interações com modelos que utilizam atributos derivados da simples contagem.
- Quantificar em termos de acurácia as diferenças entre os diversos modelos gerados
- Disseminar resultados do projeto em eventos e revistas científicas

3 Fundamentação teórica

3.1 Mineração de dados educacionias

Atualmente, vem aumentando a quantidade de instituições de ensino utilizando tecnologias em cursos semipresenciais, permitindo produzir conteúdos didáticos e acompanhar desempenho dos alunos a distância [Santos et al., 2016]. O uso destas tecnologias, como os Ambientes Virtuais de Aprendizagem (AVA), geram grandes quantidades de dados, contendo informações das interações realizadas, dados dos alunos e professores, entre outras informações [Guércio et al., 2014]. Esses dados armazenados podem ser analisados para descobrir problemas presente em determinadas disciplinas ou curso com objetivo de melhorar o desempenhos dos alunos. Para analisar dados educacionais utiliza-se técnicas de EDM (Mineração de Dados Educacionais), que busca desenvolver métodos que sejam capaz de identificar informações no AVA, com objetivo de entender comportamentos dos alunos e aspectos que influenciam nos seus desempenhos [Santos et al., 2012].

Segundo [Baker et al., 2011] a EDM tem sido aplicada na área da educação com o objetivo de fornecer resultados precisos para professores prevendo estudantes em riscos de evasão e/ou reprovação. Para reduzir as altas taxas de reprovações, alguns autores sugerem diferentes práticas pedagógicas [[Mow, 2006],[Gomes, 2010]]. Muitas dessas práticas tem 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.

Para analisar a eficácia e a eficiência de estudantes têm frequentemente associadas as diferentes medidas de suas interações dentro dos AVA, e normalmente possuem uma alta correlação com o sucesso dos mesmos nos cursos. Por exemplo,[Murray et al., 2013] observaram que estudantes que apresentaram altas taxas de acesso aos conteúdos no AVA de um curso a distância, também receberam as melhores notas, e [Dickson, 2005] descobriu que o número total de cliques dados por estudantes é fortemente correlacionado com as suas notas finais em um curso. [Garrison et al., 1999]projetaram um framework conceitual identificando os elementos críticos necessários para uma experiência educacional mediada por computador obter sucesso.

Técnicas de EDM 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 tem mencionado métodos de EDM

para prever alunos em risco devido a necessidade de melhorar a qualidade dos processos de ensino e aprendizagem, e também de compreender a situacão do rendimento dos estudantes [Khobragade, 2015]. Entre os exemplos de trabalhos de identificacão 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 predicão para evitar fracassos de alunos no decorrer do semestre utilizando diferentes algoritmos de classificacão aplicados a dados coletados das interacões dos academicos em um AVA. 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 experimentacões com diferentes configuracões dos atributos coletados.

4 Metodologia e desenvolvimento

Para atingir os objetivos deste projeto de pesquisa, os procedimentos metodológicos que foram executados são os seguintes:

- **Recolhimento de dados :** Foram recolhidos dados referentes aos logs de acesso de alunos e professores nos AVA do curso de Tecnologias da Informação e Comunicação na disciplina de Algoritmos de Programação da Universidade Federal de Santa Catarina .
- **Pré-processamento dos dados e geração das bases de dados iniciais :** Os dados foram extraídos dos registros de acesso dos cursos fornecidos pela plataforma Moodle. Destes registros foi extraídos para cada interação de cada usuário, o identificador do usuário, a interação e a data e hora da interação. A ideia principal do projeto é a comparação entre diferentes abordagens e uso de dados para a geração de modelos de predição de alunos em risco. Nesse sentido, diferentes bases de dados iniciais serão geradas a partir dos dados extraídos da disciplina de Algoritmos de Programação. Cada aluno será rotulado como aprovado ou reprovado na disciplina do curso de acordo com a situação informada nos registros acadêmicos. As interações de cada aluno serão agrupadas semanalmente. Assim, cada aluno da disciplina será descrito por um vetor de n atributos, indicando o número total das interações do aluno em cada semana. Considerando que o objetivo principal é predizer a situação final do aluno com a maior antecedência possível na disciplina, serão gerados alguns conjuntos de dados com base nas combinações incrementais das n semanas de interações existentes. Assim, serão gerados conjuntos de dados contendo somente a primeira semana de interações, contendo a primeira e a segunda semana, e assim por diante, até ser gerado o conjunto de dados com todas as n semanas de interações. Um conjunto de dados consistirá em todos os exemplos de alunos de um dado semestre de um curso. A seguir são descritas os tipos de informa-

ções específicas que serão geradas em cada uma das bases de dados que servirão para geração dos modelos de predição.

- **Bases de dados geradas**
  - **Base de dados I (BDI)- Contagem de interações:** Conterá as informações da contagem simples da quantidade de interações que os estudantes realizaram no AVA. A motivação por trás desta decisão é a possibilidade de permitir que a metodologia proposta seja generalizável a diferentes cursos, plataformas e execuções, sem gerar uma dependência em atributos excessivamente específicos. Essa base de dados não pertence de maneira completa a uma categoria de presença específica, uma vez que contém interações de todos os tipos existentes.
  - **Base de dados II (BDII)- Contagem de interações e atributos derivados:** Essa base de dados será formada pela contagem de interações simples, juntamente com alguns atributos que podem ser gerados a partir da própria variável principal de contagem, como a média e mediana. A mediana foi calculada baseada no número de interações semanais, por exemplo no primeiro dia da semana 1 o aluno realizou 1 interação no segundo dia 10, assim sucessivamente. Já a média é calculada utilizando o total de interações semanais, por exemplo na semana 1 o aluno obteve 10 interações então sua média será 10, na semana 2 não realizou interações então sua média será 5, estes cálculos são realizados até a última semana de aula. Essa base de dados ainda pode ser considerada como generalizável, uma vez que os atributos adicionais são extraídos diretamente da contagem pura das interações.
  - **Base de dados III (BDIII) - Contagem de interações , atributos derivados e questionário de perguntas :** Essa base de dados será formada por BDII juntamente com um questionário de perguntas Apéndice A, na qual, os alunos irão responder no ambiente virtual da disciplina.
- **Mineração dos Dados - Geração e teste dos modelos de predição :** Para cada curso, os dados serão divididos em conjuntos de treinamento e de teste. Essa divisão será feita de duas formas para testar duas situações distintas. Na primeira forma, denominada ”entre turmas”, a divisão é feita dentro de um mesmo semestre do mesmo curso, utilizando turmas distintas para treino e teste. Na segunda forma, denominada ”entre semestres”, a divisão é feita entre semestres distintos. Para tanto, cada conjunto de dados será dividido aleatoriamente em dois conjuntos contendo a mesma quantidade de alunos, ou seja, cada turma de alunos será transformada em duas turmas (A e B). No primeiro formato (entre turmas) uma turma (ex: turma A) de um semestre de um curso será utilizada para treinar o modelo, e a outra turma (ex: turma B) daquele semestre daquele curso será utilizada para avaliar o modelo (e vice-versa). No segundo formato (entre semestres), cada uma

das turmas (A e B) de cada semestre será utilizada para treinar os modelos, e os mesmos serão avaliados com as turmas A e B do semestre seguinte (e vice-versa). Esse processo de divisão será repetido 5 vezes, gerando conjuntos aleatórios diferentes a cada repetição. As médias dos resultados da aplicação dos classificadores nestas repetições serão reportadas. Os seguintes modelos serão aplicados aos dados: Rede Bayesiana, Rede Neural (Perceptron de Múltiplas Camadas), C4.5 e Floresta Aleatória. Para o desenvolvimento foi utilizado a linguagem de programação python 3, juntamente com frameworks pandas para processamento de dados, scikit para criação de modelos de predição.

4.1 Fases do desenvolvimento do projeto

Serão detalhadas como cada uma das fases executadas considerando os materiais e métodos utilizados.

F.0: Recolhimento dos dados

Na fase 0 foi extraído dados de acesso dos cursos fornecidos pela plataforma Moodle da Universidade Federal de Santa Catarina no curso de Tecnologias da Informação e Comunicação na disciplina de Algoritmos de Programação.

F.1: Pré-processamento e geração de base de dados

Na fase 1 será realizado o pré-processamento dos dados coletados do Moodle, para cada alunos serão extraídos as interações. A ideia principal do projeto é a comparação entre diferentes abordagens e uso de dados para a geração de modelos de predição de alunos em risco.
Cada aluno será rotulado como aprovado ou reprovado em cada disciplina dos cursos de acordo com a situação informada nos registros acadêmicos.
As interações de cada aluno serão agrupadas semanalmente. Assim, cada aluno em uma disciplina será descrito por um vetor de n atributos indicando o número total de interações daquele aluno em cada semana, mostrado na Tabela 2: interações semanais, chamada de base de dados I.

Nome	Semana 1	Semana 2	. . .	Semana (n)	Situação Final
Aluno 1	10	20	**. . .**	20	Aprovado
Aluno 2	1	20	**. . .**	20	Reprovado
. . .	. . .	. . .	. . .	. . .	. . .
Aluno m	12	2	**. . .**	20	Aprovado

Tabela 2: Interações semanais

Já na Tabela 4 : Atributos derivados e interaçõe mostra a base de dados II, na qual foi criado os atributos derivados das interações como média e mediana, descrito na sessão de métodos e materiais.

Nome	**S1**	**Média 1**	**Mediana 1**	**S2**	Média 2	Mediana 2	. . .	**S (n)**	Média (n)	Mediana (n)	**Situação Final**
Aluno 1	10	10	5	20	15	9	**. . .**	20	30	10	Aprovado
Aluno 2	1	1	5	20	11.5	7	**. . .**	20	10	8	Reprovado
.	**.**			**.**				**.**	.	.	**.**
.	**.**			**.**			**. . .**	**.**	.	.	**.**
.	**.**			**.**				**.**	.	.	**.**
Aluno m	12	12	**7**	2	7	3	**. . .**	20	30	8	Aprovado

Tabela 4: Atributos derivados e interações

Considerando que o objetivo principal é predizer a situação final do aluno com a maior antecedência possível dentro das disciplinas, serão gerados diversos conjuntos de dados com base nas combinações incrementais das n semanas de interações existentes para a disciplina. Assim, serão gerados conjuntos de dados contendo somente a primeira semana de interações, contendo a primeira e a segunda semana, e assim por diante, até ser gerado o conjunto de dados com todas as n semanas de interações. Um conjunto de dados consistirá em todos os exemplos de alunos de um dado semestre de um curso. A seguir são descritas os tipos de informações específicas que serão geradas em cada uma das bases de dados que servirão para geração dos modelos de predição.

F.2: Mineração de dados geração e testes dos modelos de predição

Nesta etapa foram gerados modelos de predição utilizando a Linguagem Python com a biblioteca Scikit-Learn. Os algoritmos utilizados foram Naive Bayes, Random Forest, Decision Tree, AdaBoost e Multilayer Perceptron. Foram realizados experimentos buscando validar os modelos de predição. Nos experimentos utilizaram a contagem das interações e atributos derivados que são uma abordagem para aumentar os conjuntos de entrada. Além, da técnica Cross-Validation de 10 folds (CV K-folds) e Leave-one-out cross-validation (LOOCV), que foram aplicadas em todas as turmas apresentadas na Tabela 1 (A1, A2,A3 e A4). Os dados pré-processados foram separados em semanas com objetivo de realizar testes com Cross Validation. E, como formas de avaliações, utilizou-se algumas métricas, sendo elas: taxa de verdadeiro positivo, verdadeiro negativo e área sob a curva ROC (Receiver Operating Characteristic). A taxa de verdadeiro positivo mostra como resultado os alunos classificados corretamente como aprovados, já a taxa de verdadeiro negativo apresenta os alunos classificados corretamente como reprovados. A curva ROC é um gráfico de sensibilidade (taxa de verdadeiro positivo - TVP) no eixo y por especificidade (taxa de falso positivo - TFP) no eixo x. A área sob a curva ROC classifica corretamente os alunos que obtiveram o estado final como reprovado e aprovado. Segundo Khouli et al. (2009) os valores considerados bons classificadores são aqueles que obtiveram valores acima de 0.7 ao longo da semana.

F.3: Comparação dos modelos e análise de resultados

Nesta etapa será analisado os resultados obtidos com os modelos de predição, para essa análise será utilizado métricas como verdadeiro positivos, verdadeiros negativos e curva roc.

F.3 Estudo do Dashboard para futura integração dos modelos de predição

Buscando fazer uma integração dos melhores modelos de predições generalizáveis em um dashboard de Analítica de Aprendizagem para dados de logs do Moodle. O Moodle foi escolhido por ser o AVA utilizado oficialmente pelos cursos que fornecerão os dados para as análises. O dashboard em questão já está em fase de implementação e atualmente apresenta para o professor alguns dados pré-processados gerados a partir dos logs de acesso as disciplinas ministradas pelos professores [Einhardt et al., 2016] . Abaixo são apresentadas a figura 1 e figura 2 do sistema que terá como novas funcionalidades os modelos de predição. Outras funcionalidades que o sistema apresentará estão relacionadas com dados básicos de acesso aos cursos virtuais, assim como estatísticas descritivas sobre esses acessos. Exemplos de informações que serão possíveis de visualizar no dashboard são: Total de visualizações de páginas por dia, total de usuários únicos por dia, total de atividades únicas por dia, tempo médio das sessões, taxas de rejeição, total de atividades por hora e dias da semana.

F.3 Desenvolvimento de artigos

Com os resultados obtidos foi publicado o dois artigos,sendo eles Identificação precoce de estudantes em risco de reprovação em disciplinas semipresenciais de programação de computadores e apresentado no II Simpósio Ibero-Americano de Tecnologias Educacionais- SITED 2018. E Comparacão de diferentes configuracões de bases de dados para a identificacão precoce do risco de reprovacão: o caso de uma disciplina semipresencial de Algoritmos e Programacão, foi enviado para XXIX Simpósio Brasileiro de Informática na Educação (SBIE), sendo aprovado para publicação .

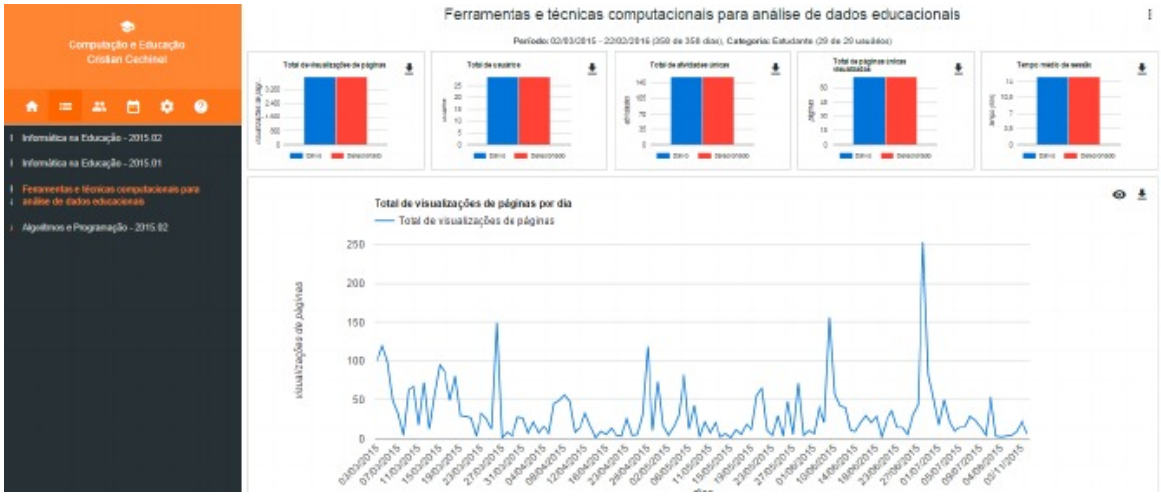

Figura 1: Moodle Dashboard

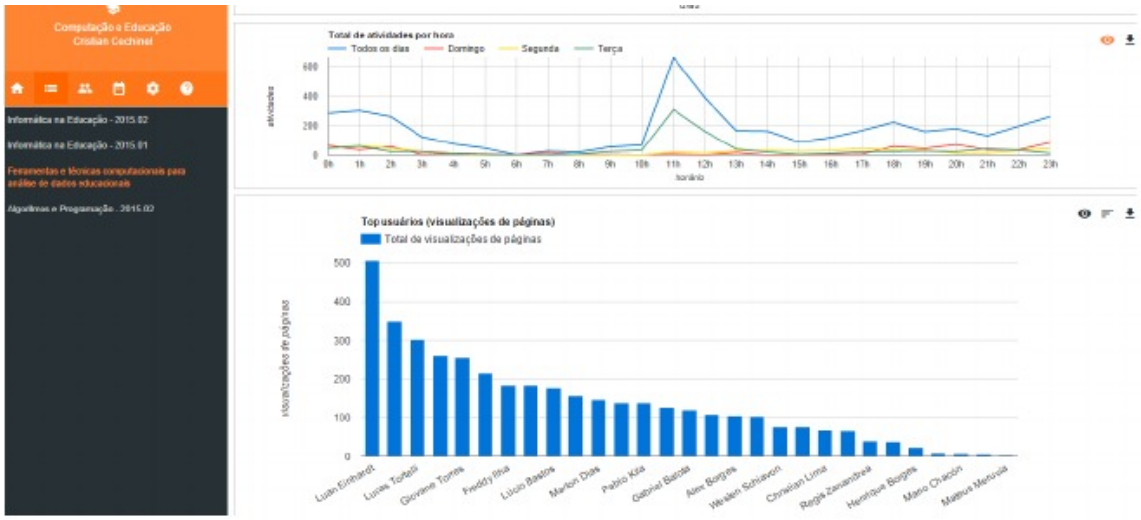

Figura 2: Moodle Dashboard

4.2 Desenvolvimento e detalhamento dos experimentos

Para o desenvolvimento deste trabalho os dados utilizados foram coletados da plataforma Moodle da UFSC Campus Araranguá, de cursos semipresenciais.

Os dados coletados são da disciplina de Algoritmos e Programação que ocorreu no ano de 2016 e 2017, ministradas nos 4 semestres resultando em 4 arquivos de logs. A disciplina é baseada na linguagem de Programação Python, sendo o primeiro contato dos alunos com linguagem de programação na universidade. O objetivo da disciplina é preparar o aluno para desenvolver soluções computacionais através da aplicação dos conceitos de lógica de programação e desenvolver programas com linguagem de alto nível estruturada.

A disciplina possui 102 horas-aulas semanais distribuídas em 6 horas-aulas por semanas, destas 6 horas-aulas 2 horas são destinadas as atividades que são realizadas a distância. As atividades a distância são disponibilizadas na plataforma Moodle contendo vídeos-aulas, textos de apoio, a atividade que deve ser realizada e um desafio. Já nas aulas presenciais é apresentado o conteúdo e realizado atividade prática no computador, utilizando a Linguagem de Programação Python.

A disciplina analisada tem duração de 17 semanas, que equivale a um semestre. A situação final do aluno é estabelecida pelo seu rendimento nas atividades realizadas durante o semestre, sendo atribuído um estado aprovado ou reprovado.

A criação do modelo é baseada apenas em interações dos alunos no AVA, sendo de fácil aplicação em outros cursos da universidade. Com isso, a metodologia foi divida em 4 etapas, sendo elas coleta de dados, pré-processamento, geração e avaliação de modelo de predição e análise dos resultados obtidos.

4.3 Fase 0: Coleta de dados

Nesta etapa realizou-se a coleta de dados do Moodle, o arquivo inicialmente gerado é formado pela quantidade de logs dos alunos. Os logs são cada interação que um determinado aluno realizou no AVA, na disciplina determinada.

A Tabela 6 apresenta a quantidade de interações realizadas na disciplina de Algoritmos e Programação durante o ano de 2016 e 2017, que obteve 99.977 interações. Ainda é possível visualizar a quantidades de alunos em cada turma e o total de aprovados e reprovados.

Tabela 6: Dados Coletados

Semestre	Turma	Quantidade de interações	Alunos	Aprovados	Reprovados
2016-1	A1	12.775	26	7	9
2016-2	A2	15.118	24	6	18
2017-1	A3	34.684	28	10	28
2017-2	A4	37.400	33	12	21
Totais		99.977	121	35	86

4.4 Fase 1: Pré-processamento e geração de base de dados

Para o pré-processamento dos dados coletados foi utilizada a biblioteca Pandas e desenvolvido um sistema em Python, com objetivo de automatizar esta etapa. O sistema efetua a leitura dos arquivos exportados do Moodle e gera um novo arquivo.

O arquivo gerado no pré-processamento contém o nome do aluno e a quantidade de intera-

ções realizada ao longo da semana no AVA, por exemplo, Aluno1 na primeira semana realizou 1 interação, já na segunda não acessou o Moodle, na terceira semana teve 25 interações e assim ao longo das 17 semanas de aula.

Com a tabela de notas coletadas, o arquivo pré-processado e com a quantidade de interações, foi desenvolvido um sistema em Python para automatizar a atribuição do resultado final. O sistema extraí as notas finais dos alunos e insere no arquivo pré-processado, criando dois campos com notas e situação final do aluno.

Além dos dados serem pré-processados somente com interações semanais dos alunos, foi gerado um conjunto de atributos derivados conforme apresenta a Tabela 7. Os atributos escolhidos são procedentes do total de interações realizadas no AVA.

Para realizar os cálculos dos atributos derivados, foi desenvolvido um sistema em Python utilizando as bibliotecas pandas e numpy. A mediana foi calculada baseada no número de interações semanais, por exemplo no primeiro dia da semana 1 o aluno realizou 1 interação no segundo dia 10, assim sucessivamente. Já a média é calculada utilizando o total de interações semanas, na semana 1 o aluno obteve 10 interações então sua média será 10, na semana 2 não realizou interações então sua média será 5, estes cálculos são realizados até a ultima semana de aula.

Tabela 7: Atributos Derivados

Atributos	Descrição
Interações por semanas (1 a 17)	Número de interação na semana
Média	Média de interação na semana
Mediana	Mediana de um conjunto de interação por semana

Também foi gerado uma base de dados retirada do questionário mostrado no apéndice A aplicado pelo professor. Os dados foram utilizados para geração de modelos de predição juntamente com as bases de dados descritas anteriormente.

4.5 Fase 2: Geração e testes de modelos de predição

Nesta etapa foram gerados modelos de predição utilizando a Linguagem Python com a biblioteca Scikit-Learn, esta biblioteca fornece modelos de algoritmos implementados open source com a possibilidade de alteração de parâmetros, sendo possível a fácil comparação entre os algoritmos testados para criação dos modelos.

Os algortimos utilizados foram Naive Bayes, Random Forest, Decision Tree, AdaBoost e Multilayer Perceptron. Naive Bayes é uma técnica de classificação baseada no teorema Bayes. O classificador Naive Bayes assume uma característica particular, que pode ser usada em

casos como marcar emails como spam ou não spam, para verificar emoções negativas ou positivas em um texto e como neste trabalho classificar alunos como aprovados ou reprovados.

Já o Decision Tree usa modelos preditivos que utilizam treinamento para classificação e predição de dados [Costa et al., 2012], esse algoritmo é utilizado geralmente em estatísticas, mineração de dados e aprendizado automático. O algoritmo Random Forest (do português, floresta aleatória) é um método de mineração de dados baseado em Decision Tree, mas prossegue criando muitas árvores formando uma floresta aleatória.

O AdaBoost é um algoritmo baseado no Boosting, no qual gera um conjunto de hipóteses e as combina por meio de votação ponderada, as hipóteses são geradas por meio do treinamento de classificadores usando uma distribuição dos dados iterativamente ajustada.

O Multilayer Perceptron (MLP) é uma rede neural que é composta por mais de uma camada de neurónios artificiais, e utiliza aprendizagem supervisionada que treina conjuntos de dados.

Foram realizados em um total de 54 experimentos buscando validar os modelos de classificadores, gerando 2 cenários de experimentos. O primeiro utilizou apenas interações da plataforma Moodle e no 2 acrescentou os atributos derivados.

No primeiro cenário foi utilizado Cross Validation de 10 folds (CV K-folds) e Leave-one-out cross-validation (LOOCV) . De acordo com [Queiroga et al., 2016] Cross Validation consiste em dividir a base de dados em 10 partes, sendo 9 partes utilizadas para treinamento e 1 para teste. Este processo é repetido 10 vezes, assim o conjunto de treino é diferente do conjunto de testes. O Cross Validation foi aplicado nas turmas na Tabela 1. Já Leave-one-out cross-validation foi aplicado em cada uma das turmas apresentadas na Tabela 1, que consiste em dividir o conjunto amostrado em parte de treino e teste, por exemplo se o conjunto possui 30 amostras o treinamento terá 29 e o conjunto de teste 1. Ainda foram realizadas combinações de conjuntos de treino em turmas de semestres diferentes e testado em outras turmas, por exemplo utilizou-se as turmas A1, A3 e A4 como conjuntos de treino e testou na Turma A2. Realizou -se outras combinações de conjuntos de treinamento e conjuntos de teste treinando A4 e testando em A2, assim por diante.

No segundo cenário foram realizados os mesmos procedimentos do cenário 1, porém utilizando atributos derivados das interações conforme apresenta a Tabela 2. Os atributos derivados é uma abordagem para aumentar os conjuntos de variáveis de entrada, ou seja, possuindo mais conjuntos de entrada referente ao estado final do aluno.

Os dados pré-processados foram separados em semanas, na qual realizaram em todos os experimentos testes semanais com Cross Validation. Na Semana 1 foi aplicado e obteve uma taxa de verdadeiro positivo e verdadeiro negativo, na semana seguinte foi feito testes na Semana 1 e Semana 2 obtendo uma nova taxa de verdadeiros positivos e verdadeiros negativos,

assim sucessivamente.

Como formas de avaliações foram utilizadas métricas como taxa de verdadeiro positivo , verdadeiro negativo e área sob a curva ROC (Receiver Operating Characteristic). A taxa de verdadeiro positivo tem como resultado os alunos classificados corretamente como aprovados, já a taxa de verdadeiro negativo apresenta os alunos classificados corretamente como reprovado. A curva ROC é um gráfico de sensibilidade (taxa de verdadeiro positivo - TVP) no eixo y por especificidade (taxa de falso positivo - TFP) e no eixo x. A área sob a curva ROC classifica corretamente os alunos que obtiveram o estado final como reprovado e aprovado, segundo [El Khouli et al., 2009] os valores considerados bons classificadores são aqueles que obtiveram valores acima de 0.7 ao longo da semana.

Os experimentos tem como objetivo principal validar os modelos criados, dessa forma identificar quais possuem um bom classificador para prever a situação final do aluno.

5 Análise de resultados

A Figura 3 mostra os resultados da área sob a curva ROC aplicado na Turma A4, sendo 1t experimento do cenário 1 utilizando Leave-one-out cross-validation (LOOCV) . Os métodos que obtiveram melhores resultados para identificar alunos como reprovados foram Random Forest e Naive Bayes. Podemos observar no gráfico que a área sob a curva ROC do Random Forest a partir da semana 7 foi acima de 0.81, logo em seguida o valor chega a aproximadamente 0.91 . Já o Naive Bayes na semana 6 obteve um valor da área sob a curva ROC acima de 0.80 mantendo-se constante ao longo das semanas.

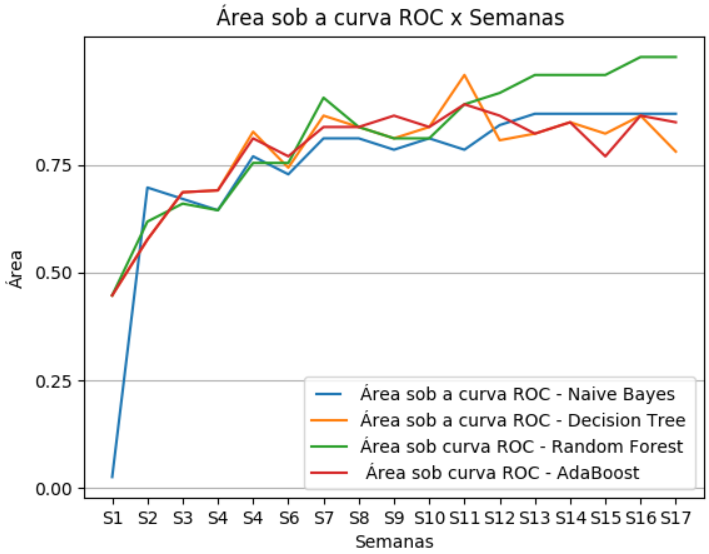

Figura 3: Resultados Área sob curva ROC - Turma A4 (LOOCV)

Na Figura 4 mostra as taxas de verdadeiros negativos e verdadeiros positivos ao longo das semanas para os mesmos experimentos. Os melhores métodos utilizados para identificar os alunos como reprovados e aprovados, foram o Naive Bayes e Random Forest. Foi possível observar que a taxa de verdadeiro negativo (VN) do Naive Bayes entre as semanas 4 e 6 variaram entre 70% a 75%, logo em seguida ocorreu um crescimento chegando a 80% de probabilidade de identificar um aluno reprovado. Na semana 11 teve um pico de 99%, porém este algoritmo não apresenta uma ótima taxa de verdadeiros positivos (VP), como podemos verificar na 4 na semana 2 a taxa de verdadeiros positivos é 89% de precisão para identificar alunos aprovados, mas ao longo do semestre reduziu para 75% mantendo-se constante. Já o Random Forest na semana 4 apresentou uma probabilidade de 70% para identificar um aluno reprovado, na semana 7 a taxa de acerto ficou acima de 75% mantendo-se constante até a semana 12, ocorrendo um crescimento até atingir 99% nas últimas semanas. Já nas taxas de verdadeiros positivos o Random Forest na semana 1 obteve 75% , mantendo-se acima de 80% na semana 5.

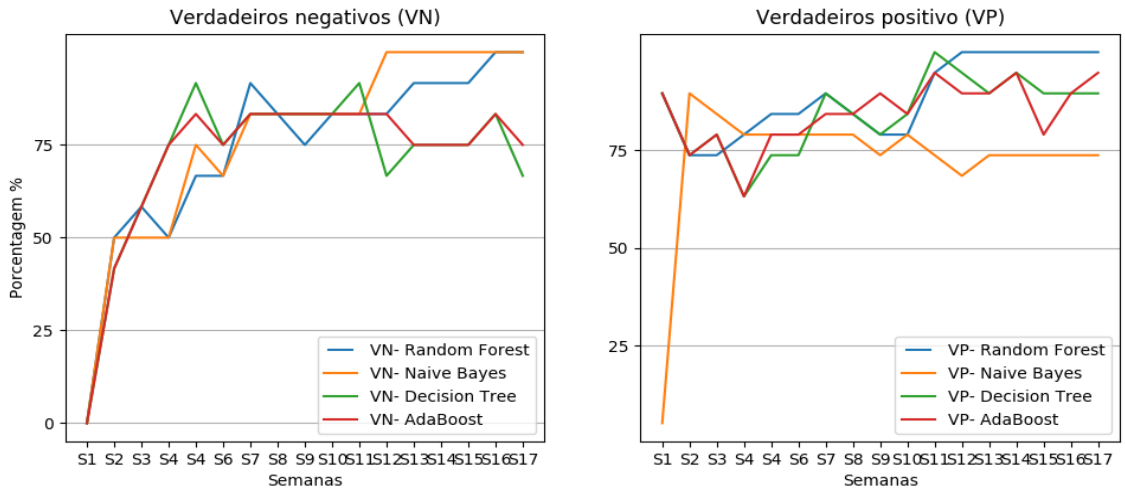

Figura 4: Resultados Verdadeiros negativos & Verdadeiros positivos - Turma A4 (LOOCV)

Foram realizados experimentos utilizando combinações de turmas de semestre diferentes para treinar os modelos testando em outras. Alguns resultados para área sob a curva ROC estão apresentados na 5. O Naive Bayes na semana 2 consegue prever os alunos reprovados com um valor acima de 0.75 para área sob a curva ROC, mantendo assim uma média ao longo das semanas acima de 0.76. Já o Random Forest foi treinada na Turma A3 e testada na A4, mantendo uma média 0.86 para o valor da área sob a curva ROC.

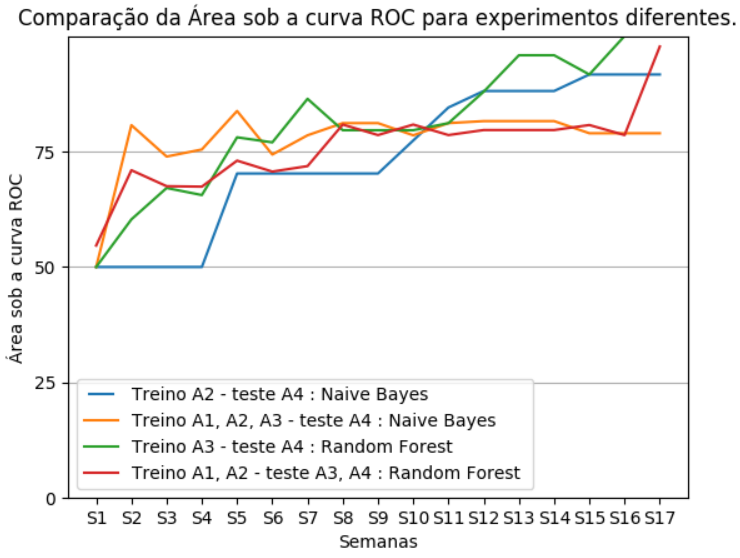

Figura 5: Resultados Área sob curva ROC para vários conjuntos treino e teste

No cenário 2 foi utilizado atributos derivados para realizar novamente os experimentos com o objetivo de melhorar os modelos de predição. A Figura 6 mostra as taxas de VP e VN para o

modelo utilizando Random Forest (RF) CV 10Folds com e sem atributos aplicado na turma A4. Foi observado que utilizando atributos nas semanas iniciais atingiu-se 70% , e na semana 7 chegou a 80%, mantendo um crescimento linear ao longo do semestre. Sem os atributos a taxa ficou acima de 70%, porém ocorreu grandes oscilações atingindo um valor de 80% nas últimas semanas do semestre. Já os verdadeiros positivos pode-se perceber que utilizando atributos derivados a partir da semana 6 obteve-se uma porcentagem acima de 80% e sem atributos derivados ocorreu uma variação voltando para mesma porcetagem na semana 11.

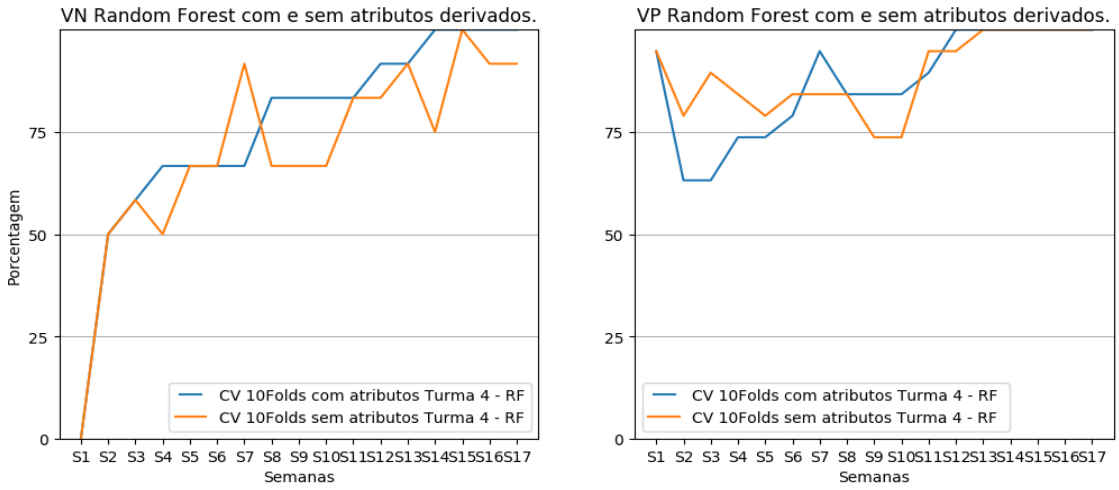

Figura 6: Comparação VN & VP para o Random Forest com e sem atributos derivados

Com base no cenário 2 foram criados modelos de treinamento utilizando atributos derivados com a Turma A3, e testados com a Turma A4 utilizando o algoritmo Naive Bayes. Em seguida realizou-se a comparação com o cenário 1. Conforme mostra a Figura 7, o algoritno Naive Bayes utilizando atributos derivados obteve 95% de certeza para identificar um aluno reprovado na semana 6, já sem atributos foi considerado apenas as interações que partir da semana 10 conseguiu-se chegar o mesmo valor. Ambos mantiveram-se estáveis após atingir esse valor. Percebe-se que a taxa de verdadeiro positivo para os algoritmos utilizando atributos derivados manteve-se acima de 80% na semana 6, já sem atributos este resultado foi obtido na semana 12.

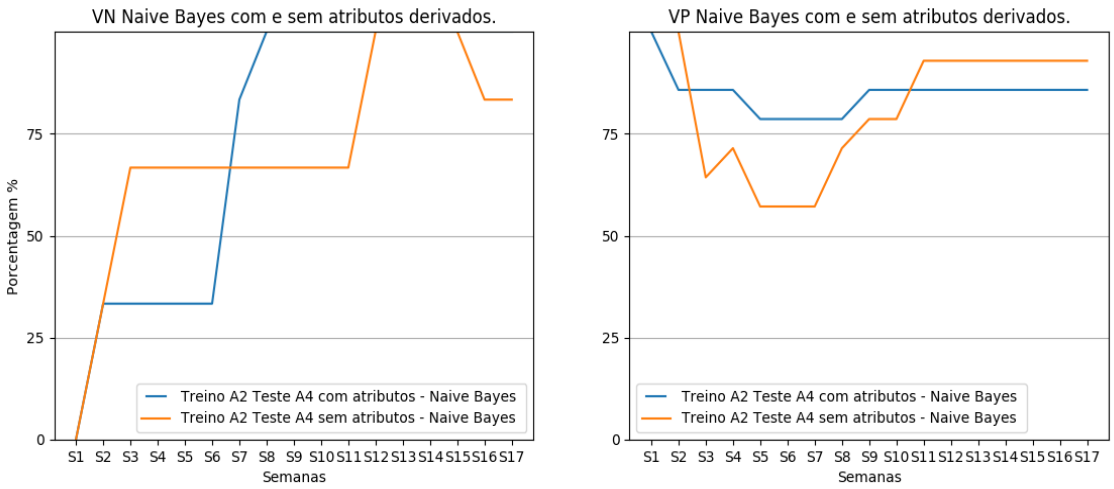

Figura 7: Comparação VN & VP para o Naive Bayes com e sem atributos

Os resultados obtidos nos cenários 1 e 2 alcançaram uma taxa de acerto igual ou acima de 80% no final do semestre. Os algoritmos Naive Bayes e Random Forest foram os que obtiveram melhores resultados nos experimentos em ambos os cenários. Na maioria dos testes realizados os modelos que apresentaram melhores desempenhos foram as turmas A3 e A4. No cenário 1 o Naive Bayes teve uma taxa de acerto igual ou acima de 75% nas primeiras semanas e o Random Forest na semana 7 obteve igual ou acima de 75% a taxa de acerto mantendo-se constante nos dois algoritmos.

6 Atividades do Bolsista

A seguir são apresentadas as atividades do bolsista **Matheus Francisco Batista Machado** conforme descritas no plano de atividades definido no projeto. Para cada atividade é indicado se a mesma foi realizada ou não. Em caso negativo é provida uma explicação.

1. **Estudo e pesquisa bibliográfica:** Revisão da literatura científica relevante para o desenvolvimento do projeto. [Atividade Realizada].

2. **Recolhimento de dados da disciplina de Algoritmos e Prorgamação do Moodle UFSC:**

   Recolhimento de dados do ambiente virtual da disciplina de Algoritmos e Programação do moodle da UFSC. [Atividade Realizada]

3. **Estudo da linguagem de programação Python e biblioteca pandas para realizar o Pre-processamento dos Dados:**

Estudo da linguagem de programção python e blibioteca pandas para o desenvolvimento do pre-processamento de dados e realização do pré-processamento dos dados com os logs do ambiente virtual. [Atividades Realizadas]

4. **Geração das bases de dados utilizadas para a criação dos modelos de predição:**

   Para geração das bases de dados foi estudado a biblioteca pandas python, assim foi possível realizar a geração da base de dados para a criação dos modelos preditivos. [Atividade Realizada]

5. **Estudo da biblioteca sckit-learn para utilizar algoritmos para criar os modelos de predição:**

   Para geração dos modelos de predição foi estudado a biblioteca scikit-learn.org, buscando utilizar modelos de predições como: Naive Bayes, Random Forest, Decision Tree e Multi Layer Perception. Assim foi criado os modelos de predições utilizando as bases de dados geradas a partir do pré-processamento dos dados. [Atividade Realizada]

6. **Análise dos resultados e comparações dos modelos:**

   Na análise dos resultados foi utilizado métricas verdadeiros negativos, verdadeiros positivos e curva roc. A análise de alguns resultados estão descritas na seção 5 [Atividade Realizada]

7. **Estudo do dashboard e iniciando integração com os modelos de predição:**

   Para iniciar a integração dos algoritmos de predição desenvolvido na linguagem de programação python com o dashboard moodle, foi necessário estudar as seguintes linguagem de programação e frameworks: coffescript , javascript , css, html e grunt.

   Devido o aluno não ter conhecimento prévio sobre as linguagens de programação, em paralelo realizar a escrita dos artigos, além de realizar o desenvolvimento dos modelos de predição, não foi possível concluir a integração inicial, mas, foi possível desenvolver uma documentação do dashboard do moodle, pois não havia uma documentação viável. Essa documentação consiste em um mapeamento das pastas de funcionalidades de cada script e implementação, listada a seguir:

   Organização das pastas

   - dist : O plugin para o navegador é gerado nesta pasta na escrutura e linguagens que o chrome suporta (html , css e js).
   - src: código fonte do plugin, desenvolvido nas linguagem coffescript, less e html
   - vendor : bibliotecas de terceiros por exemplo jquery
   - .editorcong : arquivo de configuração para seu editor ou ide

- .gitignore : arquivo ignorados pelo git
- Gruntfile.js : tarefas para gerar o plugin
- bower.json: dependências externas para o plugin compilar

Organização do projeto

- coffee : scripts realizado na linguagem coffescript
  - view : scripts de visuzlização, gráficos (O script view.coffee carrega todos os gráficos do sistema)
  - client.coffe : script cliente, modelo cliente-servidor o cliente seria o script carregado junto a aba do navegador.
  - dashboard.coffe : script servidor, no caso o servidor seria o script carregado em background que responde aos clientes com solicitações de dados, esse script armazena os dados de todos os moodles visitados em localstorage do plugin
  - i18n.coffe : script com funções para lidar com multi idomas
  - inject.coffe : identifica se as páginas que o usuário navega para tentar identificar se as mesmas são um moodle
  - moodle.coffee : funções de crawler, buscando verificar identificação do professor e fazer download das base de dados.
- html: template base do projeto
- json: configuração do plugin

Em seguida buscou realizar a junção do software de predição e dashboard moodle analitics, inciando então o desenvolvimento de uma APIrest em python para integrar com javascript. Atualmente esta sendo integrado ambos os sistemas, devido ao aluno não ter o conhecimento da ferramenta a respeito afim de ser concluída no segundo ano da bolsa de iniciação científica.

[Em desenvolvimento]

8. **Elaboração de artigo científico:**

Foram enviados 2 artigos neste primeiro ano de bolsa, sendo eles para, II Simpósio Ibero-Americano de Tecnologias Educacionais – SITE 2018 de título Identificação precoce de estudantes em risco de reprovação em disciplinas semipresenciais de programação de computadores e para Congresso Brasileiro de Informática na Educaçã - CBIE2018 de título Comparação de diferentes configurações de bases de dados para a identificação precoce dos risco de reprovação: o caso de uma disciplina semipresencial de Algoritmos e Programação.

Como resultado ambos os artigos foram aprovados para publicação sendo SITED 2018 podendo ser acessado https://publicacoes.rexlab.ufsc.br/index.php/sited/article/view/285/147, já o segundo artigo foi aprovado na Trilha 4: Inteligência em Ambientes e Sistemas Computacionais para promoção da aprendizagem (Inteligência Artificial, Sistemas Tutores Inteligentes, Web Semântica, Mineração de Dados) e sera publicado, sendo qualis B1 para computação. [Atividade Realizada]

7 Avaliação do Bolsista em Relação ao PIBIC

Gostaria de agradecer pela oportunidade de estar vinculado a um projeto de pesquisa. A experiência de participar do Programa Institucional de Bolsas de Iniciação Científica (PIBIC) demonstrou ser de grande valia para a minha formação acadêmica.

A experiência tem sido extremamente relevante visto que o projeto vai de encontro aos meus interesses e tem relação com o curso de Engenharia de Computação. Demonstrou ser atual e promoveu diversos desafios durante o período em que participei. Tais desafios me fizeram conhecer mais a área de Mineração de Dados , Processamento de dados e Data Science, a saber, linguagens e técnicas de programação, como python, javascript em serviços e coleta de dados e processamento.Vejo grande potencial de evolução e em função disso irei continuar no projeto referente ao período de 2018/2019.

Entre as evoluções do projeto citam-se, o estudo e a interação de bibliotecas desenvolvidas para machine learning com dashboard para uso de professores afim de avaliar e classificar alunos com risco de reprovação. Por fim, gostaria de agradecer a UFSC/CNPq pela oportunidade e ao orientador, Prof. Cristian Cechinel, pelo profissionalismo, conhecimento e condução do projeto.

8 Considerações Finais

O objetivo geral do projeto constituiu na proposição de criação de modelos de predição para acadêmicos em risco em crusos de programação.

Visando cumprir o objetivo, foram realizadas pesquisas bibliográficas pelo bolsista para entender melhor o contexto das áreas de suporte ao projeto, Sistemas de predição utilizando algoritmos de mineração de dados.

Para a realização e desenvolvimento do projeto outras áreas também foram investigadas, entre elas, linguagens e técnicas de programação bibliotecas de machine learning e proces-

samento de dados. A partir desta investigação definiu-se a linguagem de programação, frameworks de suporte e tecnologias. Para o desenvolvimento a linguagem de programa Python foi utilizada. Quanto aos frameworks e bibliotecas foi utilizado pandas, numpy e scikit-learn, foram utilizados o ambiente de programação VIM.

Os resultados mostram que avaliar estudantes em situação de risco de cursos semipresenciais é possível, pois, obteve taxas de acertos altas nos modelos criados. Já nas primeiras semanas os modelos apresentaram um desempenho alto chegando a uma taxa de 85%, e no decorrer da semana tiveram um crescimento ou mantiveram a taxa de acerto constante. Com isso é possível identificar os prováveis reprovados na metade do semestre, assim, o professor pode realizar alguma intervenção antes de saber a situação final do estudante.

As turmas mais recentes analisadas estão mais evoluídas no Moodle no quesito de disponibilizar maior quantidade de atividades como mostra a Tabela 2. Sendo assim, contêm informações para uso por parte dos alunos. E as interações dos mesmos no Moodle tem uma maior relevância para ser utilizadas na geração dos modelos de predições.

Como trabalhos futuros vislumbra-se a evolução da pesquisa no que tange a novos algoritmos de predição, principalmente, as abordagens que permitem mineração de dados e geração de modelos. No que tange ao desenvolvimento do dashboard juntamente com modelos de predição busca a criação de uma API onde pode-se utilizar em dashboard aclopados no moodle para predição de alunos.

Referências

[ABE, 2014] (2014). ABED (20014) Censo EaD.br - Analytic Report of Distance Learning in Brazil. Retrieved.

[Baker et al., 2011] Baker, R., Isotani, S., and Carvalho, A. (2011). Mineração de Dados Educacionais: Oportunidades para o Brasil. *Revista Brasileira de Informática na Educação*, 19(02):3–13.

[Bennedsen and Caspersen, 2007] Bennedsen, J. and Caspersen, M. E. (2007). Failure Rates in Introductory Programming. *SIGCSE Bull.*, 39(2):32–36.

[Brusilovsky and Peylo, 2003] Brusilovsky, P. and Peylo, C. (2003). Adaptive and Intelligent Web-based Educational Systems ADAPTIVE AND INTELLIGENT TECHNOLOGIES FOR WEB-BASED EDUCATIONAL SYSTEMS. *International Journal of Artificial Intelligence in Education*, 13:156–169.

[C. Santana et al., 2014] 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.

[Cechinel et al., 2015] Cechinel, C., Araujo, R. M., and Detoni, D. (2015). Modelagem e Predição de Reprovação de Acadêmicos de Cursos de Educação a Distância a partir da Contagem de Interações. *Revista Brasileira de Informática na Educação*, 23(03):1.

[Costa et al., 2012] Costa, E., Amorim, L., Magalhães, J., and Marinho, T. (2012). Mineração de Dados Educacionais: Conceitos, Téc- nicas, Ferramentas e Aplicações. *Mineração de Dados Educacionais: Conceitos, Téc- nicas, Ferramentas e Aplicações*, d:1–29.

[Detoni et al., 2014] Detoni, D., Araujo, R. M., and Cechinel, C. (2014). Predição de Reprovação de Alunos de Educação a Distância Utilizando Contagem de Interações. (Cbie):896.

[Detoni et al., 2016] 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.

[Dickson, 2005] Dickson, W. (2005). Toward a deeper understanding of student performance in virtual high school courses: Using quantitative analyses and data visualization to inform decision. *A synthesis of new research in K–12 online learning*.

[Dunican, 2002] 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.

[Einhardt et al., 2016] Einhardt, L., Tavares, T. A., and Cechinel, C. (2016). Moodle analytics dashboard: A learning analytics tool to visualize users interactions in moodle. *Proceedings - 2016 11th Latin American Conference on Learning Objects and Technology, LACLO 2016*.

[El Khouli et al., 2009] 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.

[Garrison et al., 1999] Garrison, D. R., Anderson, T., and Archer, W. (1999). Critical Inquiry in a Text-Based Environment: Computer Conferencing in Higher Education. *Internet and Higher Education*, 2(2-3):87–105.

[Gomes, 2010] 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.

[Guércio et al., 2014] Guércio, H., Marques, P., Ströele, V., Pereira, C. K., and Barrere, E. (2014). Análise do Desempenho Estudantil na Educação a Distância Aplicando Técnicas de Mineração de Dados. (Cbie):641.

[Hämäläinen and Vinni, 2010] Hämäläinen, W. and Vinni, M. (2010). Classifiers for educational data mining. *Handbook of Educational Data Mining*, pages 57–74.

[Jenkins, 2002] 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, 2015] 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.

[Macfadyen and Dawson, 2010] Macfadyen, L. P. and Dawson, S. (2010). Mining LMS data to develop an "early warning system"for educators: A proof of concept. *Computers and Education*, 54(2):588–599.

[Mishra et al., 2014] Mishra, T., Kumar, D., and Gupta, S. (2014). Mining students' data for prediction performance. *International Conference on Advanced Computing and Communication Technologies, ACCT*, (February 2014):255–262.

[Mow, 2006] 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.

[Murray et al., 2013] Murray, M., P??rez, J., Geist, D., and Hedrick, A. (2013). Student interaction with content in online and hybrid courses: Leading horses to the proverbial water. *Informing Science*, 16(1):99–115.

[Queiroga et al., 2015] Queiroga, E., Cechinel, C., and Araújo, R. (2015). Um Estudo do Uso de Contagem de Interações Semanais para Predição Precoce de Evasão em Educação a Distância. (Cbie):1074.

[Queiroga et al., 2017] 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.

[Queiroga et al., 2016] Queiroga, E. M., Cechinel, C., Matsumura Araújo, R., and Da Costa Bretanha, G. (2016). Generating models to predict at-risk students in technical e-learning

courses. *Proceedings - 2016 11th Latin American Conference on Learning Objects and Technology, LACLO 2016*.

[Ramos et al., 2015] 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.

[Rodriguez, 2015] Rodriguez, A. M. (2015). Academic Analytics: Aplicando Técnicas De Business Intelligence Sobre Datos De Performance Académica En Enseñanza Superior. *Interfaces Científicas - Exatas e Tecnológicas*, 1(2):35–46.

[Romero and Ventura, 2010] Romero, C. and Ventura, S. (2010). Educational data mining: A review of the state of the art. 40:601 – 618.

[Santos et al., 2012] Santos, H., Camargo, F., and Camargo, S. (2012). Minerando Dados de Ambientes Virtuais de Aprendizagem para Predição de Desempenho de Estudantes. *Conferencias LACLO*, 3(1).

[Santos et al., 2016] Santos, R., Pitangui, C., Vivas, A., and Assis, L. (2016). Análise de Trabalhos Sobre a Aplicação de Técnicas de Mineração de Dados Educacionais na Previsão de Desempenho Acadêmico. *V Congresso Brasileiro de Informática na Educação*, 5(1):960.

[Swan, 2003] Swan, K. (2003). Learning effectiveness: What the research tells us. 4.

[Vihavainen et al., 2014] 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 and Li, 2014] 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.

Apêndices

A Questionário

1. Você possui computador/ notebook em casa?
2. Você possui computador/notebook em casa com acesso a internet?
3. Qual a velocidade de conexçao da internet na sua casa?
4. Você possui smartphone/tablet?
5. Seu smartphone/table tem acesso a internet?
6. Quantas horas POR DIA você usa, em média, o computador/notebook para TRABALHAR?
7. Quantas horas POR DIA você usa, em média, o computador/notebook para ESTUDAR? Quantas horas POR DIA você usa o computador/notebook para LAZER?
8. Quantas horas ou minutos POR DIA você estima que gasta em uma dessas atividades (indique se são horas ou minutos):
   (a) Lendo/verificando emails?
   (b) Redes sociais(Facebook, Twitter...)
   (c) Whatsapp
   (d) Lendo notítias
   (e) Jogando
   (f) Estudando
9. O curso de TIC era sua primeira opção nas provas de acesso?
10. Quais eram as suas opções de curso (enumere-as em ordem)?
11. Em um cenário ideal, onde vocÇe tivesse a liberdade de escolher qualquer curso, sem realizar qualquer exame/teste/prova, quais cursos você escolheria?
12. Quais linguagens de programação você é capaz de programar? Considere os níveis a seguir. (1 básico, 2 intermediário, 3 experiente)

(a) C

(b) C++

(c) C

(d) Pascal

(e) Java

(f) Ruby

(g) Matlab

(h) PHP

(i) Javascript

13. Você já fez essa disciplina (ou outra relacionada, ocmo algum curso de linguagem de programação, disciplina de programação em outra instituição, reprovação/desistência na mesma instituição) em outra oportunidade?

14. Em que instituição/empresa/curso você fez essa disciplina? Teve sucesso(caso tenha falhado, aponte os motivos)?

15. Qual a sua cidade e estado de origem (antes de começar o curso de TIC)?

16. Você possui trabalho remunerado?

17. Quantas horas por dia? Qual o horário de trabalho? Você possui carteira assinada? Fale um pouco da sua função/atividade.

18. Na sua opinião, qual importância da disciplina de programação para o curso de TIC?

19. Você conhece o sistema de monitoria da UFSC?