Uma Investigação Sobre a Identificação de Indicadores de Evasão de Alunos Utilizando Mineração de Dados

Francisco da Conceição Silva[1], **Josenildo Costa da Silva**[1], **Arthur Mota França**[1], **George Sanders Carvalho Araújo**[1], **Emerson Elias Moraes**[1]

Departamento de Computação – Instituto Federal de Educação, Ciência e Tecnologia do Maranhão (IFMA)
São Luis – MA – Brazil

{francisco.conceicao,jcsilva}@ifma.edu.br,

{arthur.franca,george.carvalho,e.emerson}@acad.ifma.edu.br

Abstract. *In this work, we investigate indicators to help interested parties (teachers, managers, etc.) identifing students who may drop out later from courses of a technical and technological education institution. As part of the research, we developed a tool to identify relationships in student data on learning performance. These data were submitted to data mining techniques to generate predictive models. Decision tree-based models, specifically XGboost, had the best performances to classify a student as possible dropout with accuracy and sensitivity of 96% and 83% respectively.*

Resumo. *Neste trabalho, propõe-se uma investigação acerca da identificação de indicadores de evasão de alunos em cursos de uma instituição de ensino técnico e tecnológico por meio de técnicas de mineração de dados, para auxiliar as partes interessadas (professores, gestores, etc) na tomada de decisão. Como parte da pesquisa, foi desenvolvida uma ferramenta para identificar relações em dados de alunos sobre o desempenho de aprendizagem. Estes dados foram submetidos a técnicas de mineração de dados para geração de modelos preditivos. Os modelos baseados em árvore de decisão, especificamente, o XGboost, obtiveram os melhores desempenhos para classificar um aluno como possível evadido. Os modelos alcançaram precisão e sensibilidade de 96% e 83% respectivamente.*

1. Introdução

A identificação de indicadores de desempenho em cursos é importante para que seja possível proporcionar as condições necessárias que reduzam ou eliminem as dificuldades de aprendizagem. Para isso, são necessários métodos e ferramentas de análise de dados a fim de observar o comportamento dos alunos para auxiliar as partes interessadas na tomada de decisão [Silva et al. 2015].

Uma forma de fazer a identificação de relações relevantes em grandes bases de dados é por meio da Mineração de Dados (MD), que busca explorar e analisar esses dados para identificar regras, padrões ou desvios. A MD é um processo para extração de conhecimento que estão implícitos, são previamente desconhecidos e são muito úteis para um contexto de estudo.

A aplicação de MD abrange um número grande de áreas, como Inteligência Artificial, Estatística, Banco de Dados [Fayyad et al. 1996, Witten et al. 2011] e, mais recentemente, a área educacional (Mineração de Dados Educacionais - MDE) [Romero et al. 2013]. A MD utiliza técnicas eficientes do Knowledge Discovery in Databases (KDD) e é uma área que pode contribuir na descoberta automática de conhecimento que é potencialmente útil, por meio de algoritmos de aprendizado de máquina. As ferramentas de MD processam esses dados de forma a buscar correlações importantes entre eles e tem sido objeto de estudos interdisciplinares, tendo se desenvolvido muito nos últimos anos. No contexto educacional, a utilização de técnicas de MDE pode viabilizar melhores condições para que o professor tenha êxito na mediação pedagógica a seus alunos e pode ser aplicada a conjuntos de dados para encontrar correlações entre os dados disponíveis [Romero et al. 2013]. Alguns dos desafios nesse sentido é localizar características relevantes para o êxito do aluno no curso. Acredita-se que a identificação de padrões de desempenho de alunos possa ajudar educadores e desenvolvedores de materiais a avaliar e interpretar as atividades de um curso, as formas como são executadas e seus resultados.

Neste trabalho, propõe-se uma investigação acerca dos indicadores de desempenho de alunos em cursos de uma instituição de ensino técnico e tecnológico, por meio de técnicas de MD, para auxiliar as partes interessadas (professores, gestores, etc) na tomada de decisão. Como parte da pesquisa, foi desenvolvida uma ferramenta para identificar relações em dados de alunos sobre o desempenho de aprendizagem referentes em um curso. Os modelos baseados em árvore de decisão obtiveram os melhores desempenhos para classificar um aluno como possível evasor.

2. Trabalhos Relacionados

Diversos trabalhos na literatura dedicam-se ao estudo das dificuldades encontradas no ensino, aplicando MD para descoberta de conhecimento em bases de dados de alunos, tais como [Santana et al. 2015], [Pascoal et al. 2015], [Silva et al. 2015] e [Silva 2017].

[Santana et al. 2015] realizaram um estudo com técnicas de mineração de dados educacionais, com o objetivo de comparar a eficácia dos algoritmos de predição capazes de identificar os alunos propensos ao insucesso. Neste estudo, avaliou-se a eficácia de algoritmos de predição em duas fontes de dados diferentes e independentes, uma na modalidade presencial e outra na modalidade de ensino a distância sobre as disciplinas de programação introdutória. Os resultados mostraram que as técnicas analisadas no estudo são eficazes na identificação dos alunos propensos ao insucesso no início da disciplina. Ao fim do processo, o algoritmo máquina de vetor de suporte (Support Vector Machine - SVM) apresentou os melhores resultados, tanto na modalidade de ensino presencial quanto na modalidade a distância, alcançando uma taxa de f-measure de 83% e 92%, respectivamente.

[Pascoal et al. 2015] utilizaram técnicas de mineração de dados para a previsão de desempenhos de alunos de computação em disciplinas de programação, utilizando como dados de entrada os resultados das provas de ingresso na instituição de ensino e, para alguns casos, o desempenho dos mesmos em disciplinas pré-requisitos para as disciplinas avaliadas. Os autores selecionaram quatro algoritmos de aprendizagem de máquina distintos: IBk, RandomForest (RF), BayesNET (BNet) e MultilayerPerceptron (MLP),

todos disponíveis na ferramenta WEKA, e, para cada um deles, buscou-se, por tentativa e teste, encontrar o conjunto de parâmetros que produzisse os melhores resultados. Para a avaliação dos métodos foram utilizadas a acurácia e as taxas de verdadeiros positivos (VP) e verdadeiros negativos (VN). Os autores relatam que foi possível identificar alunos usando informações de desempenho na prova de ingresso e nas disciplinas pré-requisito com taxas de acerto superiores a 80%. Apontam, ainda, que os resultados obtidos podem ser utilizados no planejamento de estratégias para reduzir o problema da reprovação nas disciplinas de programação e, consequentemente, reduzir os índices de evasão.

[Silva et al. 2015] apresentam o desenvolvimento de um modelo preditivo de MD em um Ambiente Virtual de Aprendizagem (AVA), a partir das interações de alunos em fóruns de discussão de um curso técnico em informática. O objetivo era realizar o diagnóstico de baixo desempenho de alunos, que é considerado um forte indício para evasão, gerando relatórios que auxilie as partes interessadas na tomada de decisão. A MD foi aplicada através de cinco algoritmos de classificação: J48, BFTree, SimpleCart, Bayesianos e BayesNet, sendo comparado o desempenho de cada um, a fim de que um modelo com melhor desempenho fosse obtido. Os autores não focaram especificamente nas dificuldades de aprendizagem em programação, embora tenham utilizado dados de alunos em curso de computação. O modelo proposto pelos autores classificou os alunos em três perfis: Aprovado, Reprovado e Evadido. O algoritmo J48 alcançou melhor desempenho (73,96%).

[Silva 2017] desenvolveu uma ferramenta prática e funcional para a mineração e visualização de dados do Twitter. A extração dos dados (Tweets) foi feita por meio da API disponibilizada pelo Twitter, que permite recuperar as últimas publicações. Os dados foram submetidos a uma coleção de algoritmos de mineração de dados visando encontrar associações relevantes nos mesmos. Os resultados obtidos foram exibidos aos usuários por meios de bibliotecas gráficas como Google Charts e D3.js com o intuito de facilitar a visualização e análise. Para o autor, a ferramenta tem grande potencial para estratégias de marketing, uma vez que, a partir dos gráficos de incidência e análise de sentimentos, é possível desenvolver estratégias personalizadas para locais geográficos que de fato possam propiciar o aumento das vendas de determinado produto ou serviço. O autor afirma que a aplicação foi testada sobre temas da atualidade e se mostrou funcional, visto que foi capaz de realizar as tarefas requisitadas da maneira esperada.

Pode-se citar alguns dos grandes diferenciais deste trabalho em relação aos trabalhos relacionados, que são: uma ferramenta de fácil manuseio, coleta de dados e visualização de diagnóstico de evasão de aluno, bem como, disponibilização da ferramenta como um serviço da web para as partes interessadas.

3. Fundamentação teórica

De um modo geral, ao ingressar em um curso, boa parte dos alunos opta por aqueles que tem maior afinidade com seus interesses e habilidades. Os alunos buscam uma formação que lhes assegure competência profissional para garantir melhores condições de atuação no mundo do trabalho [Soares and Carvalho 2017]. Em relação aos cursos de computação, por exemplo, as disciplinas de programação são fundamentais para que o aluno tenha uma boa formação profissional, permitindo-lhe a abertura para um leque de áreas, tais como a Engenharia de Software, Banco de Dados e Teoria da Computação.

No entanto, logo ao iniciarem algum curso de computação, os alunos se deparam com disciplinas voltadas para a programação de computadores, em especial, a disciplina de algoritmos, e, então, muitas dificuldades começam a surgir. Dada a necessidade de habilidades lógicas para cursá-las e do esforço e dedicação para desenvolvê-las, muitos acabam abandonando o curso já nos períodos iniciais, ou optando por áreas que exijam um esforço menor de aprendizagem [Soares and Carvalho 2017]. [Vieira et al. 2015] destaca que, dada a devida importância desta disciplina, ela constitui-se como grande "divisor de águas" nos cursos de Computação, uma vez que causa grande impacto no primeiro ano de formação do aluno. Evidencia, ainda, que a não compreensão do assunto constitui uma grande barreira que impede a progressão do discente para outros períodos.

A identificação de indicadores de desempenho em cursos é importante para que seja possível proporcionar as condições necessárias que reduzam ou eliminem as possibilidades de evasão de aluno. Para isso, são necessários métodos e ferramentas de análise de dados a fim de observar o comportamento dos alunos para auxiliar as partes interessadas na tomada de decisão [Silva et al. 2015].

Uma forma de fazer a identificação de relações relevantes em grandes bases de dados é por meio da MD, que busca explorar e analisar esses dados para identificar regras, padrões ou desvios. A MD é um processo para extração de conhecimentos que estão implícitos em uma base de dados, são previamente desconhecidos e muito úteis para um contexto de estudo.

A aplicação de MD abrange um número grande de áreas, como Inteligência Artificial, Estatística, Banco de Dados ([Fayyad et al. 1996]; [Witten et al. 2011]) e, mais recentemente, a área educacional (MDE) [Romero et al. 2008]. A MD utiliza técnicas eficientes do KDD e é uma área que pode contribuir na descoberta automática de conhecimento que é potencialmente útil, por meio de algoritmos de aprendizado de máquina. As ferramentas de MD processam esses dados de forma a buscar correlações importantes entre eles e tem sido objeto de estudos interdisciplinares, tendo se desenvolvido muito nos últimos anos.

No contexto educacional, a utilização de técnicas de MDE pode viabilizar melhores condições para que o professor tenha êxito na mediação pedagógica a seus alunos e pode ser aplicada a conjuntos de dados para encontrar correlações entre os dados disponíveis, que visam [Romero et al. 2013]:

- localizar quais características e/ou comportamentos que contribuem para o êxito (ou não) na realização de um curso;
- extrair padrões úteis para ajudar educadores e desenvolvedores de materiais a avaliar e interpretar as atividades de um curso, as formas como são executadas e seus resultados;
- em um ambiente virtual, guiar automaticamente as atividades dos alunos, gerando e recomendando materiais;
- etc;

Através da MDE é possível analisar dados educacionais e buscar por padrões e relações desses dados com o objetivo de detectar características particulares, gerando conhecimento, e a partir daí realizar previsões de desempenho de um aluno.

4. Metodologia

4.1. Descrição dos dados

Os dados utilizados neste projeto foram obtidos junto ao sistema acadêmico de uma instituição de ensino técnico e tecnológico, separados em dois arquivos no formato CSV (Comma Separated Values): `matricula.csv` e `historico.csv`. Todos os dados fornecidos são anonimizados, isto é, não há atributos que possam identificar os estudantes. Cada arquivo versa sobre um aspecto da vida acadêmica do aluno. O arquivo de histórico apresenta informações acerca da performance acadêmica do aluno, bem como informações das disciplinas cursadas pelo mesmo. Já a tabela matrículas contém todas as informações socioeconômicas informadas pelo estudante no momento da matrícula no Instituto e atualizadas ao início de cada semestre letivo. O arquivo de matrícula consiste de 34.521 registros e o arquivo histórico consiste de 744.587 registros. Os dados representam todos alunos de todos os cursos, em várias modalidades diferentes, tanto técnico quanto superior, desde 2011.

No arquivo `matrícula` temos os atributos: alunoid, campus, curso, anoingresso, periodoingresso, dataconclusao, forma_acesso_seletivo, rendabruta, índice de rendimento acadêmico (IRA), modalidade, genero, raca, idade, idioma, ficou_tempo_sem_estudar, razao_ausencia_educacional, quantidade_computadores, exclusivo_rede_publica, companhia_domiciliar, mae_nivel_escolaridade, pai_nivel_escolaridade, quantidade_notebooks, estado_civil, qtd_filhos, tipo_area_residencial, trabalha, e situacao. Já no arquivo `histórico`, aparecem os atributos: alunoid, diarioid, componentecurricular, ano_letivo, periodo_letivo, carga_horaria, percentual_carga_horaria_frequentada, media_final_disciplina e situacao. O atributo alunoid, presente em ambos os arquivos, serve de chave para a manipulação das informações dos arquivos, mas foi substituido por um novo valor único para não identificar a matricula real do aluno. Existe também uma peculiaridade que é a presença de um atributo chamado situação em ambos os arquivos. Em matricula, esse atributo nos traz a informação sobre a matricula do estudante, sendo a classe objetivo do modelo a ser desenvolvido. Já em histórico, a informação trazida por esse atributo versa sobre o desempenho do aluno ao final da disciplina cursada.

4.2. Pré Processamento

Os dados foram carregados para o Google Colaboratory (Colab) por meio do GoogleAuth, com acesso à arquivos armazenados em contas do Google Drive sem fazer upload dos mesmos para o Colab. Utilizamos a biblioteca `pandas` (em Python) para carregar os dados para memória na máquina virtual do Colab. Após o carregamento dos dados, os nomes das colunas de ambos os arquivos foram padronizados para minúsculas.

O primeiro passo após o carregamento dos dados foi tratamento de valores extremos (*outliers*) e nulos. Os valores de renda, idade e IRA negativos foram igualados a zero e os extremos superiores foram limitados a um valor mais realista, após discussão realizada com o setor responsável pelos dados. Por exemplo, renda familiar foi modificado para $R\$\ 10.000,00$ e IRA máximo modificado para 10. Foram substituídos valores nulos por zero nos atributos quantidade de notebooks e quantidade de computadores. No arquivo histórico, foram criadas três novas colunas para representar a média do aluno: media_final_mean, media_final_median e media_final_frequency. A primeira substitui valores nulos pela média da disciplina. A segunda utiliza a mediana com o mesmo propósito.

A terceira utiliza valor mais frequente. Em seguida foram criados dois novos atributos: número de reprovações por nota e número de reprovações por falta. Foram excluídos os atributos dataconclusao, forma_acesso_seletivo, razao_ausencia_educacional e exclusivo_rede_publica, já que eles apresentam mais de 95% de valores nulos. Por fim, algumas colunas foram transformadas do tipo `object` para o tipo numérico. Esse passo foi necessário pois o `pandas` não conseguiu converter corretamente algumas colunas.

A coluna alvo para predição chama-se `situação` e descreve o status da matrícula dos alunos, possuindo valores tais como: matriculado, formado, concluído, evadido, trancado, entre outros. Esses valores foram agrupados em duas classes: 0 para alunos com baixa probabilidade de evasão e 1 para alunos com alguma probabilidade de evasão ou evadidos. Foram retirados do conjunto de dados os alunos ainda matriculados, já que a sua situação ainda não está definida. Após o filtro, restaram 11.331 registros, dos quais 7.879 são casos de aluno que concluíram o curso (formado e concludentes) e 3.452 são alunos que evadiram (evadidos, trancados, trancados ex-oficio).

Os dados numéricos foram normalizados para média zero e desvio padrão 1 utilizado o método `StandarScaler()` da biblioteca `sklearn`, em Python. Já os atributos categóricos foram transformados com o `OrdinalEncoder`, também da bibliteca `sklearn`.

4.3. Treinamento

Para a fase de treinamento os dados originais foram divididos em dados de treino e de testes, onde 70% dos registros foram destinados para treino e 30% para testes.

Utilizou-se os seguintes algoritmos: Naive Bayes, Árvores de decisão, SVM, Random Forest, e XGBoost. O Naive Bayes por ser o modelo mais simples, representa uma base de comparação para os demais modelos. Árvores de Decisão, Random Forest e XGBoost foram escolhidos por sua flexibilidade, embora sejam propensos a sobreajuste. SVM foi incluído como opção de um modelo mais estável em relação aos modelos baseados em árvore. Todos os algoritmos foram utilizados com seus valores padrão, não sendo realizado qualquer busca de melhores hiperparâmetros. A seleção final do modelo seguiu a análise de resultados e escolha dos atributos mais adequados definidos a partir da utilização de três algoritmos diferentes: o Mutual Information, F-Classification e o Chi-Squared [Kuhn and Johnson 2013].

4.4. Criação do serviço

Após as etapas de Mineração de dados, Treinamento e Avaliação, foi criado um serviço Web utilizando Python e biblioteca FastAPI[1]. Todos os objetos utilizados nas etapas de pré-processamento foram serializados através da biblioteca Pickle: normalizadores, categorizadores e também o modelo final. Os arquivos serializados foram carregados no aplicativo servidor. Por meio do Framework Web FastAPI, foi criado um aplicativo que possibilita a execução de requisições HTTP. O serviço responde a chamadas POST com um endpoint denominado `predict`. Este endpoint recebe os dados no corpo da requisição POST e retornam um rótulo de classificação: 0 ou 1, indicando a possibilidade de evasão.

[1] `https://fastapi.tiangolo.com/`

4.5. Implantação (Deployment)

O deployment foi feito no Heroku, que é uma plataforma em nuvem para implantação de aplicações [Kemp and Gyger 2013] [Middleton and Schneeman 2013]. Além dele, também foi utilizado o Uvicorn[2], que permite iniciar um servidor e disponibilizar as rotas para uso de outros aplicativos. Por último, é importante mencionar que o projeto está hospedado no GitHub, e qualquer alteração enviada para o repositório, causará um deploy automático para o aplicativo no Heroku.

4.6. Interface Web

Para consumir os dados disponibilizados no serviço mencionado anteriormente, foi desenvolvido um cliente web em Django[3]. O Django é um framework *full stack* baseado em Python que possibilita o desenvolvimento de aplicações Web. Por meio dele, desenvolveu-se um formulário com todos os dados dos estudantes necessários para a classificação. Ao enviar as informações, é lançada uma requisição POST para o serviço, onde é realizado o processamento, a classificação e o resultado é retornado. Este retorno pode ser visualizado na página mostrando a mensagem "Provável evasão" ou "Evasão improvável". O cliente Web também está hospedado no Heroku e pode ser acessado no endereço `https://evasao-ifma-client.herokuapp.com`.

5. Resultados e Discussão

Reduzir o número de atributos em um modelo preditivo é muitas vezes uma etapa muito importante a fim de se reduzir o consumo de recursos computacionais, principalmente tempo de processamento. No modelo que obteve melhor rendimento em termos de eficiência, o treinamento foi feito com 10 atributos. São eles: IRA, anoingresso, modalidade, curso, campus, idade, pai_nivel_escolaridade, trabalha, companhia_domiciliar, e qtd_filhos.

É importante notar que o atributo IRA (índice de rendimento acadêmico), que é a média das notas das disciplinas do aluno, apresentou um poder de predição. Utilizando apenas o IRA foi possível atingir bons resultados na predição de evasão. Entretanto, ao discutir com a equipe pedagógica e do departamento de dados, percebeu-se que o IRA é um atributo calculado após os resultados da média a cada semestre. Portanto, pode tratar-se de um caso de vazamento de informação, onde o IRA deve ser tratado como variável dependente e não como preditor. Por isso, o IRA foi removido da lista de atributos e novos modelos foram treinados. Com a retirada do atributo, percebeu-se que os modelos tiveram uma queda na performance, sobretudo na especificidade, que é a capacidade desse modelo prever a classe negativa.

5.1. Avaliação

A avaliação dos diversos modelos mostrou como os diferentes algoritmos performaram com diferentes técnicas de seleção de atributos (cf. Tabelas 1, 2, 3, e 4). Os resultados, nas tabelas a seguir, representam a performance de cada modelo nos dados de testes.

O Random Forest teve a melhor acurácia seguido de perto por XGBoost e Árvores de Decisão (ver Tabela 1). O Naive Bayes teve o pior desempenho de acurácia. O SVM teve um desempenho intermediário em termos de acurácia.

[2] `https://www.uvicorn.org/`

[3] `https://www.djangoproject.com/`

Tabela 1. Acurácia: porcentage de instâncias classificadas corretamente.

	Naive Bayes	Árvores	Random Forest	SVM	XGBoost
Mutual_info	0.77	0.93	**0.94**	0.81	0.93
F_class	0.76	0.92	**0.93**	0.81	0.92
Chi_squared	0.75	0.89	**0.91**	0.80	**0.91**

O XGBoost atinge melhor precisão, seguido do Random Forest (ver Tabela 2). Árvores de Decisão e SVM tem precisão similar, com vantagem para as árvores. Já o Naive Bayes tem a pior precisão.

Tabela 2. Precisão. Dentre todas os preditos positivos, quantas são de fato positivas.

	Naive Bayes	Árvores	Random Forest	SVM	XGBoost
Mutual_info	0.63	0.90	0.95	0.79	**0.96**
F_class	0.63	0.89	**0.94**	0.79	**0.94**
Chi_squared	0.63	0.84	0.90	0.76	**0.93**

Árvores de Decisão e Random Forest apresentam a melhor sensibilidade, seguidos por XGBoost (ver Tabela 3). SVM tem o pior desempenho, provavelmente porque não foi realizado otimização de hiperparâmetros.

Tabela 3. Recall (sensibilidade): capacidade de prever a classe positiva

	Naive Bayes	Árvores	Random Forest	SVM	XGBoost
Mutual_info	0.67	**0.88**	0.86	0.58	0.83
F_class	0.61	**0.86**	0.83	0.58	0.82
Chi_squared	0.59	0.82	**0.83**	0.55	0.79

O XGBoost gerou modelo com melhor especificidade seguido de perto por Random Forest (ver Tabela 4). Naive Bayes foi o pior nessa métrica. Árvore e SVM tem valores similares.

Tabela 4. Especificidade: capacidade do modelo de prever a classe negativa

	Naive Bayes	Árvores	Random Forest	SVM	XGBoost
Mutual_info	0.81	0.95	0.98	0.93	**0.99**
F_class	0.83	0.95	**0.98**	0.93	**0.98**
Chi_squared	0.84	0.93	0.96	0.92	**0.97**

Como estamos interessado na classe evasão, que neste contexto é nossa classe positiva, as métricas de precisão e sensibilidade são as mais importantes. A melhor precisão, considerando todos os modelos e todos os métodos de seleção de atributos foi de 0.96. Já a melhor sensibilidade foi de apenas 0.88. Isso mostra que apesar de que 0.88 dos evasores são verdadeiros positivos, há cerca de 0.12 falso positivos. A questão prática é saber

se tal modelo consegue impacto no acompanhamento pedagógico com estes valores, ou se é necessário atingir-se melhores valores de sensibilidade.

Neste contexto, o algoritmo **XGboost** nos parece ser a melhor opção pois atingiu a **melhor precisão**, de 0.96 e uma sensibilidade de 0.83, que está ligeiramente acima da média dos grupo de algoritmos analisados.

5.2. Possibilidades de uso do serviço para predição em batch

O serviço Web criado através da FastAPI prevê, ainda, uma funcionalidade a ser implementada na fase seguinte do projeto, que permite a criação de um ponto de chamada (*endpoint*) que realiza predições em lote (*batch*). Neste caso o serviço recebe dados com os mesmos atributos realizados na predição individual, porém de vários alunos. Após o processamento, retorna uma coleção de resultados, que deve ser exibida por uma aplicação cliente.

5.3. Outros aplicativos consumidores da API do serviço

Outras aplicações podem utilizar os métodos do serviço criado, utilizando outras tecnologias. Está prevista a criação de um cliente Web utilizando o Streamlit. Trata-se de um framework também feito em Python, que agiliza a criação de Front-End através de poucas linhas de código.

Um aspecto importante, mas que não investigamos neste projeto, a possibilidade de atualizar o modelo aprendido. O momento de atualização de um modelo em produção é um desafio sobre o qual ainda não há melhores práticas definidas na comunidade. Portanto, uma direção futura para este projeto é investigar como detectar que o modelo em produção aprensenta queda de performance e precisa de atualização. Neste contexto, um dos conceitos importantes é o chamado *data drift*, que representa uma mudança significativa na distribuição de probabilidade dos dados.

6. Conclusão

Neste trabalho foi investigado o fenômeno da evasão escolar em uma instituição de ensino técnico e tecnológico. Foram utilizados dados de alunos de diversos cursos e campus, em que foram aplicadas técnicas de MD para gerar modelos preditivos. A partir dos modelos gerados foi possível encontrar indicadores de desempenho que podem estar relacionados com a evasão.

Dentre os diversos indicadores identificados, destacam-se: IRA, modalidade, curso e campus. Este estudo ainda abordou aspectos práticos relativos à implantação dos modelos aprendidos. O melhor modelo foi disponibilizado através de um serviço web em forma de API *online* que pode ser consumida por diversos tipos de clientes.

Como trabalhos futuros, pretende-se realizar as seguintes melhorias:

- refinar os modelos com otimização de hiperparâmetros para que faça uma melhor classificação;
- implementar um cliente para trabalhar com lote (*batch*) de alunos;
- implantar o modelo na instituição e tratar atualizações de forma automática.

Referências

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

Kemp, C. and Gyger, B. (2013). *Professional Heroku Programming*. ITPro collection. Wiley.

Kuhn, M. and Johnson, K. (2013). *Applied Predictive Modeling*. Springer.

Middleton, N. and Schneeman, R. (2013). *Heroku: Up and Running: Effortless Application Deployment and Scaling*. O'Reilly Media.

Pascoal, T. A., Brito, D., and Rêgo, T. (2015). Uma abordagem para a previsão de desempenho de alunos de computação em disciplinas de programação. *Nuevas Ideas en Informática Educativa TISE*, 2015(454-458):2.

Romero, C., Espejo, P. G., Zafra, A., Romero, J. R., and Ventura, S. (2013). Web usage mining for predicting final marks of students that use moodle courses. *Computer Applications in Engineering Education*, 21(1):135–146.

Romero, C., Ventura, S., and García, E. (2008). Data mining in course management systems: Moodle case study and tutorial. *Computers & Education*, 51(1):368–384.

Santana, M. A. et al. (2015). *Um estudo comparativo das técnicas de predição na identificação de insucesso acadêmico dos estudantes durante cursos de programação introdutória*. Universidade Federal de Alagoas, Maceió,AL.

Silva, F., da Silva, J., Silva, R., and Fonseca, L. C. (2015). Um modelo preditivo para diagnóstico de evasão baseado nas interações de alunos em fóruns de discussão. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, page 1187.

Silva, M. M. (2017). *Mineração de dados no Twitter: uma ferramenta prática para extração e análise dos resultados.* Universidade Federal de Outro Preto.

Soares, F. A. L. and Carvalho, R. B. (2017). Proposta de um portal educacional para estudantes de programação de computadores. *Abakós*, 5(2):36–58.

Vieira, C. E. C., de Lima Junior, J. A. T., de Paula Vieira, P., et al. (2015). Dificuldades no processo de aprendizagem de algoritmos: uma análise dos resultados na disciplina de algoritmos do curso de sistemas de informação da faeterj–campus paracambi. *Cadernos UniFOA*, 10(27):5–15.

Witten, I. H., Frank, E., and Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 3 edition.