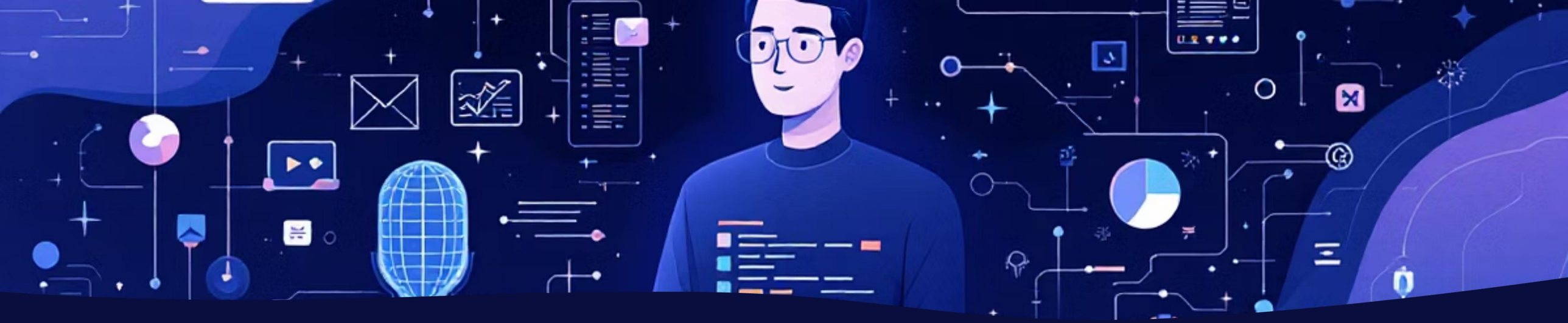
Etapa 2: Pré-processamento e Feature Engineering

A segunda etapa foca em limpar, padronizar e transformar os dados para garantir a qualidade e o desempenho ideal dos modelos de Machine Learning. É a base para uma análise robusta e resultados preditivos precisos.



Preparando o Conjunto de Dados para Modelagem Preditiva

Qualidade dos Dados

Garantir que os dados estejam limpos e sem inconsistências.

Otimização de Desempenho

Melhorar a performance dos modelos de Machine Learning.

Consistência

Padronizar formatos para evitar erros de interpretação.

2.1 - Tratamento de Categorias (Limpeza)

Objetivo

Padronizar entradas de texto para evitar múltiplas categorias para o mesmo valor, como 'Good' vs. ' good '.

Ação

Aplicamos limpeza de strings (remoção de espaços, conversão para minúsculas/maiúsculas) nas colunas categóricas como `gender`, `parental_education` e `family_income`.

Resultado

Consistência nos dados categóricos para o encoding posterior, facilitando a interpretação pelos modelos.

2.2 - Tratamento de Outliers

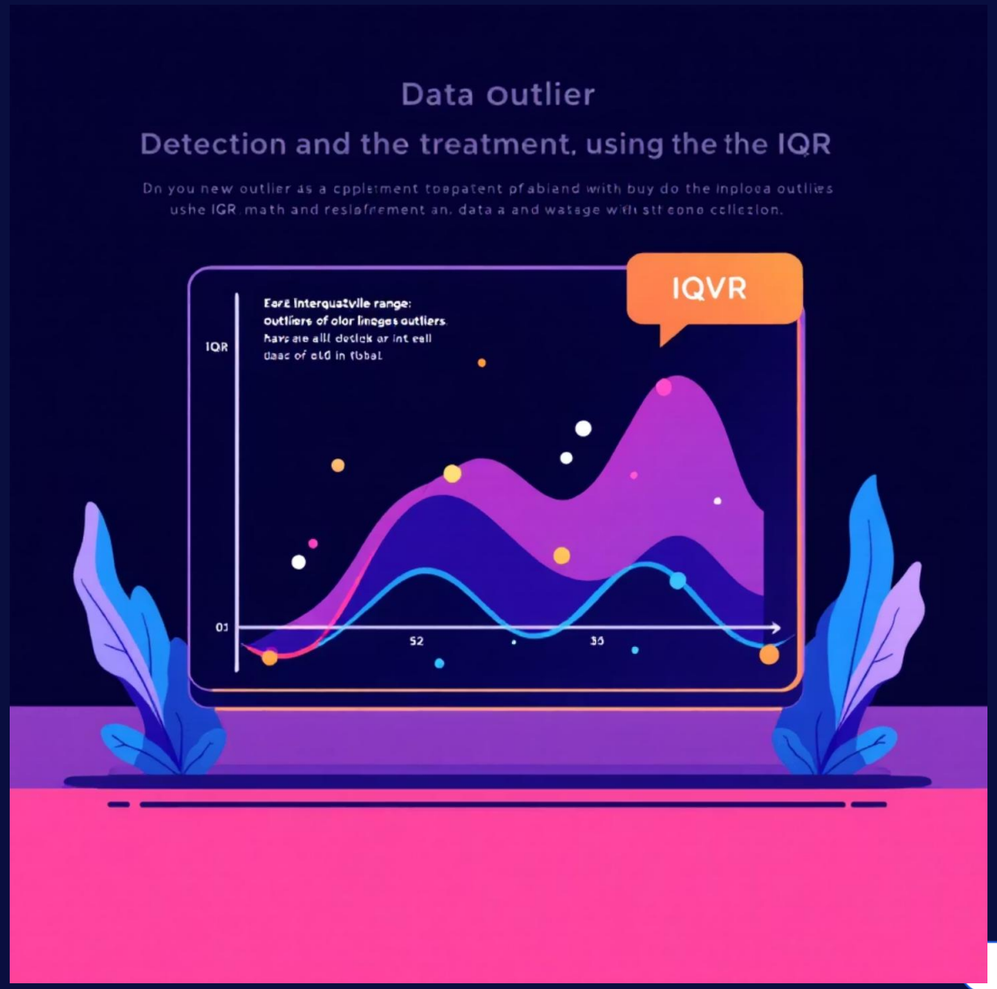
> **Mitigando o Impacto de Valores Atípicos**
>
> O tratamento de outliers é crucial para evitar que valores extremos distorçam a análise e o treinamento do modelo.

Método: Intervalo Interquartil (IQR)

Utilizamos o método IQR para identificar e limitar outliers em variáveis numéricas, garantindo que os valores extremos não prejudiquem a robustez do modelo.

Colunas Afetadas

- `study_hours_week`
- `attendance_rate`
- `sleep_hours`
- `previous_scores`



2.3 & 2.4 - Definição de Features e Pipelines

1

Features Numéricas

Imputação pela Mediana e StandardScaler.

2

Features Ordinais

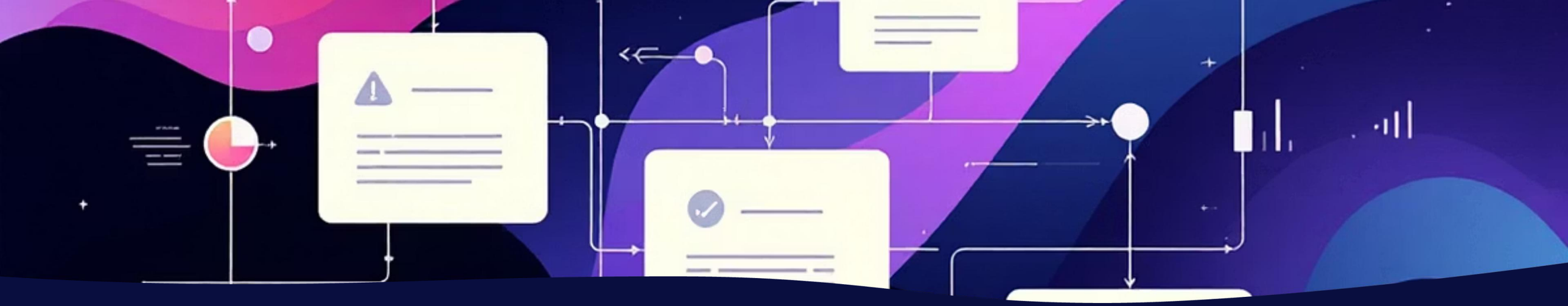
Imputação pela Moda e Ordinal Encoding.

3

Features Nominais

Imputação pela Moda e One-Hot Encoding.

As colunas foram categorizadas para aplicar pipelines de pré-processamento específicos, garantindo que cada tipo de dado receba o tratamento adequado.



Pipelines Detalhados

Pipeline Numérico

- Imputação de dados faltantes pela **Mediana**.
- **Padronização (StandardScaler)** para normalizar a escala.

Pipeline Ordinal (`parental_education`)

- Imputação pela **Moda**.
- **Ordinal Encoding** com ordem predefinida para variáveis categóricas ordinais.

Pipeline Nominal

- Imputação pela **Moda**.
- **One-Hot Encoding** para variáveis categóricas sem ordem (ex.: `gender`, `internet_quality`).

2.5 - Aplicação do Pré-processador (ColumnTransformer)

ColumnTransformer

Todos os pipelines (numérico, ordinal e nominal) foram combinados em um único **ColumnTransformer**, otimizando o processo de pré-processamento.

Execução

O pré-processador foi aplicado ao DataFrame original (`df`), resultando em um DataFrame transformado e pronto para a modelagem.

Dimensão Final do Conjunto de Dados

2.510

Linhas

Após a transformação e encoding.

63

Colunas

Após a transformação e encoding.

O conjunto de dados, antes e depois de todo o pré-processamento, está agora estruturado de forma ideal para o treinamento do modelo de Machine Learning.

2.6 - Tratamento de Assimetria (Skewness)

Objetivo

Aproximar a distribuição de variáveis numéricas de uma distribuição normal, melhorando a performance de alguns modelos.

Ação

Aplicada a transformação logarítmica (`np.log1p`) na coluna `previous_scores`.

Essa técnica ajuda a reduzir a assimetria e a lidar com a variância nos dados, tornando-os mais adequados para a análise.

Visualização da Redução da Assimetria

Antes da Transformação

Histograma mostrando a distribuição original da coluna `previous_scores`, com sua assimetria.

Após a Transformação Logarítmica

Histograma comparativo demonstrando a redução da assimetria (Skewness) após a aplicação de `np.log1p`.