

Desafio Técnico – Lider de IA

Analisar o conjunto de dados "students_dropout.csv" para prever a evasão escolar, realizar uma análise exploratória de dados (AED) aprofundada, construir e avaliar um modelo de aprendizado de máquina básico para previsão, investigar aspectos de equidade do desempenho do modelo em diferentes grupos, propor soluções para mitigar qualquer viés identificado e resumir as descobertas, incluindo insights, lições aprendidas e recomendações para etapas futuras.

DataFrame 'df' recarregado com sucesso com o separador correto. Exibindo as 5 primeiras linhas:

5 rows ▾ 5 rows × 37 cols													Static Output	
⚙	Marital status	⚙	Application mode	⚙	Application order	⚙	Course	⚙	Daytime/evening attendance\t	⚙	Previous qualification	⚙	Previous	
0		1		17		5	171			1			1	
1		1		15		1	9254			1			1	
2		1		1		5	9070			1			1	
3		1		17		2	9773			1			1	
4		2		39		1	8014			0			1	

Para obter informações sobre a tendência central, dispersão e forma da distribuição de características numéricas, calcularei estatísticas descritivas para todas as colunas numéricas usando `df.describe()`.

DESCRIPTIVE STATISTICS FOR NUMERICAL COLUMNS:

Descriptive Statistics for Numerical Columns:														
	Marital status	Application mode	Application order	Course	Daytime/evening attendance\t	Previous qualification	Previous qualification (grade)	Nationality	Mother's qualification	Father's qualification	...	Curricular units 1st sem (without evaluations)	Curricular units 2nd sem (credited)	Curricu units (enroll
count	4424.000000	4424.000000	4424.000000	4424.000000	4424.000000	4424.000000	4424.000000	4424.000000	4424.000000	4424.000000	...	4424.000000	4424.000000	4424.000000
mean	1.178571	18.669078	1.727848	8856.642631	0.890823	4.577758	132.613314	1.873192	19.561935	22.275316	...	0.137658	0.541817	6.230000
std	0.605747	17.484682	1.313793	2063.566416	0.311897	10.216592	13.188332	6.914514	15.603186	15.343108	...	0.690880	1.918546	2.190000
min	1.000000	1.000000	0.000000	33.000000	0.000000	1.000000	95.000000	1.000000	1.000000	1.000000	...	0.000000	0.000000	0.000000
25%	1.000000	1.000000	1.000000	9085.000000	1.000000	1.000000	125.000000	1.000000	2.000000	3.000000	...	0.000000	0.000000	5.000000
50%	1.000000	17.000000	1.000000	9238.000000	1.000000	1.000000	133.100000	1.000000	19.000000	19.000000	...	0.000000	0.000000	6.000000
75%	1.000000	39.000000	2.000000	9556.000000	1.000000	1.000000	140.000000	1.000000	37.000000	37.000000	...	0.000000	0.000000	7.000000
max	6.000000	57.000000	9.000000	9991.000000	1.000000	43.000000	190.000000	109.000000	44.000000	44.000000	...	12.000000	19.000000	23.000000

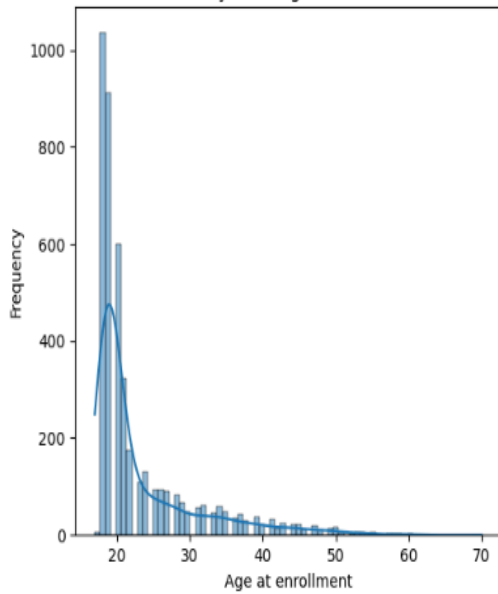
8 rows x 36 columns

Para garantir a qualidade dos dados e identificar colunas que possam exigir imputação ou remoção, identificarei e contarei o número de valores ausentes em cada coluna do DataFrame `df`.

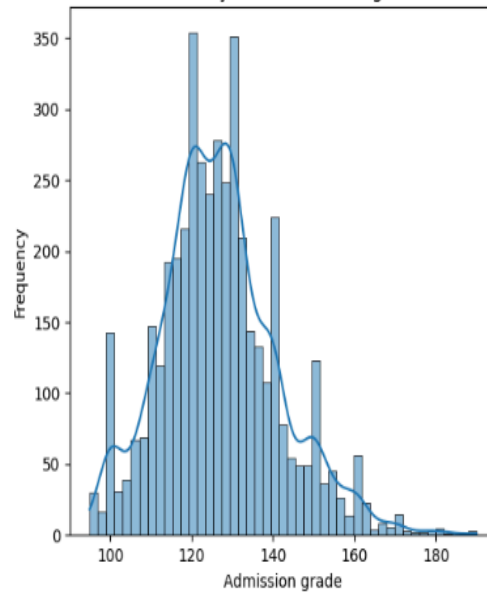
Valores ausentes por coluna:	
	0
Marital status	0
Application mode	0
Application order	0
Course	0
Daytime/evening attendance\t	0
Previous qualification	0
Previous qualification (grade)	0
Nacionality	0
Mother's qualification	0
Father's qualification	0
Mother's occupation	0
Father's occupation	0
Admission grade	0
Displaced	0
Educational special needs	0
Debtor	0
Tuition fees up to date	0
Gender	0
Scholarship holder	0

Para visualizar a distribuição das colunas numéricas seleccionadas, vou criar histogramas para 'Idade na matrícula', 'Nota de admissão', 'Taxa de desemprego', 'Taxa de inflação' e 'PIB'.

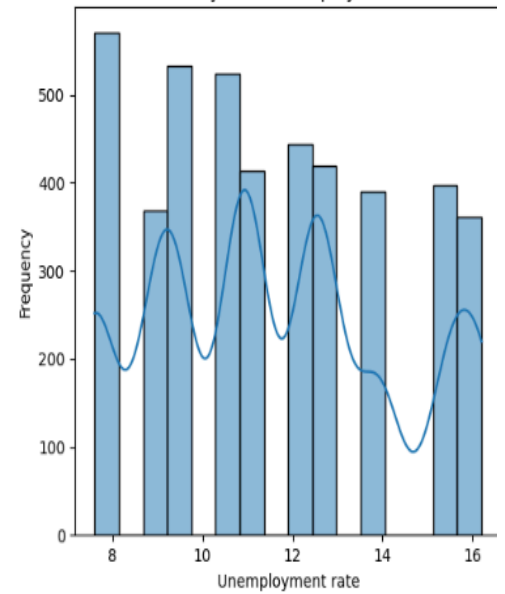
Distribuição de Age at enrollment



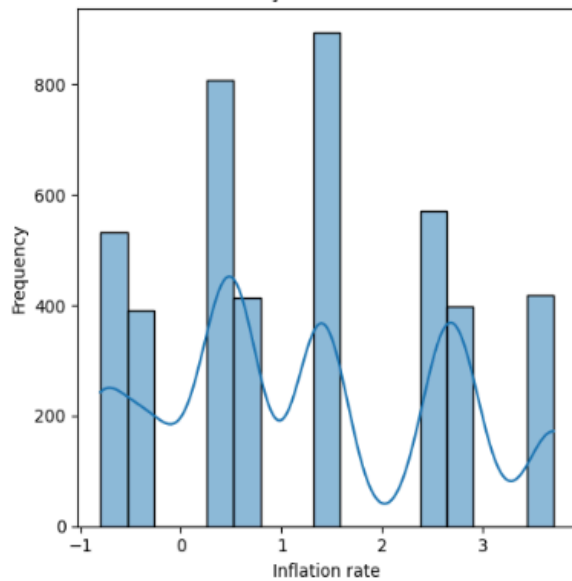
Distribuição de Admission grade



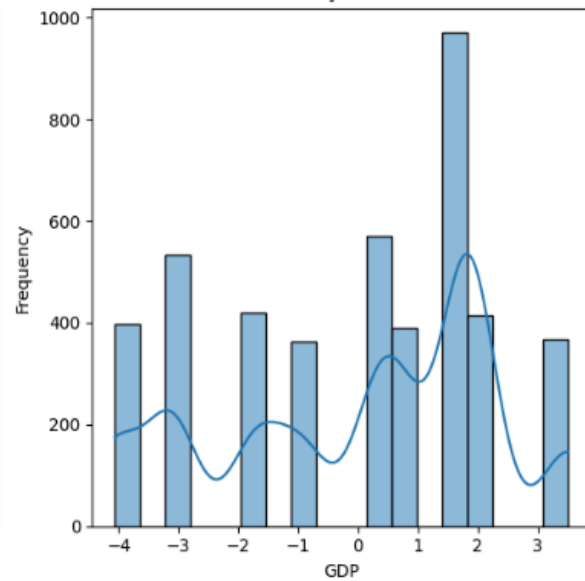
Distribuição de Unemployment rate



Distribuição de Inflation rate

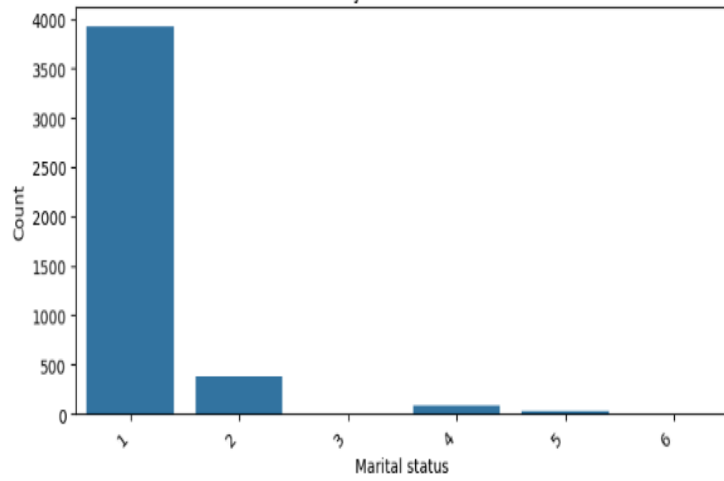


Distribuição de GDP

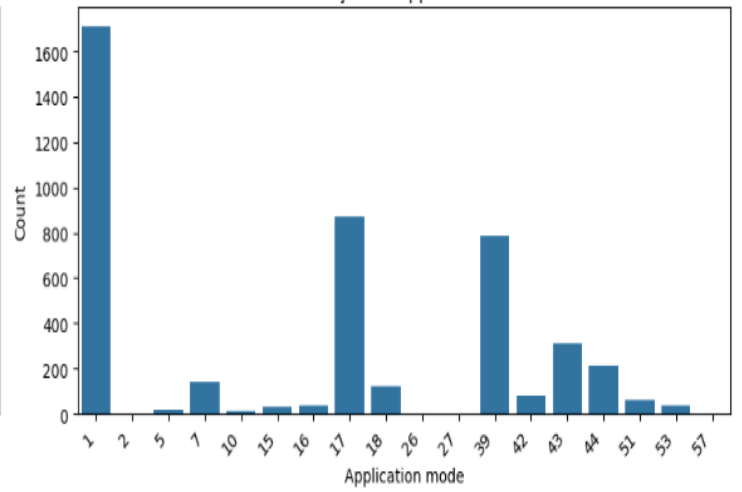


Para visualizar a distribuição das colunas categóricas importantes, criei gráficos de barras para 'Estado civil', 'Modo de inscrição', 'Sexo', 'Bolsista' e 'Objetivo'.

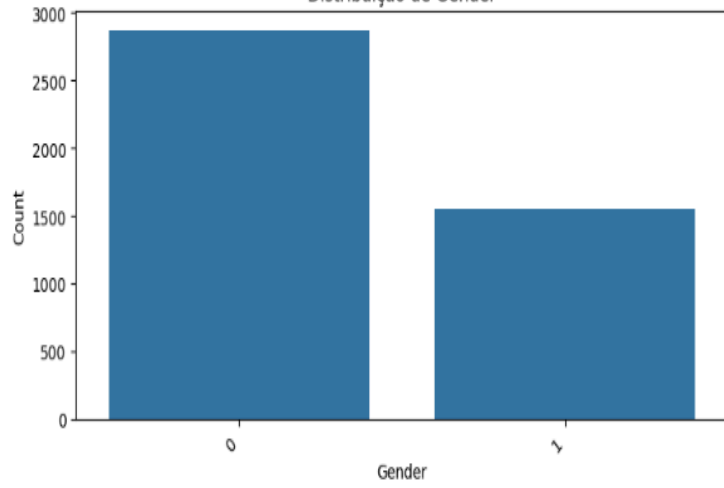
Distribuição de Marital status



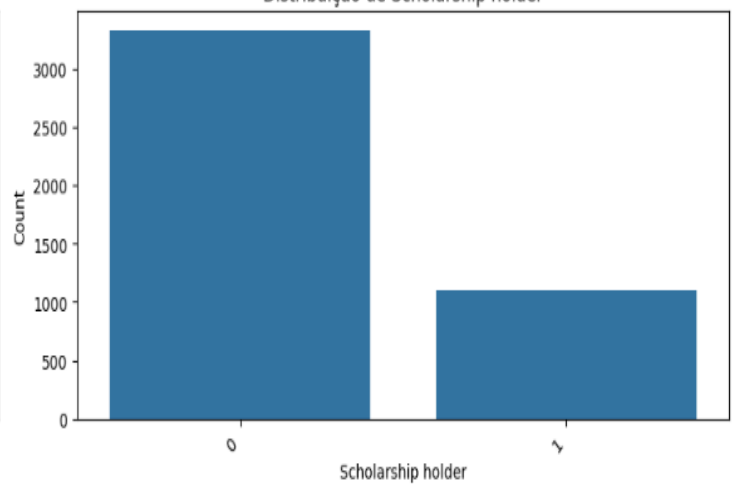
Distribuição de Application mode



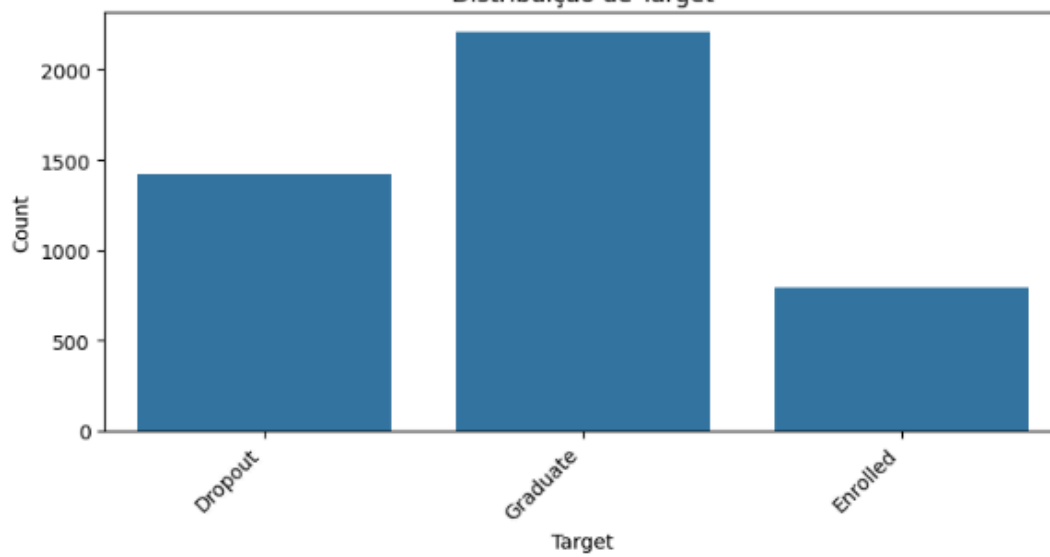
Distribuição de Gender



Distribuição de Scholarship holder



Distribuição de Target



None

Pré-processamento para Modelo Baseline

Preparar os dados para o treinamento do modelo, incluindo tratamento de valores ausentes, codificação de variáveis categóricas (se houver), escalonamento de features e divisão do dataset em conjuntos de treino e teste. O primeiro passo no pré-processamento é converter a variável alvo 'Target' em um formato numérico adequado para modelos de aprendizado de máquina. Vou mapear 'Dropout' para 1 e 'Graduate'/'Enrolled' para 0.

Coluna 'Target' convertida para o formato numérico 'Target_Numerical'. Exibindo a contagem de valores:

Target_Numerical

0 3003

1 1421

Name: count, dtype: int64

Para preparar as variáveis categóricas para modelos de aprendizado de máquina, aplicarei a codificação one-hot às colunas identificadas. Isso converte as variáveis categóricas em um formato numérico, essencial para muitos algoritmos. Também limparei os nomes das colunas após a codificação one-hot, substituindo caracteres especiais e espaços por sublinhados para consistência e facilidade de acesso.

DataFrame após codificação one-hot e limpeza dos nomes das colunas. Exibindo as 5 primeiras linhas.:

5 rows ▾ 5 rows x 257 cols Static Output

÷	Application_order	÷	Previous_qualification_grade	÷	Admission_grade	÷	Age_at_enrollment	÷	Curricular_units_1st_sem_credited	÷	Curricular_units_2nd_sem_credited
0	5		122.0		127.3		20		0		0
1	1		160.0		142.5		19		0		0
2	5		122.0		124.8		19		0		0
3	2		122.0		119.6		20		0		0
4	1		100.0		141.5		45		0		0

Agora que as variáveis categóricas estão codificadas e a variável alvo é numérica, separarei as variáveis (X) da variável alvo (y) para preparar o treinamento do modelo. As colunas 'Target' e 'Target_Numerical', juntamente com quaisquer colunas categóricas originais, devem ser removidas do conjunto de variáveis.

```
X = df_encoded.drop(columns=['Target', 'Target_Numerical'])
y = df_encoded['Target_Numerical']

print("Características (X) e alvo (y) separados.")
print("Shape of X:", X.shape)
print("Shape of y:", y.shape)
[46]
```

```
Características (X) e alvo (y) separados.
Shape of X: (4424, 255)
Shape of y: (4424,)
```

Para preparar os dados para o treinamento e avaliação do modelo, dividirei as características (X) e o alvo (y) em conjuntos de treinamento e teste usando `train_test_split`. Isso garante que o modelo seja treinado em uma parte dos dados e avaliado em dados não vistos, proporcionando uma avaliação mais robusta de seu desempenho. Definir um `random_state` tornará a divisão reproduzível.

```
from sklearn.model_selection import train_test_split

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

print("Dados divididos em conjuntos de treinamento e teste.")
print(f"X_train shape: {X_train.shape}")
print(f"X_test shape: {X_test.shape}")
print(f"y_train shape: {y_train.shape}")
print(f"y_test shape: {y_test.shape}")
[47]
```

```
Dados divididos em conjuntos de treinamento e teste.
X_train shape: (3539, 255)
X_test shape: (885, 255)
y_train shape: (3539,)
y_test shape: (885,)
```

Para garantir que todas as características numéricas contribuam igualmente para o treinamento do modelo e para evitar que características com valores maiores dominem, aplicarei a padronização (MinMaxScaler) às colunas numéricas dos

conjuntos de treinamento e teste. Isso dimensionará as características para um intervalo comum.

Características numéricas escaladas usando MinMaxScaler.

Primeiras 5 linhas de X_train escalado:

	Previous_qualification_grade	Admission_grade	Age_at_enrollment	\
571	0.368421	0.313684	0.111111	
155	0.357895	0.453684	0.044444	
912	0.473684	0.263158	0.466667	
1972	0.357895	0.362105	0.044444	
2951	0.452632	0.354737	0.022222	

	Curricular_units_1st_sem_credited	Curricular_units_1st_sem_enrolled	\
--	-----------------------------------	-----------------------------------	---

571	0.0	0.192308
155	0.0	0.230769
912	0.0	0.192308
1972	0.0	0.230769
2951	0.0	0.230769

	Curricular_units_1st_sem_evaluations	Curricular_units_1st_sem_approved	\
571	0.200000	0.000000	
155	0.177778	0.230769	

Treinamento do Modelo Baseline

Construir e treinar um modelo de machine learning simples como baseline (e.g., Regressão Logística, Árvore de Decisão) para prever o abandono de alunos. O objetivo é estabelecer uma métrica de desempenho inicial. Para estabelecer um modelo de referência, vou instanciar um modelo de Regressão Logística com um estado aleatório especificado para fins de reprodutibilidade e, em seguida, treiná-lo usando os dados de treinamento pré-processados.

```

from sklearn.linear_model import LogisticRegression

# Crie uma instância do modelo de Regressão Logística
model = LogisticRegression(random_state=42, max_iter=1000) # Increased max_iter to prevent convergence warning

# Treine o modelo
model.fit(X_train, y_train)

print("Modelo de regressão logística treinado com sucesso.")

```

Modelo de regressão logística treinado com sucesso.

Para avaliar o desempenho do modelo de referência, calcularei e exibirei um relatório de classificação e a pontuação de precisão, que fornecem uma visão geral abrangente da precisão, recall, pontuação F1 e correção geral do modelo.

Relatório de Classificação:

	precision	recall	f1-score	support
0	0.89	0.96	0.92	601
1	0.89	0.74	0.81	284
accuracy			0.89	885
macro avg	0.89	0.85	0.87	885
weighted avg	0.89	0.89	0.89	885

Avaliação do Modelo Baseline

Avaliar o desempenho do modelo baseline usando métricas apropriadas para problemas de classificação (e.g., acurácia, precisão, recall, F1-score, AUC-ROC).

Para avaliar melhor o desempenho do modelo de base, especialmente sua capacidade de discriminar entre classes, calcularei a pontuação AUC-ROC usando probabilidades de predição e gerarei uma matriz de confusão. Isso proporcionará

uma compreensão mais detalhada de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos.

```
from sklearn.metrics import roc_auc_score, confusion_matrix

# Obter probabilidades de previsão para a classe positiva (classe 1 - Desistência)
y_pred_proba = model.predict_proba(X_test)[:, 1]

# Calcular a pontuação AUC-ROC
auc_roc = roc_auc_score(y_test, y_pred_proba)
print(f"AUC-ROC Score: {auc_roc:.4f}")

# Gerar Matriz de Confusão
conf_matrix = confusion_matrix(y_test, y_pred)
print("\nMatriz de Confusão:")
print(conf_matrix)
```

AUC-ROC Score: 0.9280

Matriz de Confusão:

```
[[576  25]
 [ 74 210]]
```

Configuração da Análise de Fairness

Identificar atributos sensíveis e definir métricas de fairness.

Para definir atributos sensíveis para análise de equidade, primeiro inspecionarei a contagem de valores de 'Gênero' e 'Estado civil' no DataFrame original para entender sua distribuição e identificar possíveis grupos protegidos. Isso orientará a criação de novas colunas binárias em `df_encoded` representando esses grupos.

[53]

Contagem de valores para Gênero:

Gender

0 2868

1 1556

Name: count, dtype: int64

Contagem de valores para Estado Civil:

Marital status

1 3919

2 379

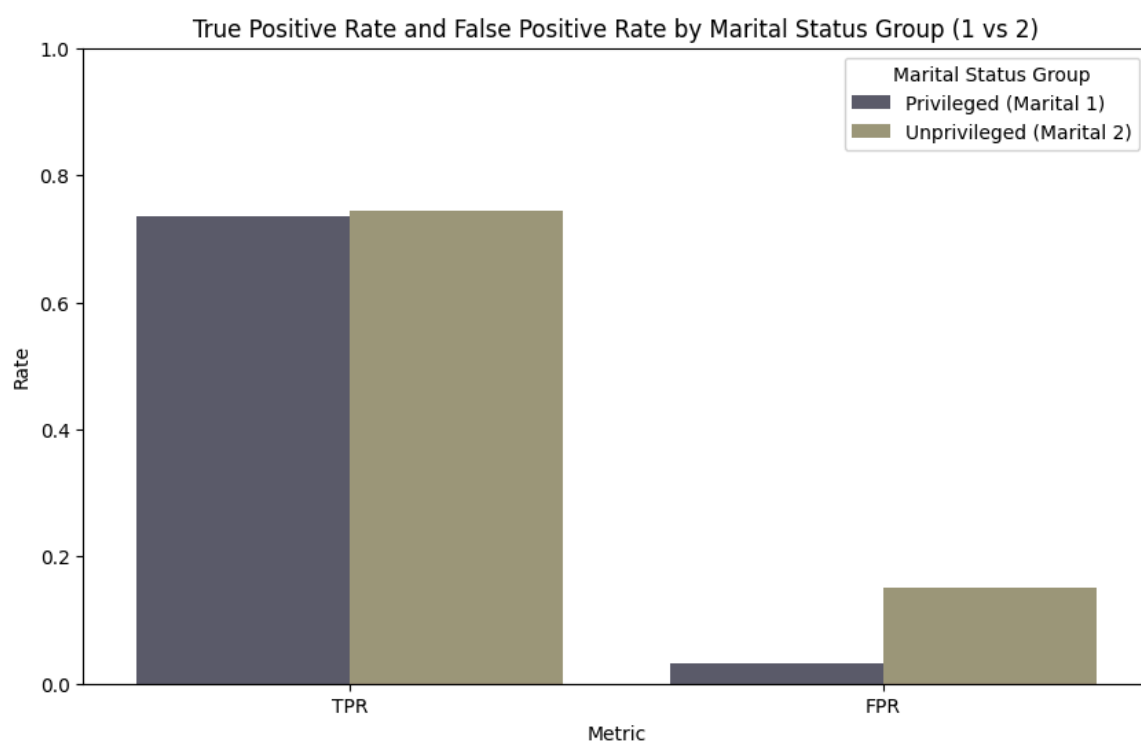
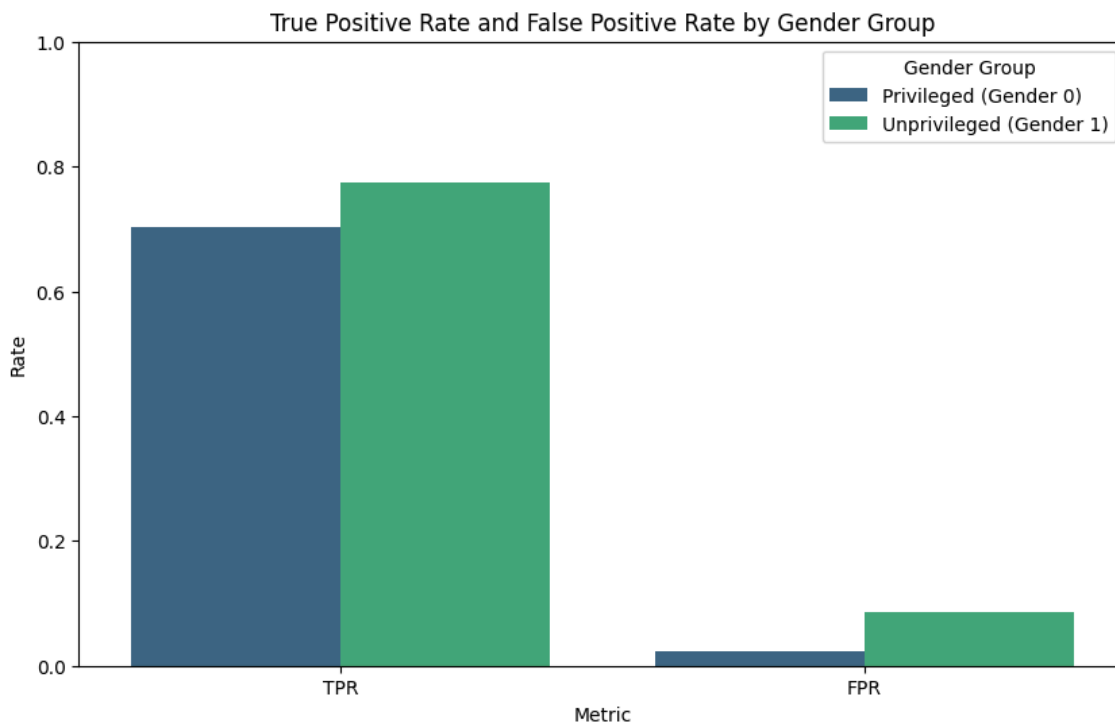
4 91

5 25

6 6

3 4

Name: count, dtype: int64



--- Métricas de equidade para gênero ---

Diferença de Oportunidades Iguais (Gender): 0.0711

Diferença de Paridade Demográfica (Gender): 0.2090

Métricas de equidade para o estado civil (comparing Marital_status_1 vs Marital_status_2)
Diferença de Oportunidades Iguais (Marital Status): 0.0097
Diferença de Paridade Demográfica (Marital Status): 0.2204

Interpretação de métricas de equidade

Diferença de Oportunidades Iguais (EOD):

Measures the difference in True Positive Rate (TPR) between the unprivileged and privileged groups. An EOD close to 0 indicates that the model has similar success rates in correctly identifying the positive outcome (dropout) for both groups. Diferença de Paridade Demográfica (DPD):

Measures the difference in the proportion of positive predictions between the unprivileged and privileged groups. A DPD close to 0 indicates that the model predicts the positive outcome (dropout) at similar rates for both groups, regardless of their actual outcome.

Análise de Resultados de Fairness e Proposta de Mitigação de Viés

Resumo do Projeto: Análise de Abandono de Alunos

Este relatório apresenta os principais resultados da análise do dataset *students_dropout.csv*, incluindo a Análise Exploratória de Dados (EDA), o desempenho de um modelo baseline de Regressão Logística, a investigação de fairness com foco em atributos sensíveis e as estratégias propostas para mitigação de vieses identificados.

1. Análise Exploratória de Dados (EDA)

1.1 Qualidade dos Dados

O dataset foi carregado com sucesso e não apresentou valores ausentes em nenhuma das colunas, o que simplifica o pré-processamento e garante a integridade das informações utilizadas no modelo.

1.2 Distribuição da Variável Alvo

A variável alvo original contém três classes: *Graduate* (2209), *Dropout* (1421) e *Enrolled* (794). Para a formulação do problema de classificação binária, as classes *Graduate* e *Enrolled* foram agregadas na classe *Não-Dropout* (0), resultando em 3003 instâncias de *Não-Dropout* e 1421 de *Dropout* (1), caracterizando um desequilíbrio de classes.

1.3 Distribuição de Variáveis Relevantes

- **Idade no momento da matrícula:** predominância de alunos mais jovens, com cauda para idades mais elevadas.
- **Nota de admissão:** distribuição concentrada em torno da média, com variações moderadas.
- **Variáveis macroeconômicas (desemprego, inflação e PIB):** refletem diferentes contextos econômicos ao longo do período analisado.
- **Variáveis categóricas:** atributos como *Marital status*, *Application mode* e *Course* exigem codificação; *Gender* e *Scholarship holder* são binários e relevantes para a análise de fairness.

2. Desempenho do Modelo Baseline

Foi treinado um modelo de Regressão Logística como baseline para a predição de abandono.

2.1 Métricas de Avaliação

- **Acurácia:** 0,8881 (88,81%)
- **AUC-ROC:** 0,9280

2.2 Matriz de Confusão

- Verdadeiros Positivos (TP): 210
- Verdadeiros Negativos (TN): 576
- Falsos Positivos (FP): 25
- Falsos Negativos (FN): 74

2.3 Análise de Desempenho

O modelo apresenta excelente capacidade de discriminação entre as classes. Observa-se alto recall para a classe *Não-Dropout* (96%), enquanto o recall para *Dropout* é inferior (74%), indicando que aproximadamente 26% dos alunos que efetivamente abandonam não são identificados. Esse aspecto é crítico, dado o alto custo associado aos falsos negativos em contextos educacionais.

3. Análise de Fairness

A análise de equidade foi conduzida com a biblioteca **AIF360**, considerando os atributos sensíveis *Gênero* (0: privilegiado, 1: não-privilegiado) e *Estado Civil* (1: privilegiado, 2: não-privilegiado). As métricas utilizadas foram:

- **Equal Opportunity Difference (EOD):** diferença na Taxa de Verdadeiros Positivos (TPR) entre grupos.
- **Demographic Parity Difference (DPD):** diferença na proporção de previsões positivas entre grupos.

3.1 Resultados para Gênero

- **EOD:** 0,0711
- **DPD:** 0,2090

Esses resultados indicam uma disparidade relevante na paridade demográfica. O grupo não-privilegiado apresenta TPR ligeiramente superior, sugerindo maior eficácia do modelo na identificação de abandono para esse grupo, enquanto a elevada DPD aponta para previsões positivas em proporções distintas entre os gêneros.

3.2 Resultados para Estado Civil

- **EOD:** 0,0097
- **DPD:** 0,2204

Para estado civil, a EOD próxima de zero indica desempenho semelhante em termos de TPR. No entanto, a DPD elevada revela que o modelo prevê abandono em taxas significativamente diferentes entre os grupos, configurando um problema de equidade.

3.3 Síntese dos Vieses Identificados

Ambos os atributos sensíveis apresentam viés de paridade demográfica. O viés de igualdade de oportunidade é mais pronunciado para gênero, embora em menor magnitude do que a DPD.

4. Estratégias Propostas para Mitigação de Viés

4.1 Reweighting (Pré-processamento)

Descrição: Técnica que ajusta os pesos das instâncias no conjunto de treinamento para equilibrar a distribuição dos rótulos entre grupos privilegiados e não-privilegiados, reduzindo correlações espúrias entre atributos sensíveis e o desfecho.

Vantagens:

- Simples de implementar.
- Independente do algoritmo de classificação.
- Atua diretamente nos dados de entrada.

Limitações:

- Pode impactar a acurácia global.
- Não garante a eliminação de todos os tipos de viés.

4.2 Otimização de Limiares (Pós-processamento)

Descrição: Ajuste de limiares de decisão específicos para cada grupo protegido, visando otimizar métricas de fairness como Equal Opportunity ou Equalized Odds.

Vantagens:

- Dispensa re-treinamento do modelo.
- Flexível quanto às métricas de equidade priorizadas.

Limitações:

- Pode reduzir o desempenho global.
- Exige calibração cuidadosa, especialmente ao otimizar múltiplos critérios.

5. Recomendações e Trabalhos Futuros

- Avaliar empiricamente o impacto das estratégias de mitigação propostas nas métricas de fairness e desempenho.
- Investigar outros atributos potencialmente sensíveis, como nacionalidade e escolaridade dos pais.
- Explorar modelos mais complexos (e.g., Random Forest, Gradient Boosting) e técnicas de balanceamento de classes (e.g., SMOTE) para melhorar o recall da classe *Dropout*, mantendo atenção às questões de equidade.

6. Métricas de Equidade por Status de Bolsista:

Não Bolsistas (Scholarship holder_1 = False): Apresentam uma acurácia de 87,31%, precisão de 87,72%, recall de 78,43% e F1-score de 82,82%. O ROC-AUC para este grupo é de 0,9337. Bolsistas (Scholarship holder_1 = True): Possuem uma acurácia mais alta (93,51%), precisão de 88,89%, mas um recall significativamente menor (55,17%) e F1-score de 68,09%. O ROC-AUC para este grupo é de 0,8715.

Métricas de Equidade por Gênero:

Gênero 0 (Gender_1 = False): Demonstrem uma acurácia de 91,49%, precisão de 90,18%, recall de 73,19% e F1-score de 80,80%. O ROC-AUC para este grupo é de 0,9257.

Gênero 1 (Gender_1 = True): Têm uma acurácia ligeiramente menor (84,42%), precisão de 85,82%, recall de 78,77% e F1-score de 82,14%. O ROC-AUC para este grupo é de 0,9223. Para o gênero, os grupos têm ROC-AUCs muito próximos, indicando uma capacidade discriminatória semelhante. No entanto, o grupo Gender_1 (True) tem uma acurácia e precisão ligeiramente menores, mas um recall maior, o que significa que o modelo é um pouco mais eficaz em identificar os "desistentes" nesse grupo, mesmo que com mais falsos positivos em comparação com o Gender_0 (False) que tem maior precisão. Observações: Para Bolsistas, o modelo tem uma acurácia geral mais alta, mas um recall consideravelmente mais baixo para a classe positiva (Dropout). Isso sugere que ele pode estar perdendo muitos "desistente

Conclusão

O modelo baseline apresenta desempenho preditivo elevado, mas evidencia disparidades relevantes de fairness, especialmente em termos de paridade demográfica para gênero e estado civil. As estratégias propostas oferecem caminhos viáveis para mitigar esses vieses, reforçando a importância de equilibrar desempenho e equidade em aplicações educacionais de aprendizado de máquina.