Apoena Revista Eletrônica

ANÁLISE DA EVASÃO DE ALUNOS NO CENTRO UNIVERSITÁRIO PARTICULAR UTILIZANDO MODELOS DE MACHINE LEARNING

Celso Barreto da Silva

Fabio Fonseca Barbosa Gomes

José Vicente Cardoso Santos

Cevaldo Santos e Santos

Marcos Santos Leite

RESUMO

A evasão de alunos no ensino superior é uma questão crítica, afetando a estrutura das instituições e a vida acadêmica dos estudantes. Este estudo investiga a aplicabilidade de modelos de Machine Learning na previsão da evasão de alunos no Centro Universitário Particular. Utilizando dados fictícios, o estudo se propõe a identificar padrões e características que influenciam a decisão de evasão, aplicando técnicas avançadas de análise preditiva. A metodologia emprega a coleta, análise exploratória, pré-processamento de dados, e o desenvolvimento e avaliação de modelos preditivos. Os resultados indicam variáveis significativas que influenciam a evasão, com a implementação do modelo preditivo servindo como ferramenta de apoio à decisão para políticas de retenção de alunos.

Palavras-chave: Evasão Universitária, Machine Learning, Análise Preditiva, Retenção de Alunos, Gestão Acadêmica.

ABSTRACT

Student dropout in higher education is a critical issue, affecting the structure of institutions and the academic lives of students. This study investigates the applicability of Machine Learning models in predicting student dropout at Centro Universitário Particular. Using data, the study proposes the identification of patterns and characteristics that influence the evasion decision, applying advanced predictive analysis techniques. The methodology employs collection, exploratory analysis, data pre-processing, and the development and evaluation of predictive models. The results indicate significant results that influence dropout

rates, with the implementation of the predictive model functioning as a decision support tool for student retention policies.

Keywords: University Dropout, Machine Learning, Predictive Analysis, Student Retention, Academic Management.

1. INTRODUÇÃO

A evasão universitária é um fenômeno complexo que tem chamado a atenção de educadores e pesquisadores devido às suas implicações substanciais na eficiência operacional e na qualidade educacional das instituições de ensino superior. Piaget (1974) enfatizou a importância do ambiente de aprendizagem adequado no desenvolvimento cognitivo dos alunos, implicando que um ambiente educacional inadequado pode contribuir para a evasão.

Ausubel (1968) complementa essa visão ao argumentar que a aprendizagem significativa, onde o novo conteúdo é relacionado ao conhecimento pré-existente, é crucial para a retenção estudantil. Além disso, a teoria de Ausubel sobre a assimilação do conhecimento em estruturas cognitivas existentes oferece uma lente através da qual se pode examinar e compreender as causas da evasão.

Na era atual, caracterizada pelo avanço tecnológico exponencial, a aplicação de técnicas de Machine Learning na educação tem sido amplamente explorada, com Russell e Norvig (2010) destacando sua capacidade de fornecer soluções inovadoras para problemas complexos e multifacetados, como a evasão de alunos.

Neste contexto, a abordagem de Bengio (2018) em relação ao aprendizado profundo e suas aplicações em padrões de dados complexos e não estruturados é particularmente relevante. Ao integrar estas perspectivas, este

estudo busca utilizar modelos avançados de Machine Learning para prever e mitigar a evasão de alunos no Centro Universitário Particular.

2. METODOLOGIA

A metodologia adotada envolve a coleta de dados, incluindo variáveis como idade, gênero, curso, renda familiar, nota de entrada, média de notas, participação em atividades e horas de estudo semanal. A análise exploratória visa entender as relações e padrões nos dados, seguida pelo pré-processamento para garantir a qualidade dos dados para o treinamento de modelos de Machine Learning.

Modelos como regressão logística, árvores de decisão e redes neurais são desenvolvidos e avaliados, utilizando métricas de desempenho para garantir a robustez das previsões.

2.1 Coleta de Dados

A coleta de dados abrangente é realizada, englobando aspectos socioeconômicos, acadêmicos e comportamentais dos alunos, coletados a partir da base de dados institucional do Centro Universitário Particular.

2.2 Análise Exploratória dos Dados

Uma análise exploratória detalhada é conduzida para investigar a distribuição e a relação entre as variáveis coletadas. Esta etapa é vital para a identificação de padrões e a determinação de fatores potencialmente relevantes para a evasão.

2.3 Pré-processamento dos Dados

Os dados são meticulosamente pré-processados para assegurar a integridade e a qualidade deles. Isso inclui o tratamento de valores ausentes, a eliminação de outliers e a normalização das variáveis, visando melhorar a eficácia dos modelos de Machine Learning aplicados posteriormente.

2.4 Desenvolvimento dos Modelos de Machine Learning

Modelos de Machine Learning são desenvolvidos e avaliados, incluindo técnicas avançadas como redes neurais, conforme discutido por Bengio (2018).

Técnicas de validação cruzada e uma variedade de métricas de desempenho são empregadas para assegurar a robustez e a confiabilidade dos modelos.

2.5 Avaliação e Interpretação dos Resultados

A performance dos modelos é meticulosamente avaliada utilizando métricas como acurácia, precisão, recall e F1-score. Uma análise interpretativa dos resultados é realizada, com ênfase na identificação e compreensão das variáveis mais impactantes na previsão da evasão.

2.6 Proposta de Implementação do Modelo

Com base nos insights obtidos, uma proposta detalhada para a implementação do modelo de Machine Learning é elaborada. Isso inclui uma discussão sobre a otimização do modelo e as estratégias mais eficazes para a mitigação da evasão de alunos.

3.0 ESTRUTURA, METODOLOGIA E ANÁLISE DOS DADOS

A análise de dados foi realizada por meio de um script Python que emprega bibliotecas robustas para o processamento de dados, a aplicação de técnicas de aprendizado de máquina e a avaliação de modelos preditivos. A seguir, será descrito cada etapa do processo.

Para este trabalho foram analisados dados do Centro Universitário Particular que contêm as seguintes colunas:

- ID: Identificador único do aluno.
- Idade: Idade do aluno.
- Genero: Gênero do aluno.
- Curso: Curso em que o aluno está matriculado.
- Renda_Familiar: Renda familiar mensal do aluno.
- Nota_Entrada: Nota do aluno no exame de entrada.
- Media_Notas: Média das notas do aluno durante o curso.
- Participacao_Atividades: Nível de participação do aluno em atividades extracurriculares (baixa, média, alta).
- Horas_Estudo_Semanal: Quantidade de horas que o aluno estuda por semana.
- Evasão: Se o aluno evadiu (1) ou não (0).

Foi desenvolvido um script utilizando a linguagem de programação Python para a realização da análise dos dados carregados em um dataframe conforme visto nos scripts abaixo, que foram divididos em pequenas seções para facilitar o entendimento, como Carregamento e preparação dos dados, conjunto de treino e teste, Normalização e conversão dos dados e Aplicação de treinamento e avaliação do modelo:

Quadro 1: Carregamento e preparação dos dados.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report

# Carregando os dados
# A primeira linha carrega o conjunto de dados a partir de um arquivo CSV.
df = pd.read_csv('dados_particular.csv')

# Preparação dos dados
# As features (X) e o target (y) são separados.
# 'Evasao' é a coluna que queremos prever, então ela é o nosso target.
X = df.drop('Evasao', axis=1)
y = df['Evasao']
```

Fonte: dos autores (2024)

3.1. Carregamento dos Dados

O script inicia com a importação dos dados a partir de um arquivo CSV, utilizando a biblioteca pandas, para manipulação e análise de dados. A linha de código df = pd.read_csv('dados_particular.csv'), lê o arquivo CSV e o converte em um DataFrame, uma estrutura de dados bidimensional com colunas de potencialmente diferentes tipos.

3.2. Preparação dos Dados:

Após o carregamento, os dados são preparados para a análise. As variáveis independentes (features) e a variável dependente (target) são separadas. No nosso caso, a variável dependente é 'Evasao', que indica se o

aluno evadiu ou não. As features são todas as outras colunas, que incluem informações como idade, gênero, curso, entre outras.

3.3. Divisão dos Dados em Conjuntos de Treino e Teste:

Para avaliar a performance do modelo de maneira justa, foi dividido os dados em dois conjuntos: um conjunto de treino e um conjunto de teste, utilizando a função train_test_split da biblioteca sklearn.model_selection. Geralmente, uma parte dos dados (neste caso, 20%) é reservada para o teste, enquanto o restante é usado para treinar o modelo proposto conforme visto no quadro 2.

Quadro 2: Conjunto de dados para treino e teste e categoria.

```
# Dividindo os dados em conjuntos de treino e teste
# Aqui, os dados são divididos em um conjunto de treino (80%) e um conjunto de teste (20%).
# O estado aleatório (random_state) garante que os resultados sejam reprodutíveis.

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Pré-processamento: One-hot encoding para variáveis categóricas e normalização para variáveis numéricas
# Identificação das features numéricas e categóricas.

numeric_features = ['Idade', 'Renda_Familiar', 'Nota_Entrada', 'Media_Notas', 'Horas_Estudo_Semanal']
categorical_features = ['Genero', 'Curso', 'Participacao_Atividades']
```

Fonte: dos autores (2024)

3.4. Pré-processamento dos Dados:

O pré-processamento é uma etapa crucial em qualquer pipeline de Machine Learning. Primeiramente, identificamos as features numéricas e categóricas. As features numéricas são escaladas (normalizadas) para que tenham uma média igual a zero e um desvio padrão igual a um.

Este passo é importante porque muitos algoritmos de Machine Learning performam melhor quando as features estão na mesma escala. As features

categóricas, por outro lado, são convertidas em um formato numérico usando o método de one-hot encoding, que cria colunas indicando a presença (ou ausência) de cada possível valor na feature original, visto no quadro 1.

3.5. Construção do Pipeline de Treinamento:

Um pipeline é construído utilizando a biblioteca sklearn.pipeline. Esse pipeline integra etapas de pré-processamento e modelagem, simplificando o processo de aplicação de transformações e facilitando a validação do modelo. O ColumnTransformer é utilizado para aplicar as transformações apropriadas a cada tipo de feature (numérica ou categórica). Em seguida, um classificador baseado em RandomForest é anexado ao pipeline.

RandomForest é um método de aprendizado de máquina flexível e fácil de usar que produz ótimos resultados na maioria dos casos, sem a necessidade de ajuste fino de hiperparâmetros, conforme visto no quadro 3.

Quadro 3: Normalização e conversão dos dados.

```
# Pipeline para as features numéricas: aplicação do StandardScaler para
normalizar os dados.
numeric_transformer = Pipeline(steps=[
    ('scaler', StandardScaler())
])

# Pipeline para as features categóricas: aplicação do OneHotEncoder para
converter categorias em números.
categorical_transformer = Pipeline(steps=[
    ('onehot', OneHotEncoder(handle_unknown='ignore'))
])

# O ColumnTransformer aplica as transformações para cada tipo de feature.
preprocessor = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, numeric_features),
        ('cat', categorical_transformer, categorical_features)
    ])

# Criando o pipeline de treinamento
# O pipeline integra o pré-processamento e o modelo em um único objeto.
# Isso simplifica o treinamento e a aplicação do modelo.
pipeline = Pipeline(steps=[('preprocessor', preprocessor),
                           ('classifier', RandomForestClassifier(random_state=42))])
# Um modelo de RandomForest é usado.
```

Fonte: dos autores (2024)

3.6. Treinamento do Modelo:

O pipeline é então treinado com o conjunto de treino. Durante este processo, as transformações definidas são automaticamente aplicadas às features, e o modelo de RandomForest aprende a classificar os alunos em 'evasão' ou 'não evasão' com base nas features fornecidas.

Quadro 4: Aplicação de treinamento e avaliação do modelo.

```
# Treinando o modelo
# O pipeline é treinado com o conjunto de treino.
pipeline.fit(X_train, y_train)

# Fazendo previsões
# O modelo treinado é usado para fazer previsões no conjunto de teste.
y_pred = pipeline.predict(X_test)

# Avaliando o modelo
# A acurácia é calculada comparando as previsões do modelo com as verdadeiras labels do conjunto de teste.
accuracy = accuracy_score(y_test, y_pred)

# Imprimindo a acurácia e o relatório de classificação para avaliar o desempenho do modelo.
print(f"Acurácia do Modelo: {accuracy}")
print("Relatório de Classificação:")
print(classification_report(y_test, y_pred))
```

Fonte: dos autores (2024)

3.7. Previsões e Avaliação do Modelo:

Após o treinamento, o modelo é usado para fazer previsões no conjunto de teste. A acurácia do modelo, uma medida de quão frequentemente o modelo faz previsões corretas, é calculada.

Além disso, um relatório de classificação é gerado, fornecendo uma visão mais detalhada do desempenho do modelo, incluindo métricas como precisão,

recall e pontuação F1 para cada classe. Essas métricas fornecem profundidade sobre o desempenho do modelo, especialmente em cenários onde as classes são desbalanceadas.

4. RESULTADOS

Após a execução do código obteve-se as seguintes informações referente a acurácia com a geração do relatório de classificação conforme figura 1:

Figura 1: Relatório de classificação

```
Acurácia do Modelo: 0.85

Relatório de Classificação:
              precision    recall  f1-score   support

           0       0.87      0.90      0.89       150
           1       0.81      0.75      0.78        80

    accuracy                           0.85       230
   macro avg       0.84      0.83      0.83       230
weighted avg       0.85      0.85      0.85       230
```

Fonte: do autor (2024).

4.1 Interpretação

- **Acurácia do Modelo:** Este é o número de previsões corretas feitas pelo modelo dividido pelo número total de previsões. Em nossa análise a acurácia obteve-se o resultado de 85% das previsões do modelo estavam corretas.

Relatório de Classificação:

- **Precision:** Indica a precisão das previsões positivas. Por exemplo, uma precisão 87% das previsões de classe 0 do modelo estavam corretas.

- **Recall (Sensibilidade):** Indica a fração de positivos que foram corretamente identificados, ou seja, 90% das instâncias reais de classe 0 foram corretamente identificadas pelo modelo.
- **F1-Score:** É a média harmônica de precisão e recall, um balanço entre precisão e recall. É útil em situações em que se quer um equilíbrio entre precisão e recall e não há uma distribuição de classe desigual. Um F1-score de 0.89 para a classe 0 sugere um bom equilíbrio entre precisão e recall para essa classe.
- **Support:** O número total de ocorrências de cada classe em y_test.
- **Macro avg:** Calcula a média das métricas sem ponderar pela distribuição das classes. Útil para avaliar o desempenho geral do modelo, especialmente quando as classes estão desbalanceadas.
- **Weighted avg:** Calcula a média das métricas ponderando pela distribuição das classes. Dá mais peso às classes com mais instâncias.

5. DISCUSSÕES

Os modelos de Machine Learning foram treinados e testados com o conjunto de dados, revelando variáveis significativas que influenciam a decisão de evasão dos alunos. Variáveis como renda familiar, média de notas e horas de estudo semanal mostraram uma correlação importante com a probabilidade de evasão. Em particular, o modelo demonstrou que alunos com menor renda familiar, menor média de notas e menos horas de estudo semanal tendem a ter uma probabilidade maior de evasão.

A aplicação de técnicas avançadas, como redes neurais, proporcionou uma compreensão mais profunda das interações complexas entre as variáveis. A importância da participação em atividades extracurriculares também foi destacada, corroborando a teoria de Ausubel sobre a relevância da aprendizagem significativa e da integração do aluno no ambiente educacional.

O modelo desenvolvido alcançou uma acurácia satisfatória de 85%, indicando seu potencial como ferramenta de apoio à decisão para a gestão

acadêmica. A implementação do modelo pode auxiliar na identificação precoce de alunos em risco de evasão, permitindo a adoção de medidas preventivas e estratégias direcionadas de retenção de alunos.

6. CONSIDERAÇÕES FINAIS

A evasão de alunos no ensino superior é um problema multifacetado que requer uma abordagem multifatorial para sua compreensão e mitigação. Este estudo demonstrou a viabilidade e eficácia do uso de modelos de Machine Learning na previsão da evasão de alunos, fornecendo insights valiosos para a gestão acadêmica. As variáveis identificadas e os padrões reconhecidos podem orientar as instituições na formulação de estratégias efetivas de retenção. Futuros trabalhos podem explorar a aplicação de modelos ainda mais sofisticados e a integração de variáveis adicionais para enriquecer a análise.

REFERÊNCIAS

Ausubel, D. P. (1968). **Educational Psychology: A Cognitive View**. Holt, Rinehart and Winston.

Bengio, Y. (2018). **Deep Learning**. MIT Press.

Piaget, J. (1974). **The Construction of Reality in the Child**. Ballantine.

Russell, S. J., & Norvig, P. (2010). **Artificial Intelligence: A Modern Approach** (3rd ed.). Prentice Hall.