

3.1 - presc

17 de julho de 2024



historico_estratificado.csv
Planilha

Analise os dados e limpe as colunas da seguinte forma:
período_ingresso: de 2006.1 até 2019.2
tipo_matricula: remover "Dispensa"
situacao: remover "Em Curso"
o novo dataframe após a limpeza deve se chamar df_filtrado



Vamos começar lendo o arquivo que você enviou para entender a estrutura dos dados. Depois, vou seguir as etapas de limpeza conforme você especificou. Primeiro, vamos carregar e visualizar os primeiros registros do arquivo.

O arquivo carregado possui diversas colunas. Agora, vou aplicar os critérios de limpeza que você especificou:

1. **período_ingresso:** Deve ser entre 2006.1 e 2019.2.
2. **tipo_matricula:** Remover as entradas que contêm "Dispensa".
3. **situacao:** Remover as entradas que contêm "Em Curso".

Vamos aplicar esses filtros e criar o novo DataFrame chamado `df_filtrado`. [?]

O novo DataFrame, `df_filtrado`, foi criado com as condições especificadas. Ele contém entradas de períodos de ingresso entre 2006.1 e 2019.2, exclui registros com "Dispensa" em `tipo_matricula` e "Em Curso" em `situacao`.

Se precisar de mais alguma análise ou modificação nesses dados, estou à disposição! [?]

Por meio de uma análise avançada, com base na redução de dimensionalidade das características acadêmicas dos alunos, como podemos prever quem são os alunos em risco de evasão? Defina 3 opções de como solucionar essa questão e siga a melhor



Para prever alunos em risco de evasão com base nas características acadêmicas, podemos aplicar técnicas de aprendizado de máquina, particularmente aquelas que envolvem redução de dimensionalidade e modelagem preditiva. A seguir, detalho três abordagens possíveis e depois sugiro qual seria a melhor opção para seguirmos:

1. PCA com Modelagem Clássica

Descrição: Utilizar a Análise de Componentes Principais (PCA) para reduzir a dimensionalidade dos dados, seguido pela aplicação de um modelo de classificação tradicional (como regressão logística, árvores de decisão ou máquinas de vetores de suporte).

- **Vantagens:** Reduz ruído e melhora a eficiência computacional; fácil interpretação dos componentes principais.
- **Desvantagens:** Assume linearidade entre as características; pode perder informações importantes contidas nas componentes menores.

2. T-SNE com Clustering

Descrição: Aplicar T-distributed Stochastic Neighbor Embedding (t-SNE) para projetar os dados em um espaço de baixa dimensionalidade e, em seguida, usar técnicas de clustering (como k-means ou DBSCAN) para identificar grupos, potencialmente indicando alunos em risco.

- **Vantagens:** Excelente para capturar a estrutura não linear dos dados; pode revelar padrões ocultos eficazmente.
- **Desvantagens:** Computacionalmente intensivo; a interpretação dos resultados pode ser desafiadora.

3. Autoencoders com Modelagem Supervisionada

Descrição: Utilizar autoencoders, uma forma de aprendizado profundo não supervisionado, para aprender uma representação de baixa dimensão dos dados. Em seguida, usar essas novas características para alimentar um modelo preditivo supervisionado.

- **Vantagens:** Capaz de capturar complexidades nos dados através de uma abordagem não linear; bom para grandes conjuntos de dados.
- **Desvantagens:** Requer um bom entendimento de redes neurais; mais difícil de treinar e afinar.

Sugestão e Implementação

Considerando a complexidade dos dados e a necessidade de um modelo robusto que possa capturar nuances nas trajetórias acadêmicas dos alunos, a **terceira opção** parece ser a mais promissora. Autoencoders, seguidos por um modelo de classificação, podem oferecer um bom equilíbrio entre capacidade de generalização e precisão preditiva, especialmente com dados multidimensionais complexos como os acadêmicos.

Próximos Passos:

1. Preparar os dados (limpeza, normalização, tratamento de valores ausentes).
2. Implementar um autoencoder para reduzir a dimensionalidade.
3. Treinar um modelo de classificação com as características reduzidas.
4. Avaliar o modelo usando métricas adequadas (como precisão, recall e AUC).

Se concordar com essa abordagem, posso começar com a preparação dos dados para modelagem.

Siga a melhor opção para solucionar esse problema



Perfeito! Vamos seguir com a terceira opção, que utiliza autoencoders para redução de dimensionalidade seguida de modelagem supervisionada para prever alunos em risco de evasão.

Etapa 1: Preparação dos Dados

Antes de implementarmos o autoencoder, precisamos preparar os dados. Isso inclui:

- **Selecionar características relevantes:** Focar em variáveis que representam desempenho acadêmico, como médias de notas, número de reprovações, e características demográficas que podem influenciar o risco de evasão.
- **Tratar valores ausentes:** Preencher ou remover dados ausentes, dependendo da quantidade e importância da variável.
- **Normalizar os dados:** Escalar os dados para que tenham média zero e variância unitária, o que é essencial para o treinamento eficaz de redes neurais.

Vamos começar com a seleção e tratamento de características, e então normalizar esses dados.

Aqui estão as informações sobre os valores ausentes nas características selecionadas:

- **apv_media_geral:** 3.23% de valores ausentes
- **media_final:** 9.93% de valores ausentes

Para tratar esses valores ausentes, consideraremos a seguinte abordagem:

- **Preencher valores ausentes:** Utilizar a média da coluna para variáveis contínuas como ``apv_media_geral`` e ``media_final``, pois são relativamente poucos os valores ausentes e isso não deverá distorcer significativamente a distribuição dos dados.

Agora, vamos tratar os valores ausentes e depois prosseguir com a normalização dos dados. Também precisaremos transformar variáveis categóricas em numéricas usando codificação adequada. [?]

Os dados foram preparados com sucesso, incluindo o preenchimento de valores ausentes, a codificação de variáveis categóricas e a normalização de variáveis numéricas. Agora temos um conjunto de dados pronto para ser usado em um modelo de aprendizado de máquina.

Etapa 2: Implementação do Autoencoder

O próximo passo é implementar um autoencoder para reduzir a dimensionalidade. O autoencoder é uma rede neural que aprende a compactar os dados em um espaço latente de menor dimensão e depois reconstruí-los. Vamos projetar um autoencoder simples e treiná-lo com nossos dados.

Vou prosseguir com a configuração e treinamento do autoencoder. [-]

Parece que estamos com um problema técnico: o ambiente atual não tem o TensorFlow e o Keras instalados, que são necessários para implementar e treinar o autoencoder. Como alternativa, posso demonstrar como proceder com um modelo semelhante usando bibliotecas que estejam disponíveis, ou posso fornecer o código e as instruções para que você execute em seu próprio ambiente de desenvolvimento que tenha essas bibliotecas.

Por favor, me avise como gostaria de proceder! [-]

Comece com Data Analyst

[Denunciar conteúdo](#) | [Termos de uso](#) | [Política de privacidade](#) |

?