

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA
Pós-graduação *Lato Sensu* em Ciência de Dados e Big Data

André Vinícius Silva Cavalcante

**APLICAÇÃO DE MODELOS DE MACHINE LEARNING NA ANÁLISE DA
INFLUÊNCIA DAS TRANSFERÊNCIAS CONSTITUCIONAIS NO IDHM: UMA
PERSPECTIVA SOCIOECONÔMICA.**

Belo Horizonte

2023

André Vinícius Silva Cavalcante

**APLICAÇÃO DE MODELOS DE MACHINE LEARNING NA ANÁLISE DA
INFLUÊNCIA DAS TRANSFERÊNCIAS CONSTITUCIONAIS NO IDHM.**

Trabalho de Conclusão de Curso apresentado
ao Curso de Especialização em Ciência de
Dados e Big Data como requisito parcial à
obtenção do título de especialista.

Belo Horizonte

2023

SUMÁRIO

1. Introdução.....	4
1.1. Contextualização	4
1.1. O problema proposto	4
2. Coleta de Dados	4
3. Processamento/Tratamento de Dados	5
4. Análise e Exploração dos Dados	5
5. Criação de Modelos de Machine Learning	5
6. Apresentação dos Resultados	5
7. Links	6
REFERÊNCIAS.....	7

1. Introdução

1.1. Contextualização

A era da informação, marcada pela avalanche de dados disponíveis, abriu um vasto campo de possibilidades para a análise e interpretação desses dados em diversos setores. No contexto da gestão pública e políticas sociais, a capacidade de analisar e interpretar dados de maneira eficiente pode resultar em insights que direcionam a formulação de políticas mais eficazes e alinhadas às necessidades da população.

As transferências constitucionais são mecanismos financeiros essenciais em muitos países, incluindo o Brasil. Elas se referem à redistribuição de recursos arrecadados pelo governo federal ou estadual para os municípios, de acordo com critérios estabelecidos na Constituição ou legislação pertinente. Essas transferências têm o objetivo de promover equilíbrio socioeconômico entre as regiões, financiar serviços públicos essenciais e garantir que municípios com menor capacidade de arrecadação possam oferecer condições mínimas de vida para sua população.

A composição das transferências constitucionais possui os seguintes fundos e repasses:

Fundo de Participação dos Municípios (FPM): Este fundo recebe 22,5% da arrecadação do IR e do IPI. A distribuição dos recursos leva em consideração o número de habitantes de cada município e o coeficiente individual de participação.

Fundo de Manutenção e Desenvolvimento da Educação Básica e de Valorização dos Profissionais da Educação (FUNDEB): Este é um fundo composto por diversos impostos e transferências dos estados, Distrito Federal e municípios, que é destinado ao financiamento da educação básica. A União ainda complementa os recursos do FUNDEB quando, no âmbito de cada estado, seu valor por aluno não alcança um mínimo definido nacionalmente.

Cota-parte dos Estados e Municípios na arrecadação do ITR (Imposto sobre a Propriedade Territorial Rural): Parte do que é arrecadado com o ITR é repassada aos municípios onde os imóveis rurais estão situados.

Royalties: são valores pagos por alguém que usa uma criação de propriedade de outro. No contexto de recursos naturais no Brasil, royalties referem-se a compensações financeiras pagas à União, estados, municípios e outros entes pela exploração de recursos minerais (como petróleo, gás natural, água e mineração). Os royalties buscam compensar os impactos sociais e ambientais da exploração desses recursos e também asseguram uma parcela da riqueza gerada para a comunidade local e a nação.

Cessão Onerosa: a cessão onerosa está relacionada à exploração de petróleo e gás no Brasil. Em 2010, o governo brasileiro e a Petrobras assinaram o contrato de cessão onerosa, permitindo à estatal explorar, sem licitação, até 5 bilhões de barris de petróleo em campos do pré-sal.

Em troca, a Petrobras pagou ao governo um valor fixo. No entanto, posteriormente, descobriu-se que as reservas nesses campos eram maiores do que o inicialmente previsto. Assim, o excedente dessa produção, ou seja, o volume de petróleo que excede esses 5 bilhões de barris inicialmente estipulados, passou a ser um tema de negociação entre o governo e a Petrobras.

Posteriormente, o governo realizou leilões desses volumes excedentes da cessão onerosa, atraindo outros investidores para a exploração desse petróleo adicional.

AFM (Auxílio Financeiro aos Municípios) e AFE (Auxílio Financeiro aos Estados): são mecanismos de transferência de recursos da União para Estados e Municípios no Brasil. Eles são exemplos de transferências constitucionais ou legais que visam a auxiliar as administrações estaduais e municipais em situações específicas.

Em diversas ocasiões, a União estabelece transferências excepcionais, temporárias ou pontuais, por meio de Medidas Provisórias ou Leis, a fim de garantir o equilíbrio fiscal e financeiro das administrações subnacionais ou para auxiliá-las diante de situações adversas, como crises financeiras, catástrofes naturais ou pandemias.

O AFM e o AFE, nesse sentido, são repasses que podem ser realizados pelo governo federal a título de auxílio, ou seja, não são transferências permanentes ou

regulares, como o FPE (Fundo de Participação dos Estados) ou o FPM (Fundo de Participação dos Municípios), que são mecanismos contínuos e previstos na Constituição Federal para redistribuição da receita federal.

Vale sempre lembrar que os critérios, os valores e as motivações para esses auxílios (AFM e AFE) podem variar conforme as decisões políticas e econômicas do momento em que são estabelecidos.

LC 173/2020, ou Lei Complementar nº 173, de 27 de maio de 2020: estabeleceu o Programa Federativo de Enfrentamento ao Coronavírus SARS-CoV-2 (Covid-19), no Brasil. Esta lei tem por objetivo trazer medidas financeiras e fiscais para auxiliar a federação, estados, municípios e o Distrito Federal no enfrentamento da pandemia.

Estas transferências são de fundamental importância para muitos municípios e estados que, muitas vezes, dependem desses recursos para financiar suas atividades e prestar serviços à população. A distribuição desses recursos é regulamentada por leis e tem como base os princípios da federação e da equidade fiscal e social.

O Índice de Desenvolvimento Humano Municipal (IDHM), por outro lado, é uma métrica que reflete a qualidade de vida e o desenvolvimento humano em nível local, considerando dimensões como longevidade, educação e renda. Seu valor é um indicativo da eficácia das políticas públicas em promover bem-estar e desenvolvimento para a população.

Dentro deste cenário, surge a questão: como as transferências constitucionais influenciam o IDHM?

A Ciência de Dados, e em particular o uso de Machine Learning, proporciona ferramentas robustas para abordar essa questão. Modelos de Machine Learning são capazes de analisar grandes volumes de dados, identificar padrões complexos e fornecer insights que métodos estatísticos tradicionais poderiam não capturar. Ao aplicar esses modelos para analisar a relação entre transferências e IDHM, é possível obter uma compreensão mais profunda e nuanceada de como diferentes fatores socioeconômicos mediam essa relação.

Deste modo, o estudo intitulado "Aplicação de Modelos de Machine Learning na Análise da Influência das Transferências Constitucionais no IDHM" visa compreender os mecanismos e critérios por trás da distribuição dos repasses financeiros constitucionais e avaliar se a forma e a magnitude dessa distribuição têm impacto direto no Índice de Desenvolvimento Humano Municipal (IDHM). Através do uso de técnicas avançadas de Machine Learning, é possível discernir padrões e correlações que podem não ser imediatamente óbvios em análises tradicionais. O resultado dessa investigação tem o potencial de revelar áreas que necessitam de atenção prioritária, facilitar uma distribuição mais eficaz dos recursos financeiros e proporcionar insights valiosos para a formulação e implementação de políticas públicas mais assertivas e alinhadas com as necessidades reais das regiões. Além disso, a utilização desses modelos de aprendizado de máquina pode servir como um instrumento de transparência, mostrando à sociedade como os recursos públicos são efetivamente utilizados e seu impacto no desenvolvimento humano.

1.2. O problema proposto:

Título:

"Como os montantes das transferências constitucionais afetam o IDHM?"

Descrição:

As transferências constitucionais têm como principal objetivo redistribuir recursos financeiros para garantir uma maior equidade entre os estados, permitindo que todos possam fornecer serviços básicos à sua população e promover o desenvolvimento. Embora a relação entre transferências constitucionais e o desenvolvimento humano seja intuitivamente direta, a magnitude e especificidade dessa relação podem variar de acordo com o contexto socioeconômico de cada município.

Problema Central:

Com a aplicação de modelos de Machine Learning, busca-se responder à seguinte questão:

Como as transferências constitucionais, em termos de montantes, influenciam diretamente o IDHM de unidades da federação com distintos perfis socioeconômicos?

Justificativa:

Entender esses mecanismos e variáveis mediadoras é crucial para formular políticas públicas mais efetivas. Se determinados perfis socioeconômicos de unidades da federação respondem de maneira mais significativa às transferências, isso pode direcionar estratégias de alocação de recursos.

Em suma, este problema visa otimizar a aplicação de recursos públicos para promover o desenvolvimento humano, garantindo que esses recursos sejam utilizados da forma mais eficiente possível, considerando as nuances e particularidades de cada contexto municipal.

2. Coleta de Dados

A pesquisa focou nos anos de 2019 e 2020. A escolha de 2019 foi motivada por representar um cenário socioeconômico anterior à pandemia de Covid-19, enquanto 2020 reflete os impactos dessa pandemia. Além da relevância de analisar o antes e depois da crise sanitária, a seleção desses anos também foi influenciada por sua recenticidade, proporcionando um olhar sobre um contexto socioeconômico contemporâneo. Adicionalmente, os conjuntos de dados utilizados para ambos os anos têm a mesma faixa temporal, garantindo uma comparação consistente entre eles.

Os dados utilizados nesta pesquisa foram retirados dos sites do Tesouro Nacional Transparente, site que faz parte da estrutura de sites do governo brasileiro, disponível em <https://www.tesourotransparente.gov.br/ckan/dataset/transferencias-constitucionais-para-municipios> e do Atlas do Desenvolvimento Humano no Brasil disponível em <http://www.atlasbrasil.org.br/consulta/planilha> em 15 de outubro de 2023. Os arquivos oriundos do site do Tesouro Transparente foram baixados manualmente em formato .csv e o arquivo proveniente do site Atlas do Desenvolvimento Humano no Brasil foi baixado em formato .xlsx.

Os dados brutos referentes às transferências constitucionais para os municípios foram organizados mensalmente e armazenados em duas pastas distintas, correspondentes aos anos de 2019 e 2020. Paralelamente, os dados brutos do IDHM por unidade federativa foram consolidados em um único arquivo, sendo alocados na pasta "IDHM". Durante a etapa de coleta, observou-se que as informações sobre transferências constitucionais estavam detalhadas tanto por unidade federativa quanto por município. Contudo, a versão mais recente do IDHM detalhado por município data de 2010, baseada nos dados do último censo realizado pelo IBGE. As informações do censo mais recente (2022-2023) ainda não foram publicadas. Dado esse cenário, optou-se por usar os dados da PNAD Contínua de 2019 e 2020. No entanto, é importante destacar que esses dados da PNAD Contínua não oferecem detalhamento por município.

3. Processamento/Tratamento de Dados

O processo de ETL (Extração, Transformação e Carga) é uma metodologia empregada para capturar informações de múltiplas fontes, formatá-las para uso analítico e, em seguida, depositá-las em sistemas de armazenamento, como data warehouses. Essa abordagem é fundamental para a área de Business Intelligence e para o desenvolvimento de sistemas que necessitam integrar dados oriundos de diversas origens. Vamos entender mais profundamente cada fase:

Extração:

Durante essa etapa, dados são capturados de várias origens, sejam elas bancos de dados tradicionais, sistemas de planejamento de recursos empresariais (ERP), arquivos em diferentes formatos ou mesmo fluxos de dados em tempo real.

O objetivo é reunir todas as informações requeridas para análises subsequentes, não importando sua origem inicial.

Transformação:

Após a extração, é comum que os dados não estejam em uma estrutura ou formato propício para análise. Pode haver inconsistências, erros, registros duplicados, entre outras questões.

Essa etapa consiste em uma série de procedimentos que visam refinar, validar e adaptar os dados para um formato mais conveniente. Esse processo pode incluir agregação de informações, conversões, codificações, normalizações e outras atividades para tratamento e enriquecimento dos dados.

Carga:

Uma vez transformados, os dados são então transferidos para seu destino, geralmente um data warehouse ou algum tipo específico de banco de dados para análises.

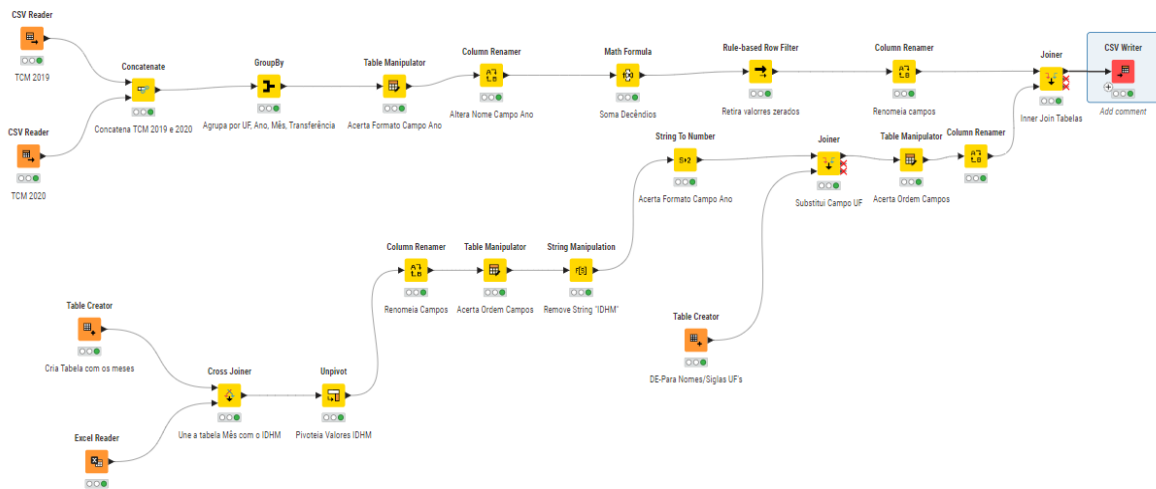
A carga pode ser executada de várias maneiras, seja através de uma carga total, onde todos os dados são inseridos simultaneamente, ou cargas incrementais, adicionando apenas registros novos ou alterados.

O ETL é essencial para assegurar que os dados em sistemas analíticos estejam sempre atuais, íntegros e preparados para serem analisados. Diante da crescente quantidade de dados e a demanda por processamentos ágeis, o conceito tradicional de ETL tem se adaptado, resultando em novas abordagens como ELT (Extração, Carga e Transformação) e o surgimento de ferramentas especializadas que otimizam essas tarefas.

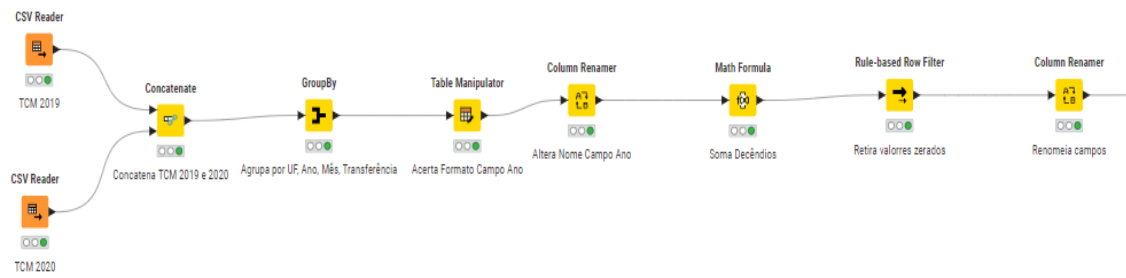
O processo de ETL foi feito totalmente pelo software Knime (versão 5.1.1 – build September 14, 2023).



O processo de ETL em sua totalidade está evidenciado na figura abaixo



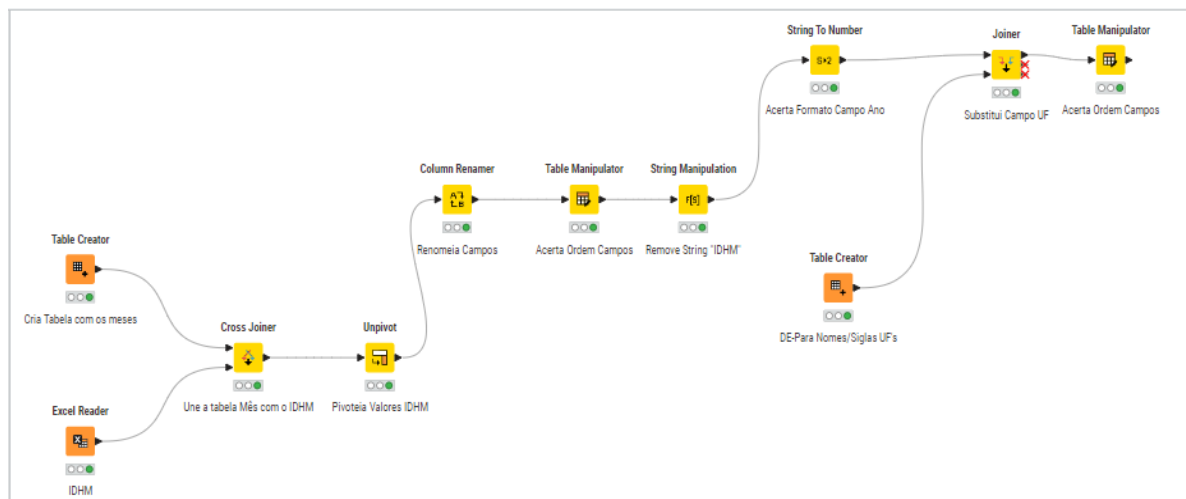
O processo de extração dos dados começou com o tratamento dos arquivos que continham as informações das transferências constitucionais para os municípios referentes aos anos de 2019 e 2020. Para extrair os dados, foi utilizado o nó “CSV Reader”. O processo de transformação iniciou-se com a concatenação dos dados dos dois arquivos, usando o nó “Concatenate”. Em seguida, foi feito o agrupamento dos três campos de medidas: “1º Decêndio”, “2º Decêndio” e “3º Decêndio” pelas dimensões UF, Ano, Mês e Transferência, e para isto, foi utilizado o nó “Group By”. O campo “ANO” precisou ter seu tipo alterado de String para Double e o nome do campo foi alterado de “ANO” para “Ano”, usando-se os nós “Table Manipulator” e “Column Renamer” respectivamente. Estes ajustes foram necessários para que fosse possível a união dos dados das transferências com os dados do IDHM. Para finalizar o processo de transformação, o nó “Math Formula” foi utilizado para somar os valores dos campos “1º Decêndio”, “2º Decêndio” e “3º Decêndio” e criar o campo “Total Repasses” com o resultado da soma. O nó “Rule-Based Row Filter” foi utilizado para excluir todos os valores zerados do campo “Total Repasses” e o nó “Column Renamer” foi usado para ajustar todos os nomes de campos de forma a adequá-los ao uso com o nó “Joiner”.



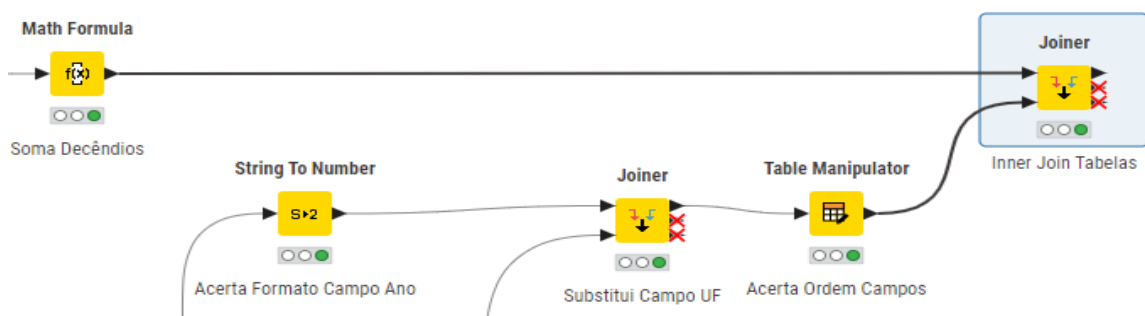
Para o processo de extração dos dados do IDHM, foi utilizado o nó “Excel Reader”. O processo de transformação iniciou-se com a adição da coluna “Mês” no arquivo, uma vez que os dados contidos neste arquivo refletem o índice de forma anual, e para harmonizá-lo com os dados das transferências constitucionais, fez-se necessário transformar o dado em indicador mensal inserindo a coluna de dimensão “Mês”. Esta coluna, evidentemente distorcerá o resultado final do dado, uma vez que está tornando um dado que é originalmente anual em um dado mensal. Mas tomou-se por padrão que este número refletiria uma média do índice e que esta alteração não seria suficientemente importante para inserir uma grande influência nos dados da pesquisa.

Para tanto, utilizou-se o nó “Table Creator” para gerar uma tabela com os meses de 1 a 12 e em seguida, utilizou-se o nó “Cross Joiner” que gerou um produto cartesiano dos meses em relação a cada linha de registro da tabela “IDHM”, inserindo a coluna “Mês” na tabela e gerando uma linha para cada registro. Em seguida, o nó “Unpivot” foi usado para transpor as colunas “IDMH 2019” e “IDHM 2020” para linhas. A transposição das colunas para linhas gerou duas novas colunas chamadas “Column Name” e “Column Values” com as descrições dos anos e índices do IDHM respectivamente. Estas novas colunas foram renomeadas com os nomes “Ano” e “IDHM”. A coluna “Territorialidades” foi renomeada para “UF” e para esta tarefa foi utilizado o nó “Column Renamer”. O próximo passo foi utilizar o nó “Table Manipulator” para ajustar a ordem das colunas. A coluna “Column Name” ao ser transposta trouxe a palavra “IDHM” junto com a informação do ano e esta coluna precisava ser alterada para conter apenas a informação do ano. Esta transformação foi feita utilizando-se o nó “String Manipulation” e em seguida, o campo ano que estava com o formato string foi convertido para o formato numérico pelo nó “String To Number”.

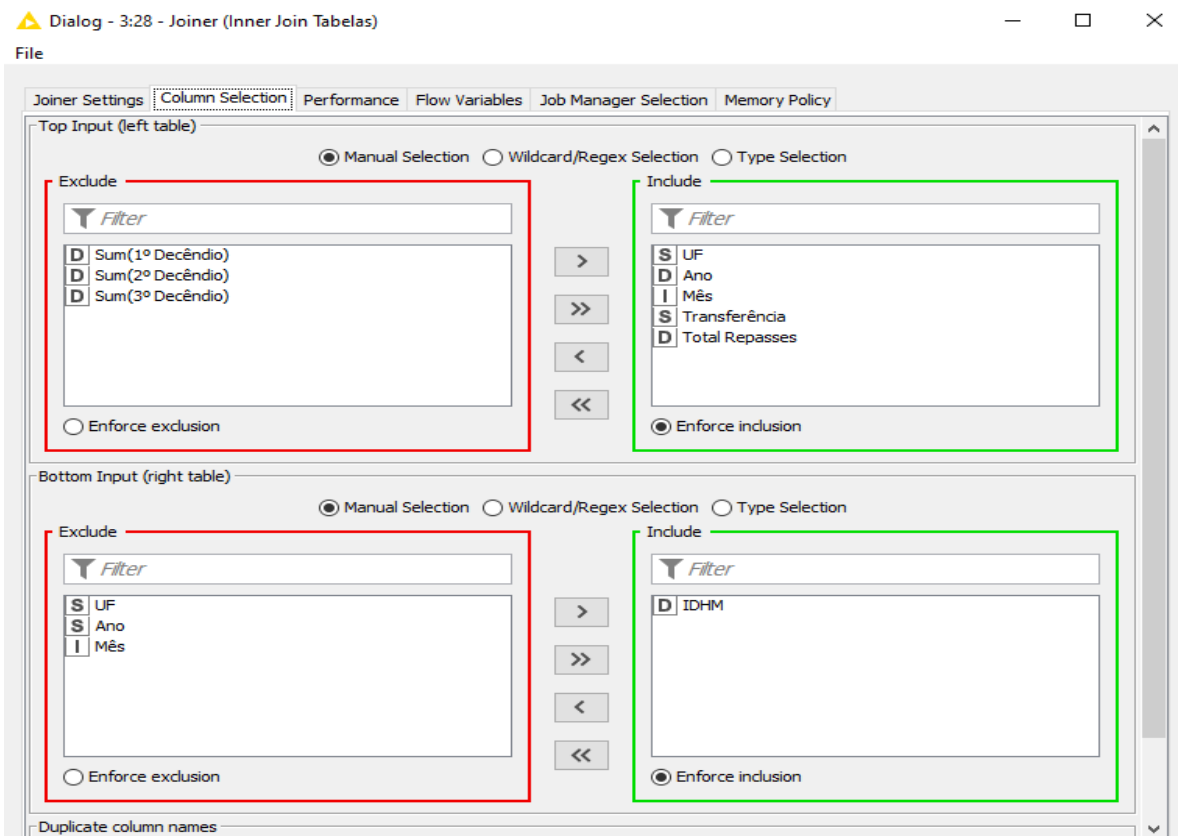
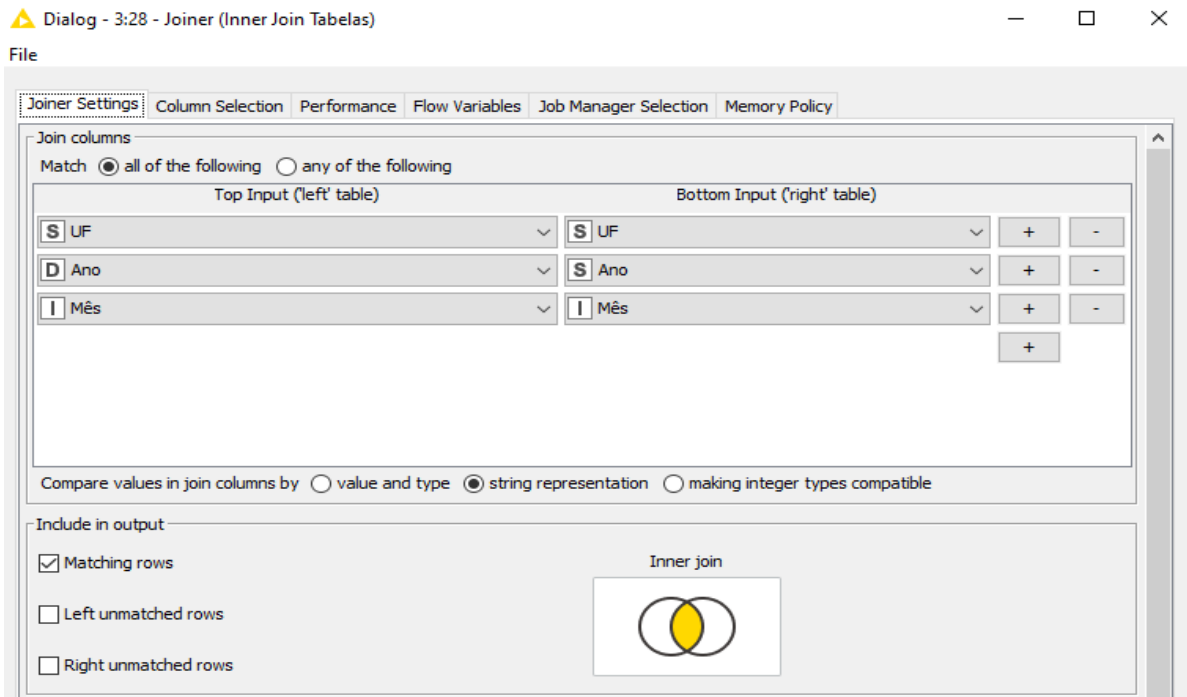
A tabela de dados do IDHM precisou de mais um ajuste no campo UF, visto que este continha o nome completo da UF e para fazer a junção com a tabela de transferências constitucionais, foi necessário alterar o nome completo da UF para a sigla da UF e para esta tarefa, o nó “Table Creator” foi mais uma vez utilizado para criar uma tabela com duas colunas contendo o nome completo da UF e uma coluna contendo a sigla da UF, gerando assim uma tabela De-Para. Esta tabela com o De-Para das UF’s foi unida com a tabela com os dados do IDHM pelo nó “Joiner”, fazendo ao mesmo tempo a substituição da coluna com a descrição da UF pela coluna com a sigla da UF. Para finalizar, mais uma vez foi necessária a ordenação das colunas da tabela de forma a torná-la com a mesma sequência de colunas da tabela de transferências constitucionais.



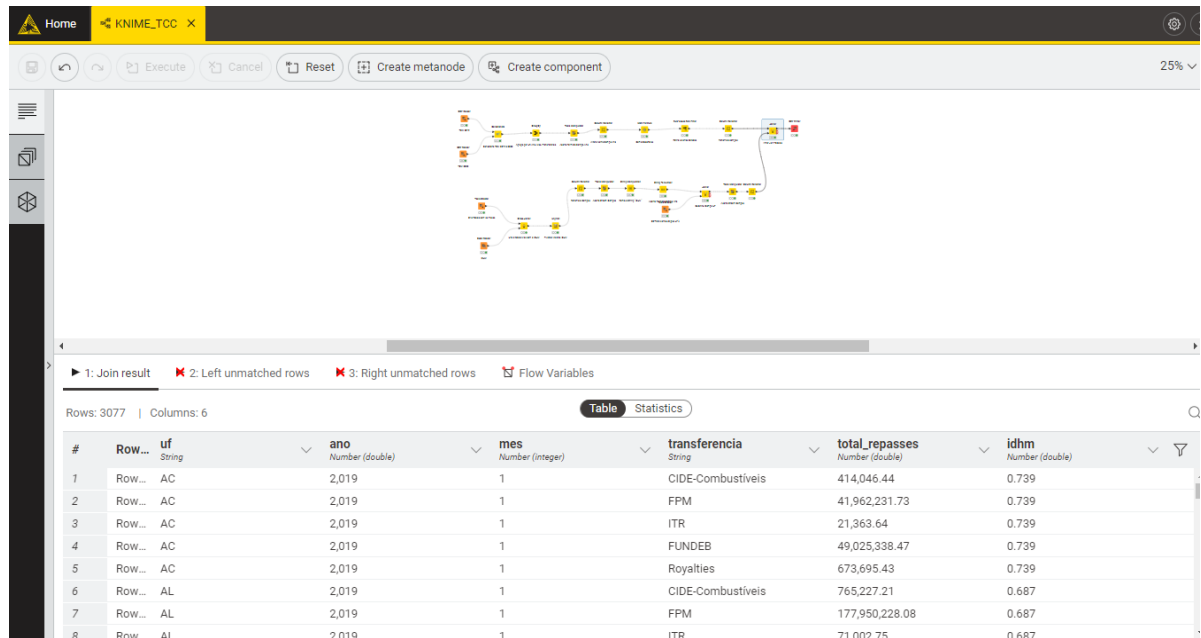
Para finalizar o processo de transformação, utilizou-se o nó “Joiner” para unir as tabelas tratadas das transferências constitucionais com a tabela do IDHM utilizando-se para isto uma união do tipo “Inner Join



Um "Inner Join" retorna apenas as linhas que possuem uma correspondência em ambas as tabelas envolvidas no Join. As figuras abaixo demonstram a configuração utilizada no nó "Joiner" para obter o resultado desejado:



O resultado final do processo de extração e transformação apresentado no nó “Joiner” foi o apresentado na figura abaixo:



The screenshot shows the KNIME software interface. At the top, there's a menu bar with 'Home' and 'KNIME_TCC'. Below it, a toolbar contains icons for 'Execute', 'Cancel', 'Reset', 'Create metanode', and 'Create component'. The main workspace displays a complex data workflow with various nodes connected by lines. Below the workspace, a status bar indicates '1: Join result', '2: Left unmatched rows', '3: Right unmatched rows', and 'Flow Variables'. A table view is open, showing 3077 rows and 6 columns. The table has columns: #, Row..., uf, ano, mes, transferencia, total_repasses, and idhm. The data is organized into 8 rows, each representing a different state or entity.

#	Row...	uf	ano	mes	transferencia	total_repasses	idhm
1	Row...	AC	2,019	1	CIDE-Combustíveis	414,046.44	0.739
2	Row...	AC	2,019	1	FPM	41,962,231.73	0.739
3	Row...	AC	2,019	1	ITR	21,363.64	0.739
4	Row...	AC	2,019	1	FUNDEB	49,025,338.47	0.739
5	Row...	AC	2,019	1	Royalties	673,695.43	0.739
6	Row...	AL	2,019	1	CIDE-Combustíveis	765,227.21	0.687
7	Row...	AL	2,019	1	FPM	177,950,228.08	0.687
8	Row...	AL	2,019	1	ITR	71,002.75	0.687

Para o processo de gravação dos dados transformados pelo Knime foi utilizado o nó “CSV Reader”.

Ao final do processo de ETL, o arquivo .csv resultante gerou o dataset abaixo com um total de 3.195 linhas:

Nome da coluna/campo	Descrição	Tipo
uf	Unidade da Federação que recebeu as transferências e do IDHM	Texto
ano	Ano das transferências e do IDHM	Número
mes	Mês das transferências e do IDHM	Número
transferencia	Tipo da transferência	Texto
total_repasses	Valor total das transferências	Número
idhm	Valor do IDHM	Número

4. Análise e Exploração dos Dados

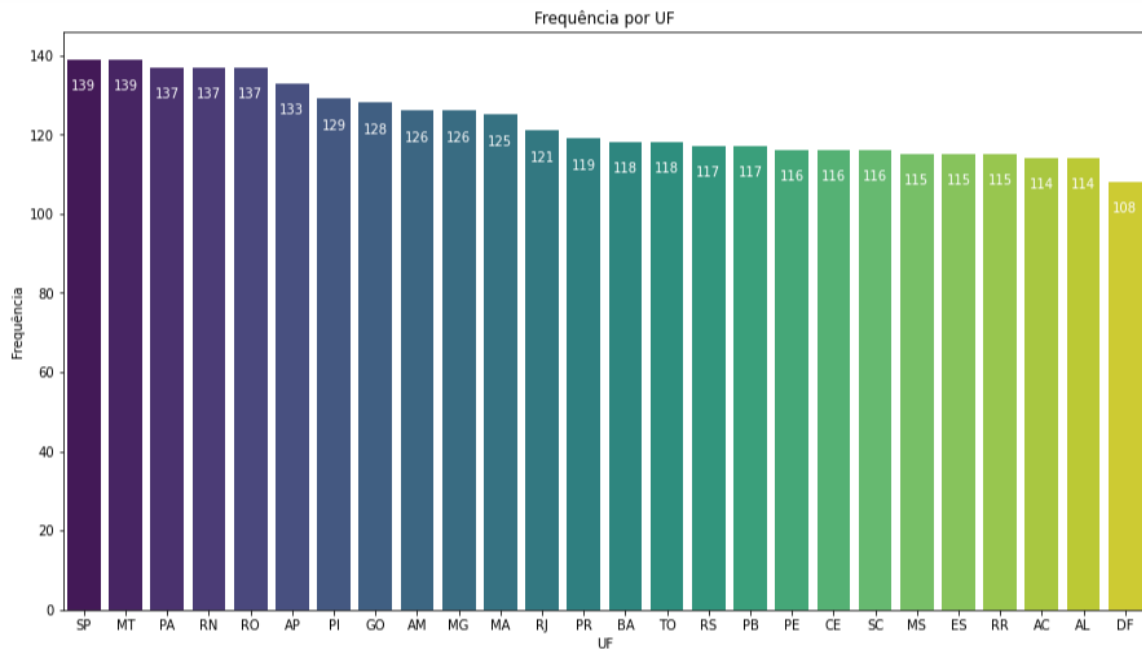
A análise iniciou com um resumo estatístico do data frame.

```
In [1]: 1 import pandas as pd
2
3 df = pd.read_csv('C:/Users/andre/Desktop/TCC PUGMG/DATASET TCC PUGMG/2_Transformação/df_tcc_t.csv', delimiter=';')
4
5 resumo_all = df.describe(include='all')
6 print(resumo_all)
7
```

	uf	ano	mes	transferencia	total_repasses	idhm
count	3195	3195.000000	3195.000000	3195	3195	3195
unique	26	NaN	NaN	10	3146	45
top	MT	NaN	NaN	FUNDEB	38653051	0,739
freq	139	NaN	NaN	624	4	255
mean	NaN	2019.560876	6.533333	NaN	NaN	NaN
std	NaN	0.496358	3.406925	NaN	NaN	NaN
min	NaN	2019.000000	1.000000	NaN	NaN	NaN
25%	NaN	2019.000000	4.000000	NaN	NaN	NaN
50%	NaN	2020.000000	7.000000	NaN	NaN	NaN
75%	NaN	2020.000000	9.000000	NaN	NaN	NaN
max	NaN	2020.000000	12.000000	NaN	NaN	NaN

O histograma das frequências por UF no data frame mostra quais as UF's mais se repetiram no conjunto de dados analisado. Para gerar o histograma, o código abaixo foi utilizado:

```
In [4]: 1 import seaborn as sns
2 import pandas as pd
3 import matplotlib.pyplot as plt
4
5 # Carregando o DataFrame
6 df = pd.read_csv('C:/Users/andre/Desktop/TCC PUGMG/DATASET TCC PUGMG/2_Transformação/df_tcc_t.csv', delimiter=';')
7
8 # Calculando as top 5 'uf' mais frequentes
9 top_5_uf = df['uf'].value_counts().head(30)
10
11 # Define o tamanho da figura
12 plt.figure(figsize=(15, 8))
13
14
15
16 # Plotando o gráfico de barras das top 5 'uf'
17 ax = sns.barplot(x=top_5_uf.index, y=top_5_uf.values, palette='viridis')
18 plt.title('Frequência por UF')
19 plt.xlabel('UF')
20 plt.ylabel('Frequência')
21
22 # Adicionando os valores de frequência dentro de cada barra
23 for p in ax.patches:
24     ax.annotate(f'{int(p.get_height())}',
25               (p.get_x() + p.get_width() / 2., p.get_height()),
26               ha='center', va='center',
27               xytext=(0, -20),
28               textcoords='offset points',
29               color='white')
30
31 plt.show()
32
```

Em seguida, foram avaliados os valores mínimo e máximo das transferências constitucionais e o percentual entre eles:

```
In [11]: 1 import pandas as pd
2
3 df = pd.read_csv('C:/Users/andre/Desktop/TCC PUGMG/DATASET TCC PUGMG/2_Transformação/df_tcc_t.csv', delimiter=';')
4
5 df['total_repasses'] = df['total_repasses'].str.replace('.', '').str.replace(',', '.').astype(float)
6
7 # Garantindo que mínimo e máximo são floats
8 minimo = df['total_repasses'].min()
9 maximo = df['total_repasses'].max()
10
11 minimo_formatado = f"{minimo:.2f}".replace(".", "x").replace(",", ".").replace("x", ",")
12 maximo_formatado = f"{maximo:.2f}".replace(".", "x").replace(",", ".").replace("x", ",")
13 perc = minimo / maximo * 100
14
15 print(f"Valor mínimo é R$: {minimo_formatado}")
16 print(f"Valor máximo é R$: {maximo_formatado}")
17 print(f"O valor mínimo é {perc:.9f}% do valor máximo.")
```

Valor mínimo é R\$: 1,43
 Valor máximo é R\$: 2.591.459.222,00
 O valor mínimo é 0.000000055% do valor máximo.

Foi utilizado o código abaixo para avaliar o valor médio dos repasses constitucionais:

```
In [1]: 1 import pandas as pd
2
3 df = pd.read_csv('C:/Users/andre/Desktop/TCC PUGMG/DATASET TCC PUGMG/2_Transformação/df_tcc_t.csv', delimiter=';')
4
5 df['total_repasses'] = df['total_repasses'].str.replace('.', '').str.replace(',', '.').astype(float)
6
7 media = df['total_repasses'].mean()
8 media_formatada = f"{media:.2f}".replace(".", "x").replace(",", ".").replace("x", ",")
9 print(f"O valor médio dos repasses é de R$: {media_formatada}")
10
```

O valor médio dos repasses é de R\$: 146.293.647,20

A avaliação da UF com maior valor de repasses recebido foi São Paulo

```
In [4]: 1 import pandas as pd
2 df = pd.read_csv('C:/Users/andre/Desktop/TCC PUGMG/DATASET TCC PUGMG/2_Transformação/df_tcc_t.csv', delimiter=';')
3
4 df['total_repasses'] = df['total_repasses'].str.replace('.', '').str.replace(',', '.').astype(float)
5
6
7
8
9 sum_by_uf = df.groupby('uf').sum()
10
11 # Encontrando o cliente com o maior valor gasto
12 maior_uf = sum_by_uf['total_repasses'].idxmax()
13 maior_valor = sum_by_uf['total_repasses'].max()
14 maior_valor_formatado = f"{maior_valor:,.2f}".replace(".", "x").replace(",", ".").replace("x", ",")
15 print(f"Estado com maior valor de repasses é {maior_uf} com um valor total de R$ {maior_valor_formatado}")
```

Estado com maior valor de repasses é SP com um valor total de R\$ 76.599.038.879,26

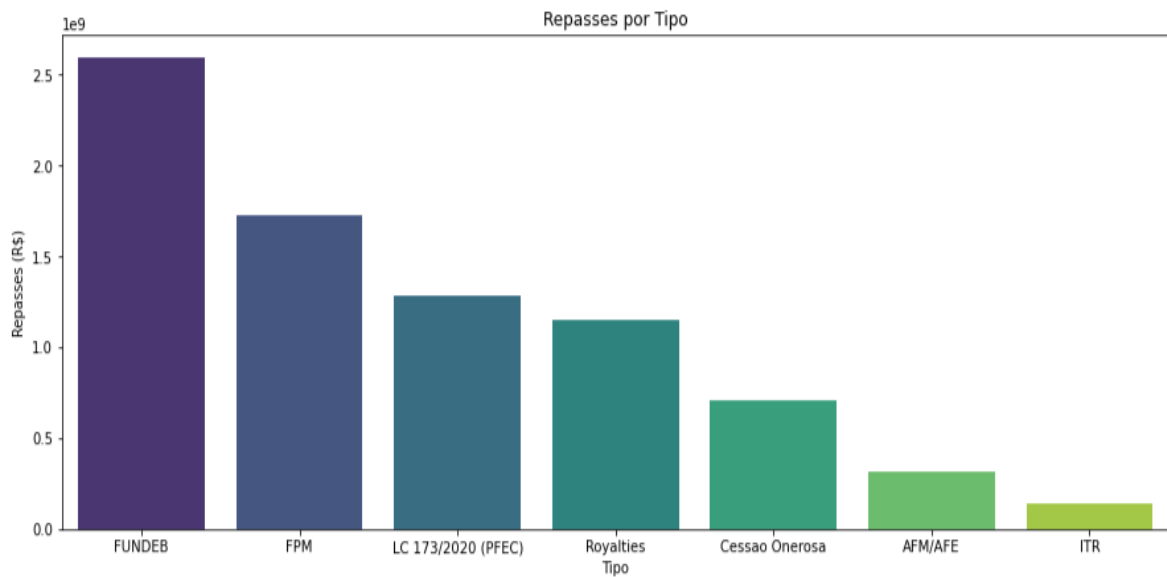
Enquanto a UF com menor valor recebido foi o Distrito Federal

```
In [4]: 1 import pandas as pd
2 df = pd.read_csv('C:/Users/andre/Desktop/TCC PUGMG/DATASET TCC PUGMG/2_Transformação/df_tcc_t.csv', delimiter=';')
3
4 df['total_repasses'] = df['total_repasses'].str.replace('.', '').str.replace(',', '.').astype(float)
5
6
7
8
9 sum_by_uf = df.groupby('uf').sum()
10
11 # Encontrando o cliente com o maior valor gasto
12 menor_uf = sum_by_uf['total_repasses'].idxmin()
13 menor_valor = sum_by_uf['total_repasses'].min()
14 menor_valor_formatado = f"{menor_valor:,.2f}".replace(".", "x").replace(",", ".").replace("x", ",")
15 print(f"Estado com menor valor de repasses é {menor_uf} com um valor total de R$ {menor_valor_formatado}")
```

Estado com menor valor de repasses é DF com um valor total de R\$ 549.893.313,96

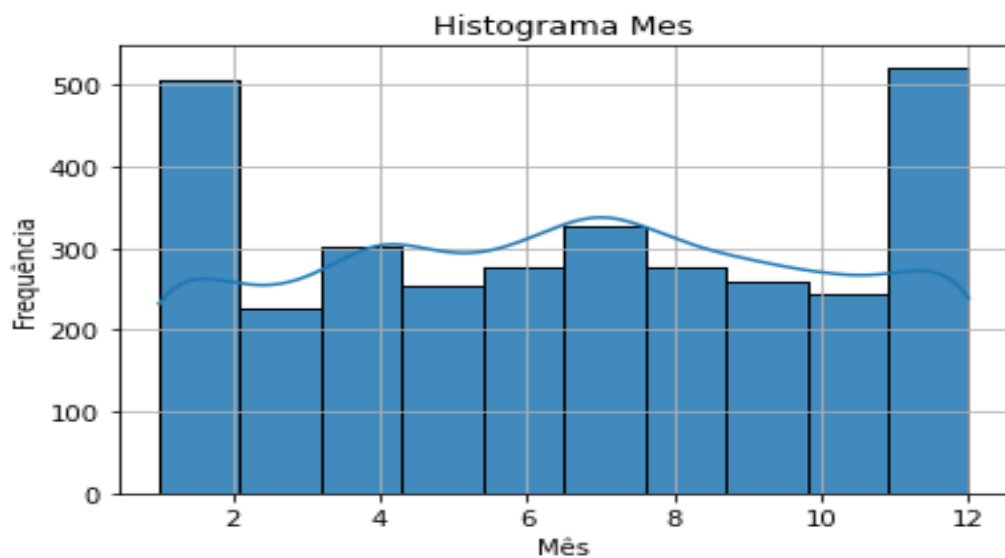
Para demonstrar o valores repassados por tipo, o código abaixo foi gerado um gráfico de barras demonstrando os dados em ordem decrescente.

```
In [5]: 1 import seaborn as sns
2 import pandas as pd
3 import matplotlib.pyplot as plt
4
5
6
7 # Carregando o DataFrame
8 df = pd.read_csv('C:/Users/andre/Desktop/TCC PUGMG/DATASET TCC PUGMG/2_Transformação/df_tcc_t.csv', delimiter=';')
9
10 df['total_repasses'] = df['total_repasses'].str.replace('.', '').str.replace(',', '.').astype(float)
11
12 #df['total_repasses'] = "{:,.2f}".format('total_repasses').replace(".", "x").replace(",", ".").replace("x", ",")
13
14
15 # Calculando as 5 UF que mais receberam repasses (ajuste o nome da coluna conforme seu DataFrame)
16 top_5_uf_idhm = df.groupby('transferencia')['total_repasses'].max().nlargest(7)
17
18
19
20 # Define o tamanho da figura
21 plt.figure(figsize=(15, 6))
22
23
24 # Plotando o gráfico de barras das 5 UF que mais receberam repasses
25 ax = sns.barplot(x=top_5_uf_idhm.index, y=top_5_uf_idhm.values, palette='viridis')
26 plt.title('Repasses por Tipo')
27 plt.xlabel('Tipo')
28 plt.ylabel('Repasses (R$)')
29
30 plt.show()
31
```



O histograma de repases por mês demonstra que os maiores valores de transferências de recursos estão concentrados nos meses de dezembro e janeiro. O código abaixo foi utilizado para gerar o gráfico.

```
In [12]: 1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 df = pd.read_csv('C:/Users/andre/Desktop/TCC PUGMG/DATASET TCC PUGMG/2_Transformação/df_tcc_t.csv', delimiter=';')
5
6 df['mes'].hist(bins=10, edgecolor='black', alpha=0.7)
7 plt.title('Histograma IDHM')
8 plt.xlabel('IDHM')
9 plt.ylabel('Frequência')
10 plt.show()
```

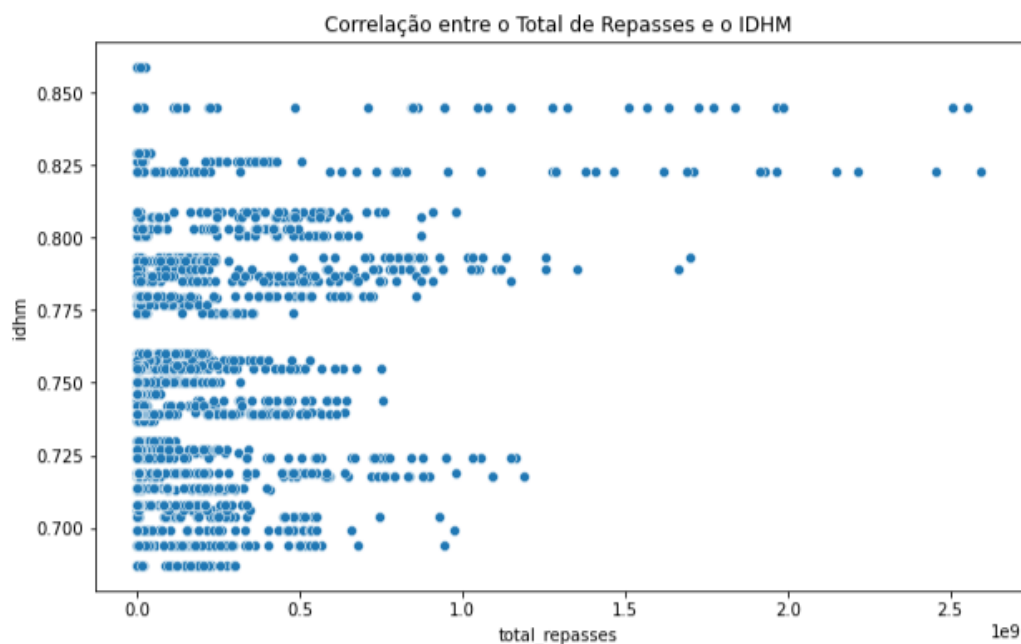


A correlação entre o total de repasses e o IDHM é baixa. Está em torno de 19%.

```
In [5]: 1 import seaborn as sns
2 import pandas as pd
3 import matplotlib.pyplot as plt
4
5 df = pd.read_csv('C:/Users/andre/Desktop/TCC PUGMG/DATASET TCC PUGMG/2_Transformação/df_tcc_t.csv', delimiter=';')
6
7 df['total_repasses'] = df['total_repasses'].str.replace('.', '').str.replace(',', '.').astype(float)
8 df['idhm'] = df['idhm'].str.replace('.', '').str.replace(',', '.').astype(float)
9
10 df['total_repasses'] = pd.to_numeric(df['total_repasses'], errors='coerce')
11
12 df['idhm'] = pd.to_numeric(df['idhm'], errors='coerce')
13
14 correlation = df['total_repasses'].corr(df['idhm'])
15 print(f'Correlação entre o Total de Repasses e o IDHM: {correlation:.10f}')
16
17 plt.show()
18
```

Correlação entre o Total de Repasses e o IDHM: 0.1897301543

```
In [4]: 1 import seaborn as sns
2 import pandas as pd
3 import matplotlib.pyplot as plt
4
5 df = pd.read_csv('C:/Users/andre/Desktop/TCC PUGMG/DATASET TCC PUGMG/2_Transformação/df_tcc_t.csv', delimiter=';')
6
7 df['total_repasses'] = df['total_repasses'].str.replace('.', '').str.replace(',', '.').astype(float)
8 df['idhm'] = df['idhm'].str.replace('.', '').str.replace(',', '.').astype(float)
9
10 df['total_repasses'] = pd.to_numeric(df['total_repasses'], errors='coerce')
11
12 df['idhm'] = pd.to_numeric(df['idhm'], errors='coerce')
13
14 correlation = df['total_repasses'].corr(df['idhm'])
15 print(f'Correlação entre o Total de Repasses e o IDHM: {correlation:.10f}')
16
17 plt.figure(figsize=(10, 6))
18 sns.scatterplot(data=df, x='total_repasses', y='idhm')
19 plt.title('Correlação entre o Total de Repasses e o IDHM')
20 plt.show()
21
```

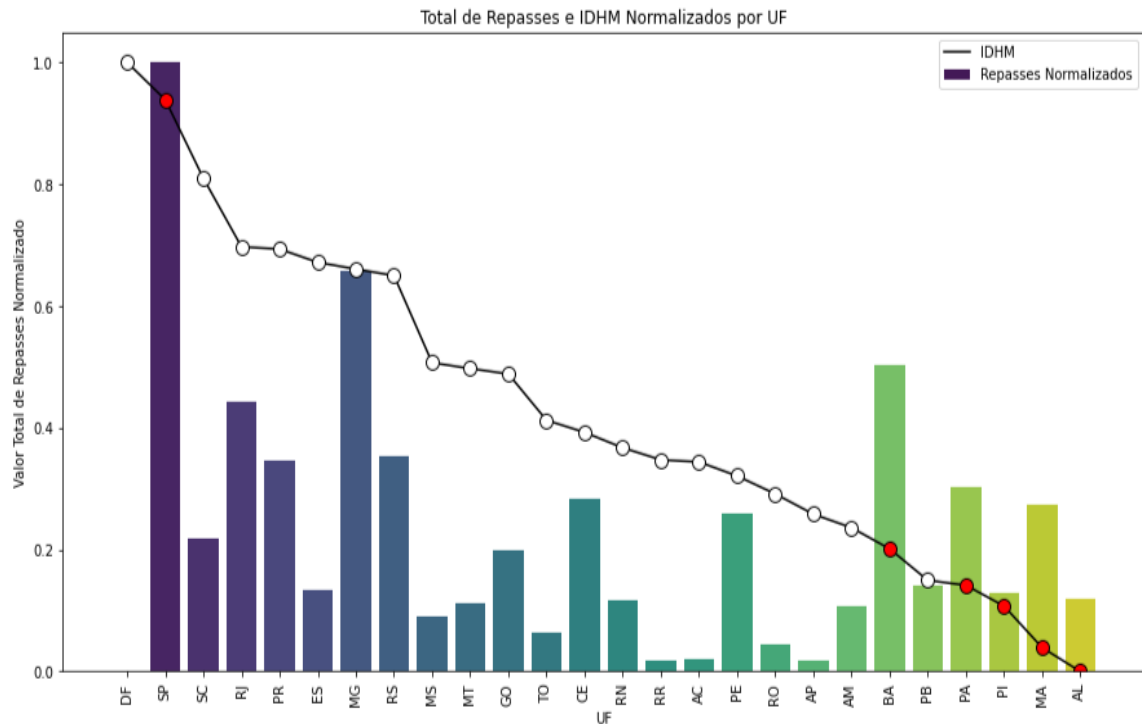


Para finalizar análise, foi traçado um gráfico que correlaciona o valor total de repasses e o IDHM por UF utilizando-se para isto, a normalização das colunas, visando colocar as duas métricas dentro de um mesmo patamar de valores, facilitando assim, a visualização dos dados. Desta forma, foram observados alguns fatos relevantes, entre eles, está a grande disparidade entre o IDHM e os repasses, isto é devido as características constitucionais que cada tipo de repasse possui.

Outro fato relevante, é o Distrito federal possuir o maior IDHM do Brasil, mas receber valores muito baixos de transferências constitucionais e isto prende-se ao fato do Distrito Federal possuir uma posição única na federação brasileira, combinando atributos de estados e municípios. Devido a essa singularidade, o regime de repasses constitucionais para o DF difere dos demais entes federativos. Para além destes fatos, uma questão foi detectada: qual seria o comportamento do IDHM do ano de 2023, caso os valores dos repasses continuassem conforme o que foi avaliado no conjunto de dados ?

Para gerar o gráfico, utilizou o código abaixo:

```
In [1]: 1 import seaborn as sns
2 import pandas as pd
3 import matplotlib.pyplot as plt
4
5 # Carregando o DataFrame
6 df = pd.read_csv('C:/Users/andre/Desktop/TCC PUGMG/DATASET TCC PUGMG/2_Transformação/df_tcc_t.csv', delimiter=';')
7
8 df['total_repasses'] = df['total_repasses'].str.replace('.', '').str.replace(',', '.').astype(float)
9 df['idhm'] = df['idhm'].str.replace('.', '').str.replace(',', '.').astype(float)
10
11 # Calculando os repasses por UF
12 repasses_by_uf = df.groupby('uf')['total_repasses'].sum()
13
14 # Calculando o IDHM por UF
15 idhm_by_uf = df.groupby('uf')['idhm'].mean()
16
17 # Organizando pela ordem decrescente do IDHM
18 ordered_ufs = idhm_by_uf.sort_values(ascending=False).index
19 repasses_by_uf = repasses_by_uf[ordered_ufs]
20 idhm_by_uf = idhm_by_uf[ordered_ufs]
21
22 # Normalizando os valores de repasses e IDHM
23 repasses_by_uf_normalized = (repasses_by_uf - repasses_by_uf.min()) / (repasses_by_uf.max() - repasses_by_uf.min())
24 idhm_by_uf_normalized = (idhm_by_uf - idhm_by_uf.min()) / (idhm_by_uf.max() - idhm_by_uf.min())
25
26 # Criando uma figura e um conjunto de subplots
27 fig, ax = plt.subplots(figsize=(15, 8))
28
29 # Plotando o gráfico de barras para os repasses normalizados
30 sns.barplot(x=repasses_by_uf_normalized.index, y=repasses_by_uf_normalized.values, palette='viridis', ax=ax, label="Repasses")
31
32 # Plotando o gráfico de linha para o IDHM normalizado
33 sns.lineplot(x=idhm_by_uf_normalized.index, y=idhm_by_uf_normalized.values, color='black', ax=ax)
34
35 # Plotando o gráfico de pontos para o IDHM normalizado
36 colors = ['white' if idhm_value > repasses_value else 'red' for idhm_value, repasses_value in zip(idhm_by_uf_normalized, repasses_by_uf_normalized)]
37 for idx, color in enumerate(colors):
38     ax.plot(idx, idhm_by_uf_normalized[idx], 'o', color=color, markersize=10, markeredgecolor='black')
39
40 plt.title('Repasses e IDHM Normalizados por UF')
41 plt.xlabel('UF')
42 plt.ylabel('Valor Normalizado')
43 plt.legend()
44 plt.xticks(rotation=90)
45
46 plt.show()
```



5. Criação de Modelos de Machine Learning

Para prever o IDHM de 2023, com base nos padrões observados no conjunto de dados, empregamos três métodos: regressão linear, regressão Ridge e Keras LSTM. A escolha por regressão linear e regressão Ridge deve-se ao fato de que são técnicas tradicionalmente indicadas para problemas de regressão. Por outro lado, embora o Keras LSTM não seja uma ferramenta tradicionalmente empregada para regressão, ele foi considerado devido à sua habilidade de modelar sequências temporais através de redes neurais, o que pode ser útil para prever tendências futuras do IDHM.

Utilizamos Python para desenvolver os modelos de machine learning. Em todas as abordagens, implementamos precauções para evitar o "data leakage", garantindo assim que a avaliação dos modelos de treino e teste fosse o mais justa e realista possível. Para a regressão linear e regressão Ridge, empregamos a técnica de validação cruzada. No entanto, para o Keras LSTM, a validação cruzada pode não ser apropriada devido às peculiaridades das sequências temporais, e, portanto, não foi aplicada neste contexto.

Regressão Linear

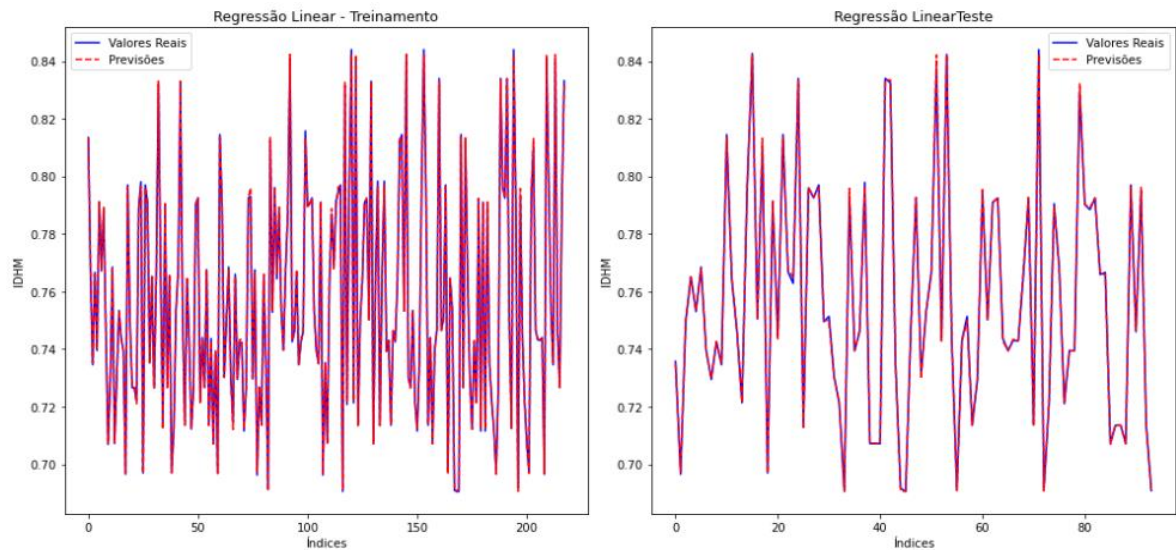
Foi utilizada a regressão linear múltipla para modelar a relação entre o IDHM (variável dependente) e as variáveis mês, total de repasses e estado (UF) - estas sendo as variáveis independentes. O objetivo é utilizar esse modelo para fazer previsões sobre o IDHM em 2023.

A análise de regressão linear visa prever o valor de uma variável (dependente) com base no valor de uma ou mais outras variáveis (independentes). Essa análise estima os coeficientes da equação linear que melhor descrevem a relação entre as variáveis. O ajuste é feito de tal forma que a discrepância entre os valores observados e os valores previstos pela equação seja a menor possível. Esse ajuste é frequentemente realizado utilizando o método dos mínimos quadrados.

Para esclarecer, na regressão linear simples, o modelo se ajusta a uma linha reta que melhor representa a relação entre a variável dependente e independente. No entanto, na regressão linear múltipla, a modelagem pode envolver múltiplas dimensões devido à presença de múltiplas variáveis independentes.

A regressão linear é uma ferramenta poderosa e amplamente utilizada em análises estatísticas. Porém, como todos os modelos, tem suas suposições e limitações que precisam ser consideradas ao interpretar os resultados.

Os gráficos a seguir apresentam as previsões do modelo em comparação com os valores reais para os conjuntos de treinamento e teste, dando uma visão mais clara sobre a precisão e eficácia do modelo no que se refere à projeção do IDHM em 2023.



RMSE: 0.000984415056039309

R²: 0.9994163867673004

O RMSE e o R² são calculados para avaliar o desempenho do modelo no conjunto de teste. O RMSE mede o erro médio das previsões, enquanto o R² indica a proporção da variância na variável dependente que é previsível a partir das variáveis independentes. O RMSE mede a magnitude média dos erros entre os valores observados e os valores previstos por um modelo. Um RMSE de 0.000984415056039309 é extremamente baixo, sugerindo que o modelo tem um erro médio muito pequeno em suas previsões. Para o caso do R², um valor de 0.9994163867673004 sugere que o modelo explica aproximadamente 99.94% da variação nos dados, o que é extremamente alto.

Com validação cruzada os resultados também são muito próximos e podem ser interpretados da mesma forma da avaliação dos dados sem a validação cruzada:

RMSE (validação cruzada): 0.0009708455203538355

R² (validação cruzada): 0.9993859340382493

Regressão RIDGE

A regressão Ridge é uma técnica de regularização usada em modelos de regressão linear para evitar o sobre ajuste e lidar com a multicolinearidade. A multicolinearidade surge quando duas ou mais variáveis independentes no modelo são altamente correlacionadas, o que pode tornar difícil ou impossível estimar seus coeficientes de regressão de forma confiável. Ao contrário de técnicas de seleção de variáveis, que excluem algumas variáveis do modelo, a regressão Ridge encolhe a magnitude dos coeficientes, em especial para variáveis colineares.

A técnica faz isso adicionando uma penalidade ao quadrado da magnitude dos coeficientes, buscando reduzir a magnitude deles. Esta penalização resulta em coeficientes menores, tornando o modelo mais robusto e menos sensível a variações nos dados de treinamento.

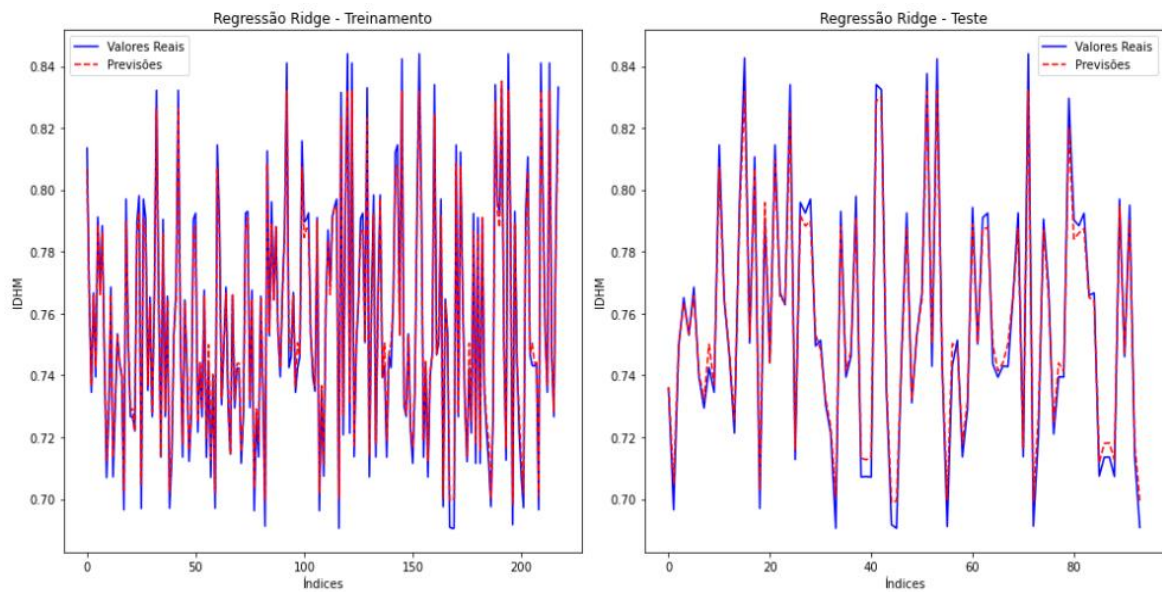
Matematicamente, o objetivo da regressão Ridge é minimizar a seguinte função de custo:

$$J(\theta) = \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^n \theta_j^2$$

Onde:

- $J(\theta)$ é a função de custo.
- $h_{\theta}(x^{(i)})$ é a hipótese do modelo (predição) para a i-ésima observação.
- $y^{(i)}$ é o valor real da i-ésima observação.
- θ é o vetor de coeficientes do modelo.
- λ é o parâmetro de regularização.
- n é o número de características (sem incluir o termo de interceptação).

O termo $\lambda \sum_{j=1}^n \theta_j^2$ é a penalidade que é adicionada à função de custo normal da regressão linear. A magnitude do parâmetro de regularização λ determina o peso da penalidade.



Ao aplicar o modelo Ridge de regressão, foi obtido um RMSE de 0.005070794523817809. Este valor de RMSE, que representa o erro quadrático médio da raiz, pode ser considerado baixo, indicando que o modelo tem um pequeno desvio entre os valores previstos e os valores observados. O coeficiente de determinação, R^2 , apresentou um valor de 0.9845146630053412. Isso significa que o modelo é capaz de explicar aproximadamente 98,45% da variação nos dados, um valor extremamente alto e indicativo de um bom ajuste do modelo aos dados.

RMSE - Regressão Ridge: 0.005070794523817809
 R^2 - Regressão Ridge: 0.9845146630053412

Após aplicar a técnica da validação cruzada, os resultados foram surpreendentemente semelhantes aos obtidos sem o uso da validação cruzada. Enquanto as métricas individuais podem ser interpretadas da mesma maneira, a consistência dos resultados entre a validação padrão e a validação cruzada sugere que nosso modelo é robusto e tem um desempenho estável em diferentes subconjuntos do conjunto de dados.

RMSE (validação cruzada) - Teste: 0.005070794523817809
 R^2 (validação cruzada) - Teste: 0.9845146630053412

Keras LSTM

Keras é uma biblioteca de código aberto escrita em Python que facilita a criação de modelos de aprendizado profundo, incluindo redes neurais profundas. Além de sua simplicidade e facilidade de uso, o Keras oferece uma interface modular e extensível, permitindo a criação tanto de modelos sequenciais quanto de modelos mais complexos, baseados em grafos. Ele se integra perfeitamente com bibliotecas de backend populares como TensorFlow, Theano e Microsoft Cognitive Toolkit (CNTK), oferecendo eficiência e otimização no treinamento de modelos.

LSTM, que significa "Long Short-Term Memory", é um tipo específico de Rede Neural Recorrente (RNN). Ao contrário das redes neurais tradicionais, as RNNs têm a capacidade de manter informações de estados anteriores em sequências. Isso é crucial para tarefas em que a ordem e a continuidade dos dados são importantes. Por essa razão, elas têm sido amplamente utilizadas em aplicações como tradução automática, geração de texto e análise de sentimento. No entanto, as RNNs padrão enfrentam desafios, principalmente o desaparecimento e a explosão do gradiente, que interferem no aprendizado de dependências de longo prazo nos dados.

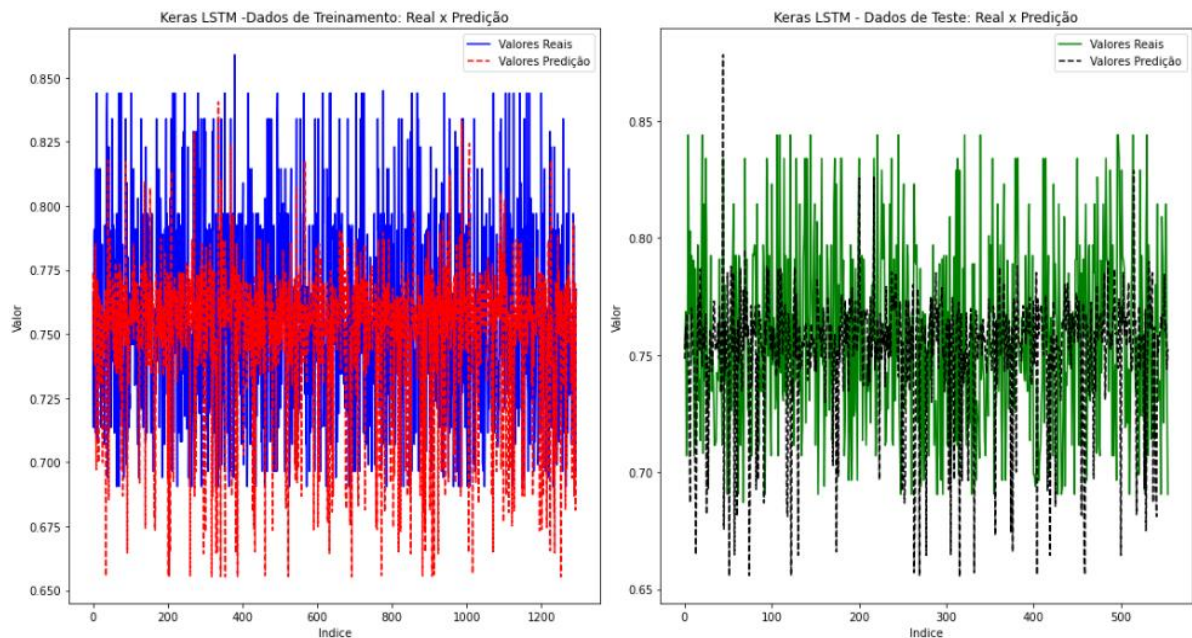
A LSTM, introduzida por Sepp Hochreiter e Jürgen Schmidhuber em 1997, foi uma revolução no campo do aprendizado profundo. Seu design inteligente, com portões que controlam o fluxo de informações, permitiu que as redes aprendessem dependências de longo alcance, superando os problemas enfrentados pelas RNNs tradicionais. Além dos portões de entrada, esquecimento e saída, a LSTM possui um estado de célula que armazena informações ao longo do tempo, garantindo que o sinal não se atenua à medida que passa pela rede.

As LSTMs são um tipo de rede neural recorrente (RNN) que é bem adequado para sequências ou dados temporais. Embora sejam frequentemente usadas para tarefas de classificação, como reconhecimento de padrões em séries temporais ou análise de sentimentos em texto, elas também podem ser configuradas para prever valores contínuos, o que é o objetivo de modelos de regressão.

```

Epoch 1/12
41/41 [=====] - 2s 10ms/step - loss: 0.3994 - val_loss: 0.2561
Epoch 2/12
41/41 [=====] - 0s 4ms/step - loss: 0.1381 - val_loss: 0.0462
Epoch 3/12
41/41 [=====] - 0s 4ms/step - loss: 0.0262 - val_loss: 0.0198
Epoch 4/12
41/41 [=====] - 0s 4ms/step - loss: 0.0176 - val_loss: 0.0161
Epoch 5/12
41/41 [=====] - 0s 4ms/step - loss: 0.0143 - val_loss: 0.0132
Epoch 6/12
41/41 [=====] - 0s 4ms/step - loss: 0.0116 - val_loss: 0.0107
Epoch 7/12
41/41 [=====] - 0s 5ms/step - loss: 0.0091 - val_loss: 0.0085
Epoch 8/12
41/41 [=====] - 0s 4ms/step - loss: 0.0070 - val_loss: 0.0065
Epoch 9/12
41/41 [=====] - 0s 4ms/step - loss: 0.0053 - val_loss: 0.0050
Epoch 10/12
41/41 [=====] - 0s 4ms/step - loss: 0.0039 - val_loss: 0.0038
Epoch 11/12
41/41 [=====] - 0s 4ms/step - loss: 0.0030 - val_loss: 0.0030
Epoch 12/12
41/41 [=====] - 0s 4ms/step - loss: 0.0023 - val_loss: 0.0024
18/18 [=====] - 0s 2ms/step
1/1 [=====] - 0s 17ms/step
1/1 [=====] - 0s 19ms/step
1/1 [=====] - 0s 21ms/step
1/1 [=====] - 0s 23ms/step
1/1 [=====] - 0s 26ms/step
1/1 [=====] - 0s 24ms/step
1/1 [=====] - 0s 26ms/step
1/1 [=====] - 0s 14ms/step
1/1 [=====] - 0s 18ms/step
1/1 [=====] - 0s 17ms/step
1/1 [=====] - 0s 18ms/step
1/1 [=====] - 0s 20ms/step

```



Para este algoritmo, foram utilizados para 12 épocas. Uma época é quando todo o conjunto de treinamento é passado para frente e para trás através da rede neuronal apenas uma vez.

Os parâmetros `loss` e `val_loss` estão diminuindo à medida que as épocas avançam, indicando que o modelo está aprendendo e melhorando seu desempenho à medida que é treinado. Um total de 41 lotes foram processados de um total de 41 no conjunto de treinamento. 18/18 indica o mesmo, mas para um conjunto de validação ou teste. Após as 12 épocas, o modelo foi testado em várias amostras individuais (conjuntos de teste muito pequenos) consecutivamente, este padrão é indicado pela existência de várias linhas com "1/1". Desta forma, entendeu-se que o treinamento foi bem-sucedido, já que a perda no conjunto de treinamento e no conjunto de validação diminuiu consistentemente ao longo das 12 épocas.

```
MSE (Treinamento): 0.0046
MAE (Treinamento): 0.0498
MSE (Teste): 0.0051
MAE (Teste): 0.0538
```

MSE (Erro Quadrático Médio)

MSE (Treinamento): 0.0046: Isso significa que o erro quadrático médio das previsões do modelo no conjunto de treinamento é 0.0046. O MSE dá mais peso aos erros maiores, pois os erros são elevados ao quadrado. Em outras palavras, erros grandes são penalizados mais do que erros pequenos.

MSE (Teste): 0.0051: Isso indica que o modelo tem um erro quadrático médio ligeiramente maior no conjunto de teste em comparação com o conjunto de treinamento. É comum ter um desempenho ligeiramente inferior no conjunto de teste porque o modelo não viu esses dados durante o treinamento.

MAE (Erro Absoluto Médio)

MAE (Treinamento): 0.0498: Isso significa que, em média, as previsões do modelo estão a cerca de 0.0498 unidades do valor real no conjunto de treinamento.

MAE (Teste): 0.0538: No conjunto de teste, as previsões do modelo estão, em média, a cerca de 0.0538 unidades do valor real. Isso é ligeiramente mais alto do que no conjunto de treinamento, o que, novamente, é comum.


Avaliação geral do desempenho do modelo:

Proximidade de desempenho entre treinamento e teste: Os valores de MSE e MAE são relativamente próximos entre os conjuntos de treinamento e teste, o que pode ser entendido como uma boa performance. Isso sugere que o modelo não sofreu overfitting (onde o modelo se ajusta demais aos dados de treinamento e tem um desempenho ruim em dados não vistos).

Magnitude dos erros: Um MAE de 0.0538 significa que o modelo está errando, em média, 5.38% do valor real no conjunto de teste. Este percentual é aceitável considerando-se o conjunto de dados e o contexto avaliado.

6. Apresentação dos Resultados

O fluxo de trabalho deste estudo está representado através do Data Science Workflow Canvas:


Data Science Workflow Canvas*

Start here. The sections below are ordered intentionally to make you state your goals first, followed by steps to achieve those goals. You're allowed to switch orders of these steps!

Title:

<div>● Problem Statement</div> <p>Como os montantes das transferências constitucionais afetam o IDHM ?</p>	<div>● Outcomes/Predictions</div> <p>Previsão através de variáveis</p> <p>X = Mês, Total Repasses, UF</p> <p>y = IDHM</p>	<div>● Data Acquisition</div> <p>Tesouro Transparente</p> <p>Atlas do Desenvolvimento Humano no Brasil</p>
<div>● Modeling</div> <p>Algoritmos de machine learning de aprendizado supervisionado para modelos de regressão.</p> <p>Regressão Linear</p> <p>Regressão Ridge</p> <p>Keras - LSTM</p>	<div>● Model Evaluation</div> <p>Métricas: RMSE e R² (Regressões)</p> <p>Métricas: MSE e MAE (Keras LSTM)</p>	<div>● Data Preparation</div> <p>Extrair os dados necessários</p> <p>Tratar os dados</p> <p>Encontrar os dados necessários para a criação dos modelos</p>

✓ Activation

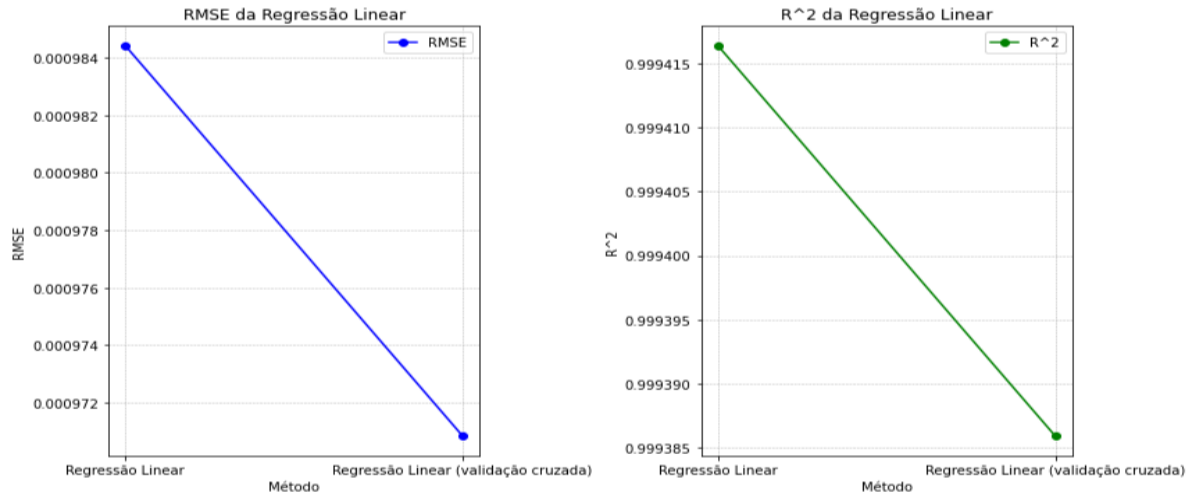
When you finish filling out the canvas above, now you can begin implementing your data science workflow in roughly this order.

● Problem Statement → ● Data Acquisition → ● Data Prep → ● Modeling → ● Outcomes/Preds → ● Model Eval

Conceptualized by Jasmirine Vasconcelos using notes from General Assembly's Data Science Immersive. Format inspired by Business Model Canvas.

Os resultados dos modelos testados são mostrados abaixo:

Regressão Linear:



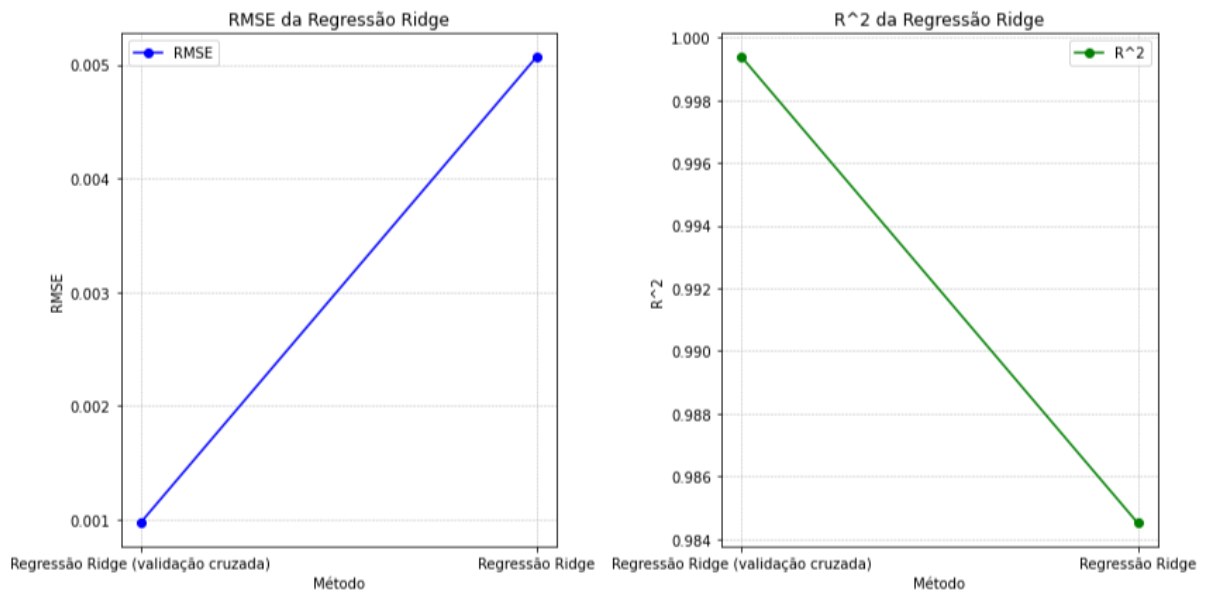
Ambos os modelos, com e sem validação cruzada, apresentam desempenho excepcional, com erros muito baixos (RMSE) e altíssima capacidade de explicar a variabilidade dos dados (R^2). A diferença entre os RMSEs dos dois modelos é muito pequena, mas sugere que a validação cruzada proporciona um ligeiro aumento na precisão das previsões. No entanto, essa diferença é tão pequena que, em muitos casos práticos, pode não ser significativa. A validação cruzada é uma técnica que avalia a capacidade de generalização do modelo. Dada a pequena diferença nos resultados, pode-se inferir que o conjunto de dados é bastante consistente e que o modelo não sofre de problemas comuns como sobreajuste.

Em termos de escolha de modelo, ambos são altamente recomendáveis, mas a decisão de usar validação cruzada pode depender de considerações como o tempo de computação, o tamanho do dataset, entre outros.

Desta forma, a Regressão Linear mostrou-se extremamente eficaz para este conjunto de dados, tanto com quanto sem validação cruzada. A pequena diferença nos resultados sugere que o modelo é robusto e que o conjunto de dados é consistente.

Desempenho excepcionalmente bom com erros muito baixos e uma alta capacidade de explicar a variabilidade dos dados. Pequena diferença entre os resultados com e sem validação cruzada, indicando robustez do modelo e consistência do conjunto de dados.

Regressão Ridge:



Em geral, os dois modelos apresentaram bons resultados, visto que ambos os valores de RMSE são baixos e os R^2 são altos. No entanto, a validação cruzada, que é uma técnica para avaliar a capacidade do modelo de generalizar para novos dados, mostrou resultados superiores tanto em RMSE quanto em R^2 . Isso sugere que o uso de validação cruzada pode proporcionar um modelo mais robusto e bem ajustado aos dados.

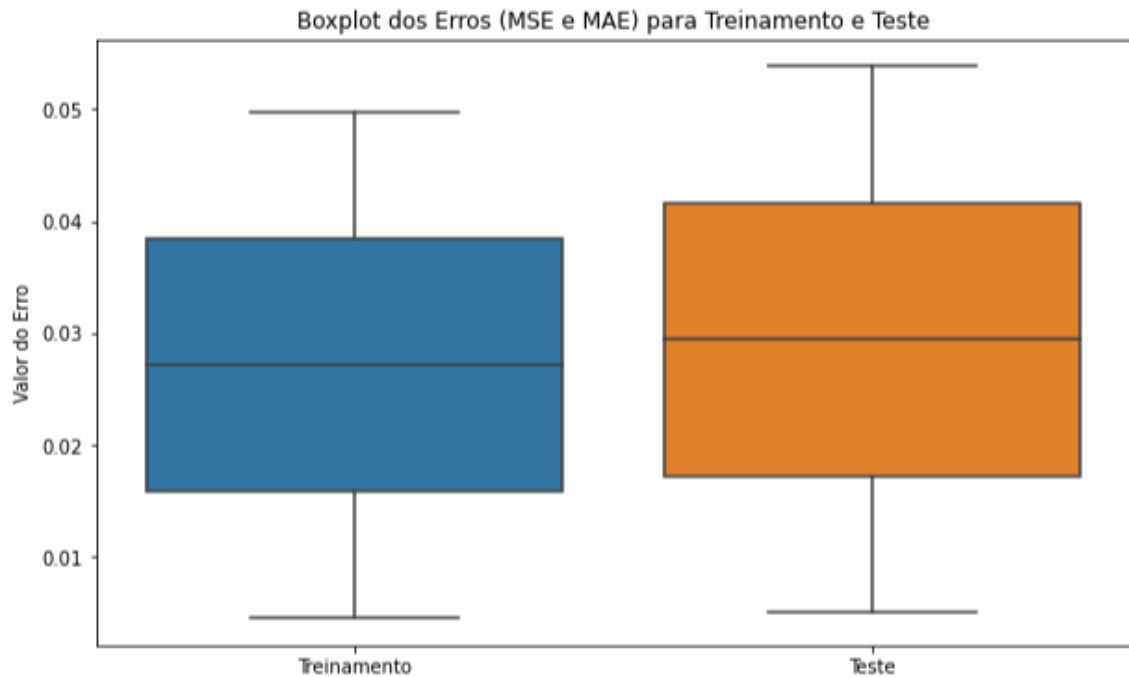
A diferença no RMSE entre os modelos com e sem validação cruzada é notável. O RMSE mais alto (sem validação cruzada) sugeriu que esse modelo pode ter algum sobreajuste aos dados de treino, diminuindo sua capacidade de generalização.

Em aplicações práticas, considerando estes resultados, seria aconselhável optar pelo modelo com validação cruzada, pois oferece melhor capacidade de generalização e ajuste mais preciso aos dados.

Em resumo, enquanto ambos os modelos são eficazes, a Regressão Ridge com validação cruzada demonstrou ser a mais robusta e confiável para essa aplicação específica.

O modelo mostrou um bom desempenho, mas a validação cruzada trouxe melhorias notáveis tanto em RMSE quanto em R^2 . A diferença no RMSE entre os modelos com e sem validação cruzada sugere uma melhor capacidade de generalização no modelo com validação cruzada.

Keras – LSTM:



O desempenho do modelo LSTM parece consistente entre os conjuntos de treinamento e teste, visto que as métricas MSE e MAE são relativamente próximas em ambos os conjuntos. Isso é um bom sinal, pois sugere que o modelo não está sobreajustado ao conjunto de treinamento. A diferença entre MSE e MAE pode ser interpretada da seguinte forma: o MSE será influenciado por erros maiores (outliers), enquanto o MAE fornecerá uma visão mais "central" dos erros. O fato de o MSE ser relativamente baixo, mas o MAE ser mais alto (comparativamente), pode sugerir que existem alguns erros pontuais maiores que estão influenciando o MSE. Mas, de forma geral, os números apresentam uma margem de erro baixo, considerando o contexto da pesquisa. Desempenho consistente entre treinamento e teste.

A diferença entre MSE e MAE indica que pode haver alguns erros pontuais maiores influenciando o MSE. Mesmo assim, o modelo Keras LSTM mostra um bom desempenho, com erros consistentes entre treinamento e teste.

Para finalizar os resultados do estudo, concluímos que para casos de aplicação reais e que demandam altos níveis de qualidade, não existe um modelo melhor do que o outro e sim, que estes modelos possuam performance e aplicação mais adequadas conforme a necessidade. Contudo, consideramos alguns aspectos e demonstramos uma avaliação final sobre o estudo efetuado e qual o modelo mais adequado para tal:

Para a avaliação final, foram considerados os seguintes aspectos:

Se o objetivo é obter um modelo com a melhor capacidade de generalização e precisão:

A Regressão Ridge com validação cruzada seria a escolha recomendada, já que mostrou os melhores resultados em termos de RMSE e R^2 , sugerindo maior robustez e confiabilidade.

Se a preocupação é a simplicidade do modelo e rápida implementação:

A Regressão Linear, seja com ou sem validação cruzada, pode ser a preferida, pois tem um desempenho excepcionalmente bom e é geralmente mais simples e rápida de ser implementada quando comparada com técnicas mais complexas como LSTM.

Se o conjunto de dados tem características sequenciais ou temporais, e a capacidade de lidar com sequências é essencial:

O Keras LSTM pode ser a melhor escolha. LSTMs são especialmente adequadas para lidar com séries temporais ou dados sequenciais, e o desempenho consistente indicado na avaliação sugere que este modelo também é uma opção sólida.

Em conclusão, todos os modelos avaliados têm seus pontos fortes, e a escolha do modelo ideal deve ser baseada no tipo de dados, objetivos da análise e recursos disponíveis (como tempo de computação).

Em termos estritamente de desempenho, a **Regressão Ridge com validação cruzada pareceu ser a mais promissora** para este conjunto de dados específico.

Considerações Finais:

Neste estudo, abordamos a tarefa de prever o IDHM para 2023 utilizando três técnicas diferentes: regressão linear, regressão Ridge e Keras LSTM. Cada um desses métodos trouxe insights distintos sobre o problema e sobre o conjunto de dados analisado.

A regressão linear, sendo um dos métodos mais tradicionais, mostrou um desempenho notável, com o modelo explicando quase 99,94% da variação nos dados. Este resultado é indicativo de sua robustez e relevância contínua em problemas de regressão.

A regressão Ridge, uma extensão da regressão linear com regularização, também exibiu desempenho excelente, abordando potenciais problemas de multicolinearidade e garantindo que o modelo não se ajustasse excessivamente aos dados. A semelhança entre os resultados com e sem validação cruzada confirma a confiabilidade do modelo.

O Keras LSTM representou uma abordagem mais moderna, explorando a capacidade das redes neurais em modelar sequências temporais. Apesar de não ser a ferramenta tradicional para regressão, seus resultados foram promissores, indicando que vale a pena considerar essa abordagem para previsões temporais futuras.

No entanto, é crucial observar que, embora os modelos tenham demonstrado desempenho de alto nível, sempre há margem para melhoria. A natureza do IDHM e suas influências podem ser mais complexas do que os modelos sugerem. Além disso, o verdadeiro teste de qualquer modelo preditivo é sua capacidade de prever dados futuros com precisão, por isso será interessante observar o desempenho desses modelos à medida que novos dados do IDHM estiverem disponíveis em 2023 e além.

Por fim, esta análise ressalta a importância da experimentação em ciência de dados. Diferentes abordagens trazem diferentes perspectivas e, muitas vezes, a

combinação dessas perspectivas pode oferecer o quadro mais completo e preciso de um problema. Assim, é fundamental que os profissionais da área permaneçam curiosos, flexíveis e abertos a novas técnicas e métodos.

7. Links

Link Youtube

https://youtu.be/ZM_ZbfTgFVw?si=PjXKIkxI-lwM8iCR

GituHub

<https://github.com/andrevcavalcante/TCCPUCMG.git>

REFERÊNCIAS

- HOSMER, D. W.; LEMESHOW, S.; STURDIVANT, R. X. **Applied Logistic Regression**. Wiley, 2013.
- Weiss, S., M., Kulikowski, C., A., **Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Networks, Machine Learning and Expert Systems**, Morgan Kaufmann, San Mateo, 1991.
- J. Han and M. Kamber, **Data Mining: Concepts and Techniques**. Morgan Kaufmann Publishers, San Francisco, 2001.
- Escovedo, Tatiana (2020-02-27T22:58:59). **Introdução a Data Science**. Casa do Código. Edição do Kindle.
- Géron, Aurélien. **Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow**. Edição do Kindle.”
- <https://www.tesourotransparente.gov.br/ckan/dataset/transferencias-constitucionais-para-municipios>
- <https://www.undp.org/pt/brazil/o-que-%C3%A9-o-idh>
- <http://www.atlasbrasil.org.br/consulta/planilha>
- https://www.planalto.gov.br/ccivil_03/constituicao/constituicao.htm