Aprendizado de Máquina utilizado para Previsão da Evasão no Ensino Superior

Vinícius Pacheco Vieira[1], **Gustavo Cantarelli**[1], **Mirkos Ortiz Martins**[1]]

[1]Curso de Ciência da Computação - Centro Universitário Franciscano

{vinipachecov}@gmail.com, gustavo@unifra.br, mirkos@unifra.br

Abstract. *This meta-paper describes the style to be used in articles and short papers for SBC conferences. For papers in English, you should add just an abstract while for the papers in Portuguese, we also ask for an abstract in Portuguese ("resumo"). In both cases, abstracts should not have more than 10 lines and must be in the first page of the paper.*

Resumo. *Este trabalho busca evidenciar a capacidade e a possibilidade de se utilizar técnicas de aprendizado de máquina para o auxílio do problema da evasão discente nas Instituições de Ensino Superior.(continuar)...*

1. Introdução

A Evasão escolar é um problema enfrentado em todas as Instituições de Ensino Superior (IES) seja na esfera pública federal ou privada. Enquanto os esforços são geralmente divididos no encontro de padrões e variáveis fundamentais a respeito da evasão discente ou no desenvolvimento de "cursos"e outras estratégias de resultado duvidoso, é preciso considerar e encarar o sucesso acadêmico de forma mais individualizada, mudando o foco de atenção além do viés das responsabilidades da instituição com o aluno e do aluno com a instituição, sendo o sucesso a grande chave para a retenção dos alunos.

Diante desse cenário, surgiram estudos que embasam a eficácia da retenção nas IES em função do sucesso acadêmico, bem como uma nova área de pesquisa nesse segmento para criar soluções de software baseados em técnicas de aprendizado de máquina. O intuito destes estudos é de auxiliar na identificação e predição de grupos de riscos e disponibilizar melhores dados para que outros profissionais da educação estruturem seus planos de ação.

Esse trabalho se insere no problema e tenta resolver através de algoritmos de aprendizado de máquina sem auxílio de implementações prontas como foi feito no trabalho de [Amorim et al. 2008] ao utilizar o Weka, software neo-zelandês que possui um conjunto de algoritmos de aprendizado de máquina implementados em Java. Dessa forma o presente trabalho se diferencia já pelo aspecto mais básico que é o desenvolvimento do código e a implementação dos algoritmos em uma linguagem e não somente o processo de avaliação da performance das implementações.

2. Objetivos

Dentro do tema, o objetivo do presente trabalho é além de averiguar a validade de se utilizar redes neurais de arquitetura Multi-Layer-Perceptron e outros algoritmos de machine

learning da Scikit-Learn(como complemento) para o problema de predição de evasão discente, é também avaliar a possibilidade do desenvolvimento de um sistema para auxiliar Instituições de Ensino Superior na retenção de alunos.

2.1. Objetivos Específicos

- Fazer estudo sobre Redes Neurais principalmente as de modelo Multi-Layer Perceptron, para identificar a melhor abordagem para a classificação dos dados.
- Construir um modelo de Rede Neural para classificação da evasão de alunos no Ensino Superior.
- Fazer uso da Linguagem Python juntamente da API do IBM-Watson para construção do modelo de classificação.
- Como alternativa ao IBM-Watson será usada linguagem Python com o framework TensorFlow da Google como base para construção do modelo de classificação.
- Fazer uso de dados objetivos dos alunos do Centro Universitário Franciscano como amostra para auxiliar no treinamento do modelo construído.
- Testar os dados com um modelo similar na biblioteca scikit-learn, com outros métodos de Machine Learning para verificar a significância dos resultados.
- Verificar a possibilidade de uso da solução como um software de auxílio para Instituições de Ensino Superior na identificação de alunos propensos a evadirem.

São elementos esperados a se atingir ao final deste trabalho:

- Um sistema capaz de ser treinado com dados dos Centro Universitário Franciscano e fazer estimativas se ele está ou não em um grupo de risco.
- A partir de suas análises possibilitar pesquisa no campo acadêmico para estudar a evasão no Centro Universitário Franciscano incorporando outras áreas do conhecimento.

3. Trabalhos Correlatos

Com base nessa temática foram desenvolvidos trabalhos como [Dekker et al. 2009], [Zhang et al. 2010],[Delen 2010] e [MARTINHO et al. 2013] com enfoques diferentes apesar de terem como cerne a questão da predição e compreensão da evasão discente. As soluções se utilizam de ténicas de aprendizado de máquina como redes neurais, random forests, regressão logística, maquinas de vetores de suporte e outros algoritmos. A relevância da área de da pelos altos valores de acurácia ou de outros qualificadores estatísticos.

4. Organização e Estratégia

O trabalho seguirá um workflow organizado para atingir as metas e objetivos do trabalho de forma eficiente e robusta. Para isso foi organizado um fluxograma com os passos a serem elaborados e seguidos:

4.1. Coleta de Dados

Como o objetivo é a classificação de dados é necessário sempre haver uma base de dados. Dependendo da complexidade da aplicação ou o do que se pretende trabalhar não é obrigatório o uso de SGBD's ou de frameworks como o Hadoop para Big Data. Neste trabalho serão utilizados arquivos de tabelas como .xls ou de dados separados por vírgula como .csv.

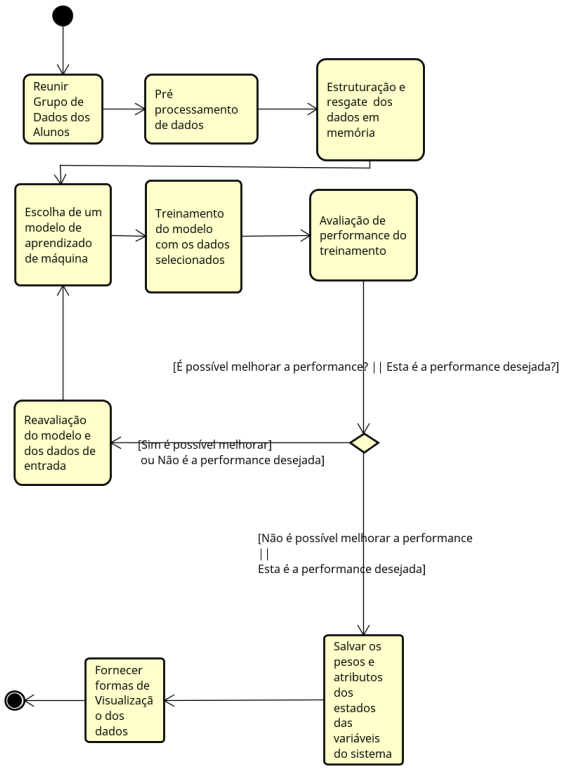

Figura 1. Ordem do Processo de Machine Learning

4.2. Pré Processamento

A segunda etapa necessária para fazermos uma classificação de dados é organizarmos os dados recebidos e pre-processarmos eles para poderem ser encaixados em algoritmos de aprendizagem de máquina.

4.2.1. Dados incompletos

Na existência de entradas de dados com variáveis incompletadas, é possível duas escolhas, a retirada desse exemplar ou atribuir alguma valor a variável.No caso do dataset ter um tamanho pequeno, a retirada pode comprometer o equilíbrio entre as classes a serem classificadas por conter alguma informação importante. Nesse caso poderia se fazer a média dos valores da coluna a qual o dado faltante se encontra e preencher com o resultado do cálculo. Em casos de datasets é possível se desfazer de dados faltantes sem comprometer tanto o balanço entre classes, o que não impede de também adotar a primeira opção de calcular a média ou outros cálculos para preencher com o resultado.

4.2.2. Lidando com Dados multivalorados

Muitos dados em datasets podem se encaixar como multivalorados, ou seja que uma coluna de uma variável ou feature pode receber mais de um valor. Esse caso pode ser melhor exemplificado quando temos por exemplo alunos de vários países em uma Universidade. Nesse caso não é possível inserir os nomes dos países dentro das equações

dos modelos matemáticos de machine learning e para isso é preciso codificá-los para que seja possível. Nesse processo de codificação que podemos chamar de Label Encoding ou One-Hot-Encoding, iremos transformar as representações textuais em um grupo possível de valores. No caso dos alunos de vários países e um conjunto hipotético [Brasil, Chile, Argentina], uma possível solução seria um conjunto de 3 valores:

- 0 para representar alunos brasileiros
- 1 para representar alunos chilenos
- 2 para representar alunos argentinos

Dessa forma, para todos atributos multivalorados teremos um conjunto de valores de 0 até K-1 valores onde K é o número de valores possíveis do determinado atributo.

4.3. Padronização ou Normalização de dados

Após a coleta de dados verificamos que muitos dados possuem uma variância ou dispersão muito grande, o que irá comprometer o treinamento e os resultados dos algoritmos de machine learning.Isso se deve ao fato de que se não fizermos esse processo, também chamado de feature scaling, algumas variáveis com valores com mais alta dispersão estatística terão peso maior de importância apenas por conta do seu valor bruto e não pela sua significância estatística pelo modelo. Esse resultado errado ocorre por conta de que a maioria dos algoritmos de classificação utilizam a fórumula da distância euclidiana. Com esse problema em mente, podemos resolver de duas formas: normalização ou padronização. Os dois métodos conforme na Figura 2 tem como propósito colocar os dados de uma determinada variável (coluna da tabela dos nossos dados) em uma escala comum e principalmente reduzida. Dessa forma todas as variáveis irão contribuir de maneira proporcional dentro do modelo escolhido de machine learning.

Standardisation	Normalisation
$x_{\text{stand}} = \frac{x - \text{mean}(x)}{\text{standard deviation }(x)}$	$x_{\text{norm}} = \frac{x - \min(x)}{\max(x) - \min(x)}$

Figura 2. Formas de colocar dados na mesma escala

O resultado da padronização ou normalização z-escore é quando rescalamos nossas features para que elas tenham como propriedades de uma distribuição normal com média igual a zero e desvio padrão igual a um. A padronização de features para que elas fiquem centradas em zero e com desvio padrão em 1 não é somente importante se iremos comparar medições com diferentes unidades de medida, mas em geral é um requisito para muitos algoritmos de machine learning. Se tomarmos como exemplo o algoritmo gradient descent, usado para minimizar as funções de custo, quando as features estão em diferentes escalas, alguns pesos irão se atualizar mais rapidamente que outros pois os valores brutos do dataset participam na atualização dos valores dos pesos.

4.4. Separação em Dataset de Treino e Testes

Na separação de datasets também é um passo importante para otimizarmos os resultados do nosso algoritmo de machine learning. O motivo dado para essa separação é para que dentro do nosso próprio dataset já tenhamos uma avalição prévia da performance do algoritmo, onde uma parcela do dataset ficará para treinamento(mais da metade) e o restante para testes. Como deve se imaginar, o dataset de testes serve como entrada para avaliarmos as predições baseadas no treinamento do dataset de treinos. Utilizar esse tipo de técnica sugere também que o dataset deve ser grande o suficiente para conter um número razoável de casos do problema a ser estudado e não cair em falhas simples de generalização.

5. Estruturação e Resgate dos dados em Memória

Após a etapa de pré-processamento é possível passar para a próxima etapa que é resgatar os registros do banco de dados, esteja ele em uma spreasheet em um arquivo .xls ou em linhas e sepearados por vírgula como um arquivo .csv ou em um SGBD como MYSQL ou Postgre. Com isso é preciso utilizar alguma linguagem de programação (junto ou não de um framework) ou uma plataforma como o Weka para trazer os dados em memória para que sejam trabalhados. Nesse processo é importante perceber que geralmente a última coluna da tabela corresponde ao resultado e por isso deve ser trazida separadamente formando um vetor de resultados. O restante da tabela de variáveis é nossa matriz de atributos.

6. Escolha de um algoritmo de Aprendizado de Máquina

A escolha de um algoritmo de apredizado de máquina deverá ser pensado de acordo com a necessidade do problema. Os problemas de aprendizado de máquina se dividem basicamente entre regressão e classificação de dados. Para compreender em qual conjunto um dado problema se encontra basta pensar se o problema tem um número contínuo de valores, como por exemplo o preço de uma ação na bolsa de valores, ou se existe um conjunto finito e discreto de valores que o nosso problema pode originar. No primeiro caso estamos falando de problemas de regressão, no segundo de classificação de dados.

7. Treinamento do algoritmo

Para o treinamento do algoritmo ter um resultado desejado e levar a predições relevantes é preciso que todas as etapas anteriores devam ter sido completas, do contrário os algortimos mesmo sendo executados, não entregarão o resultado desejado nas etapas seguintes. Para realizar o treinamento é necessário compreender a estrutura interna do algoritmo e como funciona o processo de aprendizado que será utilizado para tal. Cada algoritmo adota uma estratégia diferente apesar de sempre se basearem em conceitos comuns de erro ou custo geral dos datasets.

8. Avalição de Performance do Treinamento

Para que tenhamos uma orientação realista dos resultados oferecidos pelo treinamento do nosso algoritmo e seu modelo, é absolutamente necessario utilizar os indicadores estatísticos e compreendê-los plenamente antes de propor conclusões. Entre os indicadores estatísticos mais utilizados estão, acurácia, precisão e recall. Acurácia pode ser traduzida

como os acertos das predições do algoritmo(no dataset de testes ou em novos casos), ou seja, quando as predições do algoritmo levam a uma avaliação correta do que já aconteceu ou daquilo que se verifica como verdadeiro no futuro(estimativa de probablidade).

8.1. Classes desbalanceadas

Entretanto utilizar apenas a acurácia como único indicador de performance não é indicado e pode nos dar falsas impressões sobre uma alta performance. Se pegarmos um exemplo crítico como um classificador para diagnosticar um determinado tipo de câncer e tivermos 99% de taxa de acerto(acurácia) podemos nos impressionar e pensarmos que tenhamos um excelente resultado em mãos. O problema é se desses 99%, apenas 0.5% do dataset tinha a doença e caiu na taxa de erro, nesse caso acertando diagnósticos onde não havia câncer mas falhando onde havia. Certamente não podemos considerar um bom resultado, ainda mais se tratando de uma questão crítica como uma indicação de probabilidade de uma doença tão grave. Esse tipo de situação pode ser melhor compreendida e mensurada por outros classificadores como precisão e recall. Em outras palavras, podemos ver o problema dos emails da seguinte forma:

	Positive	**Negative**
Positive	True Positive	False Positive
Negative	False Negative	True Negative

Figura 3. Confusion Matrix

Dentro do nosso problema da predição do câncer, podemos interpretar os elementos da tabela da seguinte forma

- True Positive (TP): Número de exemplos positivos da doença e marcados como tal.
- False Positive (FP): Número de exemplos falsos da doença e marcados como positivos.
- True Negative (TN): Número de exemplos negativos da doença e marcados como tal.
- False Negative (FN): Número de exemplos positivos da doença e marcados como negativos.

Se tomarmos como exemplo o problema que estávamos analisando como o do diagnóstico de câncer, nos deparamos com dois possíveis problemas. Primeiro o erro de classificar uma pessoa como portadora da doença mas que na verdade ela não contém. Segundo o erro de não classificar uma pessoa como portadora da doença.

9. Testes

Descrever os testes que serão feitos para validar o conceito e algoritmo.

10. Cronograma

Aqui tem que descrever, por meses, quais serão as atividades, tanto de TFGI como de TFGII.

Referências

Amorim, M. J., Barone, D., and Mansur, A. U. (2008). Técnicas de aprendizado de máquina aplicadas na previsao de evasao acadêmica. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 1, pages 666–674.

Dekker, G., Pechenizkiy, M., and Vleeshouwers, J. (2009). Predicting students drop out: A case study. In *Educational Data Mining 2009*.

Delen, D. (2010). A comparative analysis of machine learning techniques for student retention management. *Decision Support Systems*, 49(4):498–506.

MARTINHO, V., Nunes, C., and Minussi, C. R. (2013). Predição do grupo de risco de evasão discente em cursos superiores presenciais utilizando uma rede neural artmap-fuzzy. In *CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA-COBENGE*, volume 41, page 2013.

Zhang, Y., Oussena, S., Clark, T., and Hyensook, K. (2010). Using data mining to improve student retention in he: a case study.