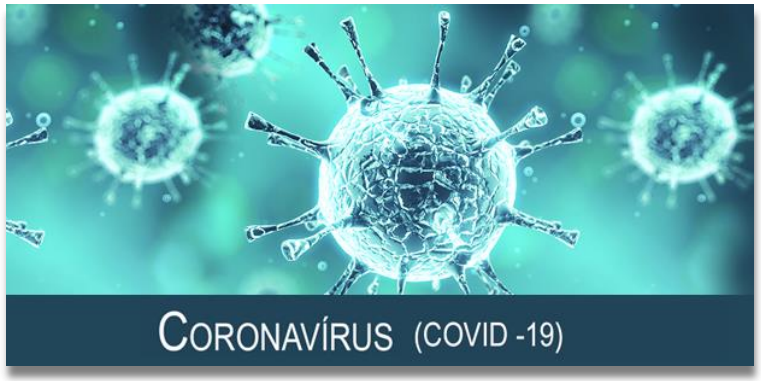
PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA
Pós-graduação *Lato Sensu* em Ciência de Dados e Big Data

ANÁLISE DO IMPACTO DE OUTLIERS EM MODELO DE REGRESSÃO PARA PREVISÃO DO VALOR MÁXIMO DE DESPESAS DE CAMPANHA DE CANDIDATOS ELEITOS NAS ELEIÇÕES MUNICIPAIS BRASILEIRAS DE 2020 COM BASE NO PIB E PORTE POPULACIONAL

Arthur de Moura Mota

CONTEXTUALIZAÇÃO



CONTEXTUALIZAÇÃO

Valor das Campanhas Políticas:

No ano de 2020, o Brasil testemunhou um marco nas eleições municipais, caracterizado por um notável aumento nos gastos de campanha em relação a pleitos anteriores. Esse fenômeno foi motivado, em parte, pela proibição das doações empresariais em 2015, que impeliu os candidatos a dependerem mais fortemente de recursos próprios e de apoio financeiro obtido por meio de financia- mento coletivo (crowdfunding).

PIB Municipal:

O PIB municipal, um indicador crítico, espelha a atividade econômica de cada localidade. Em 2020, muitos municípios brasileiros enfrentaram desafios econômicos iminentes devido aos impactosda pandemia de COVID-19, refletindo-se diretamente no desempenho do PIB de diversas regiões. Alémdisso, é imprescindível destacar a profunda desigualdade econômica que permeia o Brasil, com municípios de variados tamanhos e níveis de desenvolvimento.

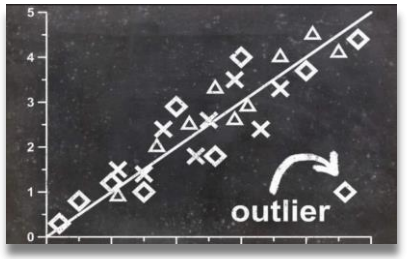
Porte Populacional de Município:

O porte de um município está intrinsicamente ligado a seu tamanho populacional e à sua importância econômica. No contexto brasileiro, os municípios podem ser categorizados como pequenos,médios e grandes, baseados em critérios populacionais e de PIB. Essa classificação exerce uma influência direta na quantidade de recursos disponíveis para as campanhas políticas, bem como na complexidade das estratégias eleitorais adotadas pelos candidatos.

O PROBLEMA PROPOSTO E OBJETIVOS

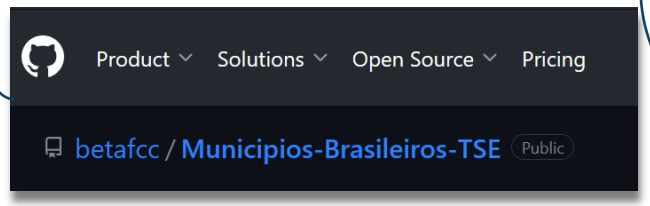
Este trabalho irá analisar dados das eleições municipais de 2020 no Brasil, abrangendo candidatos eleitos em diferentes municípios. Nossas fontes principais de dados incluem informações eleitorais, econômicas, como o Produto Interno Bruto (PIB) municipal, e dados populacionais.

O objetivo principal é investigar como fatores econômicos (PIB municipal) e demográficos (porte populacional dos municípios) influenciaram estratégias de campanha e resultados eleitorais em 2020. Concentraremos nossa análise na identificação e análise de outliers (valores atípicos nos) gastos de campanha e em entender como esses valores atípicos afetaram os resultados das eleições.



COLETA DE DADOS

ELEIÇÕES



DEMOGRÁFICOS

PROCESSAMENTO/TRATAMENTO DE DADOS

TECNOLOGIAS E BIBLIOTECAS


```python
# Importando a biblioteca pandas para manipulação de dados
import pandas as pd

# Importando as bibliotecas matplotlib e seaborn para visualização de dados
import matplotlib.pyplot as plt
import seaborn as sns

# Configurando a paleta de cores do Seaborn para uma visualização agradável
sns.set_palette("viridis", 30)

# Definindo a paleta de cores do Seaborn como um mapa de cores (cmap) para uso posterior
sns.color_palette("viridis", as_cmap=True)

# Importando bibliotecas do scikit-learn para preparação de dados e modelagem
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.ensemble import GradientBoostingRegressor  # Importando o modelo GradientBoostingRegressor
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error

# Importando a biblioteca statsmodels para análise estatística
import statsmodels.api as sm

# Importando a função ols (Ordinary Least Squares) do statsmodels para ajuste de modelos lineares
from statsmodels.formula.api import ols

# Exibe todas as colunas do DataFrame sem truncamento
pd.set_option('display.max_columns', None)

# Configuração para exibir os números com precisão fixa de 6 casas decimais
pd.set_option('display.float_format', '{:.4f}'.format)

pd.set_option('display.max_rows', None)
```

PROCESSAMENTO/TRATAMENTO DE DADOS

PRINCIPAIS PROCESSAMENTOS E TRATAMENTOS

```
# Lista dos valores de 'DS_SIT_TOT_TURNO' desejados
eleitos_selecionados_eleito = ['ELEITO', 'ELEITO POR QP', 'ELEITO POR MÉDIA']

# Filtra o dataframe para manter apenas as linhas com as situações desejadas
df_eleitos = df_eleitos[df_eleitos['DS_SIT_TOT_TURNO'].isin(eleitos_selecionados_eleito)]
```

```
# Campos preenchidos com #NULO significam que a informação está em branco no banco de dados.
# O correspondente para #NULO nos campos numéricos é -1;
```

```
# Removendo valores negativos do DataFrame df_eleitos_com_IBGE
df_eleitos_IBGE = df_eleitos_IBGE[df_eleitos_IBGE['VR_DESPESA_MAX_CAMPANHA'] >= 0]

# Filtrando apenas os dados relacionados aos vice-prefeitos após a remoção de valores negativos
print("DataFrame df_eleitos_vices depois da remoção de valores negativos:")
df_eleitos_vices = df_eleitos_IBGE[df_eleitos_IBGE['DS_CARGO'] == 'VICE-PREFEITO']
print(df_eleitos_vices.info())
```

```
DataFrame df_eleitos_vices depois da remoção de valores negativos:
<class 'pandas.core.frame.DataFrame'>
Int64Index: 0 entries
Data columns (total 8 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   DS_CARGO                0 non-null      object
 1   TP_AGREMIACAO           0 non-null      object
 2   SG_PARTIDO              0 non-null      object
 3   NM_PARTIDO              0 non-null      object
 4   ST_REELEICAO            0 non-null      object
 5   VR_DESPESA_MAX_CAMPANHA 0 non-null      float64
 6   capital                 0 non-null      int64
 7   codigo_ibge             0 non-null      object
dtypes: float64(1), int64(1), object(6)
memory usage: 0.0+ bytes
None
```

```
df_dados_demograficos = df_renda + df_populacao
```

```
# Realizando a junção entre os DataFrames usando a coluna 'SG_UE' e 'codigo_tse' como chave
df_eleitos_IBGE = pd.merge(df_eleitos, df_TSE_IBGE, left_on='SG_UE', right_on='codigo_tse',
                           how='inner')[colunas_df_eleitos + colunas_df_TSE_IBGE]
```

```
df_eleitos_dados_demograficos = df_eleitos_IBGE + df_dados_demograficos
```

ANÁLISE E EXPLORAÇÃO DOS DADOS

SEPARAÇÃO DOS DADOS EM DATAFRAMES DIFERENTES

```
# Criar DataFrames df_eleitos_prefeitos  e df_eleitos_vereadores separados com base na coluna 'DS_CARGO'
df_eleitos_prefeitos = df_eleitos_dados_demograficos[df_eleitos_dados_demograficos['DS_CARGO'] == 'PREFEITO']
df_eleitos_vereadores = df_eleitos_dados_demograficos[df_eleitos_dados_demograficos['DS_CARGO'] == 'VEREADOR']
```

```
Informações sobre df_vereadores:
<class 'pandas.core.frame.DataFrame'>
Int64Index: 57956 entries, 18571 to 32662
Data columns (total 19 columns):
 #   Column                                                                                                                                  Non-Null Count  Dtype
---  ------                                                                                                                                  --------------  -----
 0   NOME DO MUNICÍPIO                                                                                                                       57956 non-null  object
 1   Código do Município                                                                                                                     57956 non-null  object
 2   UF                                                                                                                                      57956 non-null  category
 3   Nome_da_Grande_Região                                                                                                                   57956 non-null  category
 4   POPULAÇÃO ESTIMADA                                                                                                                      57956 non-null  int64
 5   capital                                                                                                                                 57956 non-null  category
 6   DS_CARGO                                                                                                                                57956 non-null  category
 7   VR_DESPESA_MAX_CAMPANHA                                                                                                                 57956 non-null  float64
 8   SG_PARTIDO                                                                                                                              57956 non-null  category
 9   NM_PARTIDO                                                                                                                              57956 non-null  category
 10  TP_AGREMIACAO                                                                                                                           57956 non-null  category
 11  ST_REELEICAO                                                                                                                            57956 non-null  category
 12  Valor adicionado bruto da Agropecuária, a preços correntes (R$ 1.000)                                                                   57956 non-null  float64
 13  Valor adicionado bruto da Indústria, a preços correntes (R$ 1.000)                                                                      57956 non-null  float64
 14  Valor adicionado bruto dos Serviços, a preços correntes - exceto Administração, defesa, educação e saúde públicas e seguridade social (R$ 1.000)  57956 non-null  float64
 15  Valor adicionado bruto da Administração, defesa, educação e saúde públicas e seguridade social, a preços correntes (R$ 1.000)          57956 non-null  float64
 16  Valor adicionado bruto total, a preços correntes (R$ 1.000)                                                                             57956 non-null  float64
 17  Impostos, líquidos de subsídios, sobre produtos, a preços correntes (R$ 1.000)                                                          57956 non-null  float64
 18  Produto Interno Bruto per capita, a preços correntes (R$ 1,00)                                                                          57956 non-null  float64
dtypes: category(8), float64(8), int64(1), object(2)
memory usage: 5.8+ MB
None
```

```
**********************************************************************************************************
INICIO Análise e Exploração dos Dados - Criação de Modelos de Machine Learning
**********************************************************************************************************
Tamanho de df_eleitos_prefeitos: (5492, 19)
Tamanho de df_eleitos_vereadores: (57956, 19)

Informações sobre df_prefeitos:
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5492 entries, 18569 to 32667
Data columns (total 19 columns):
 #   Column                                                                                                                                  Non-Null Count  Dtype
---  ------                                                                                                                                  --------------  -----
 0   NOME DO MUNICÍPIO                                                                                                                       5492 non-null   object
 1   Código do Município                                                                                                                     5492 non-null   object
 2   UF                                                                                                                                      5492 non-null   category
 3   Nome_da_Grande_Região                                                                                                                   5492 non-null   category
 4   POPULAÇÃO ESTIMADA                                                                                                                      5492 non-null   int64
 5   capital                                                                                                                                 5492 non-null   category
 6   DS_CARGO                                                                                                                                5492 non-null   category
 7   VR_DESPESA_MAX_CAMPANHA                                                                                                                 5492 non-null   float64
 8   SG_PARTIDO                                                                                                                              5492 non-null   category
 9   NM_PARTIDO                                                                                                                              5492 non-null   category
 10  TP_AGREMIACAO                                                                                                                           5492 non-null   category
 11  ST_REELEICAO                                                                                                                            5492 non-null   category
 12  Valor adicionado bruto da Agropecuária, a preços correntes (R$ 1.000)                                                                   5492 non-null   float64
 13  Valor adicionado bruto da Indústria, a preços correntes (R$ 1.000)                                                                      5492 non-null   float64
 14  Valor adicionado bruto dos Serviços, a preços correntes - exceto Administração, defesa, educação e saúde públicas e seguridade social (R$ 1.000)  5492 non-null   float64
 15  Valor adicionado bruto da Administração, defesa, educação e saúde públicas e seguridade social, a preços correntes (R$ 1.000)          5492 non-null   float64
 16  Valor adicionado bruto total, a preços correntes (R$ 1.000)                                                                             5492 non-null   float64
 17  Impostos, líquidos de subsídios, sobre produtos, a preços correntes (R$ 1.000)                                                          5492 non-null   float64
 18  Produto Interno Bruto per capita, a preços correntes (R$ 1,00)                                                                          5492 non-null   float64
dtypes: category(8), float64(8), int64(1), object(2)
memory usage: 562.3+ KB
None
```

ANÁLISE E EXPLORAÇÃO DOS DADOS

ANÁLISE DE VARIÁVEIS CATEGÓRICAS

```python
def analisar_ANOVA(df,colunas_categoricas):
    # Criar uma fórmula para a ANOVA
    formula_anova = 'VR_DESPESA_MAX_CAMPANHA ~ ' + ' + '.join(['C(' + coluna + ')' for coluna in colunas_categoricas])
    modelo_anova = ols(formula_anova, data=df).fit()

    tabela_anova = sm.stats.anova_lm(modelo_anova, typ=2)

    # Exibir os resultados da ANOVA
    print(f"Resultados da Análise de Variância (ANOVA) para as colunas: {', '.join(colunas_categoricas)}")
    print(tabela_anova)
```

ANOVA para df_eleitos_dados_demograficos

Resultados da Análise de Variância (ANOVA) para as colunas: DS_CARGO, TP_AGREMIACAO, SG_PARTIDO, NM_PARTIDO, ST_REELEICAO, capital, UF, Nome_da_Grande_Região

	sum_sq	df	F	PR(>F)
C(DS_CARGO)	10132758441165.1270	1.0000	198.0481	0.0000
C(TP_AGREMIACAO)	4994050246960.6172	1.0000	97.6103	0.0000
C(SG_PARTIDO)	95691006891.2541	28.0000	0.0668	1.0000
C(NM_PARTIDO)	95691006866.0873	28.0000	0.0668	1.0000
C(ST_REELEICAO)	45377846450.3708	1.0000	0.8869	0.3463
C(capital)	618254698413249.5000	1.0000	12083.9888	0.0000
C(UF)	49689644749.0736	25.0000	0.0388	0.9971
C(Nome_da_Grande_Região)	7950343159.7408	4.0000	0.0388	0.9971
Residual	3243179687725583.0000	63389.0000	NaN	NaN

ANOVA para df_prefeitos

Resultados da Análise de Variância (ANOVA) para as colunas: DS_CARGO, TP_AGREMIACAO, SG_PARTIDO, NM_PARTIDO, ST_REELEICAO, capital, UF, Nome_da_Grande_Região

	sum_sq	df	F	PR(>F)
C(DS_CARGO)	10132758441165.1270	1.0000	198.0481	0.0000
C(TP_AGREMIACAO)	4994050246960.6172	1.0000	97.6103	0.0000
C(SG_PARTIDO)	95691006891.2541	28.0000	0.0668	1.0000
C(NM_PARTIDO)	95691006866.0873	28.0000	0.0668	1.0000
C(ST_REELEICAO)	45377846450.3708	1.0000	0.8869	0.3463
C(capital)	618254698413249.5000	1.0000	12083.9888	0.0000
C(UF)	49689644749.0736	25.0000	0.0388	0.9971
C(Nome_da_Grande_Região)	7950343159.7408	4.0000	0.0388	0.9971
Residual	3243179687725583.0000	63389.0000	NaN	NaN

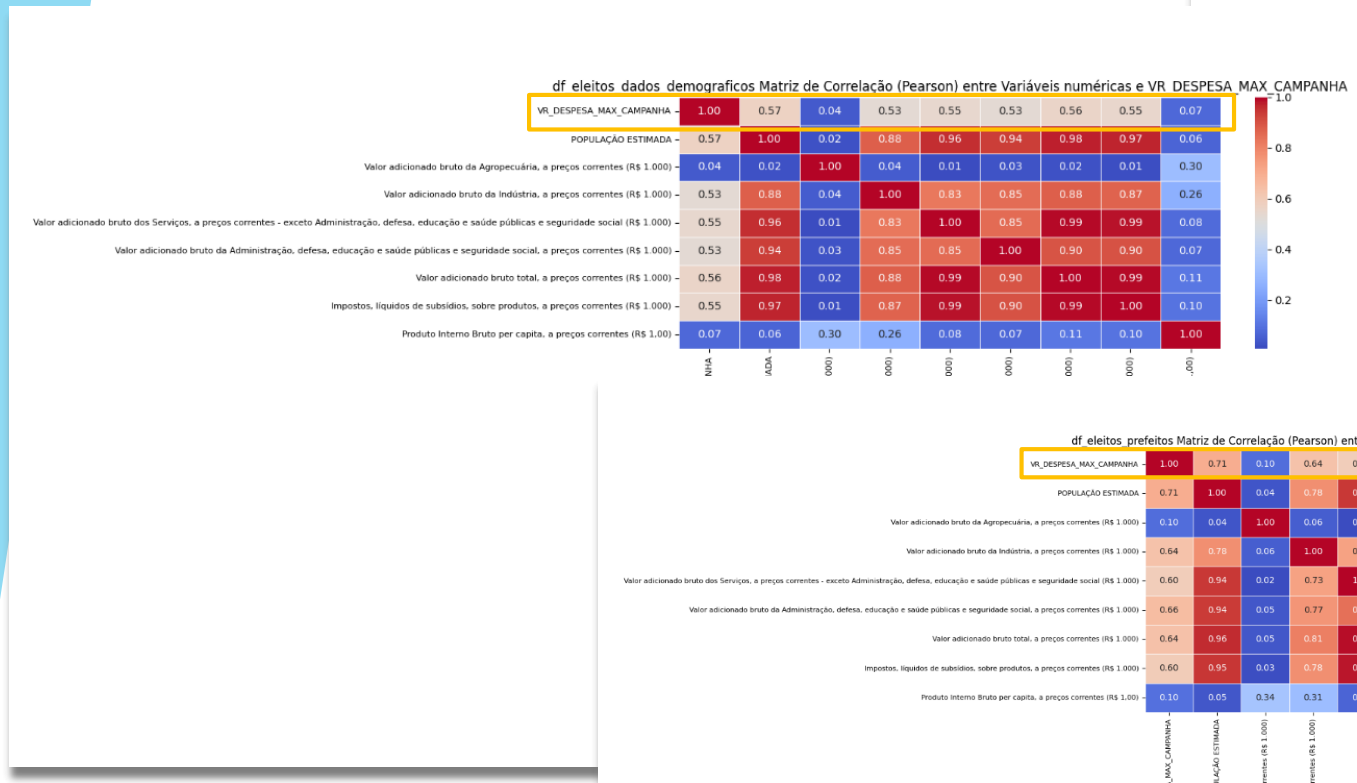
ANOVA para df_vereadores

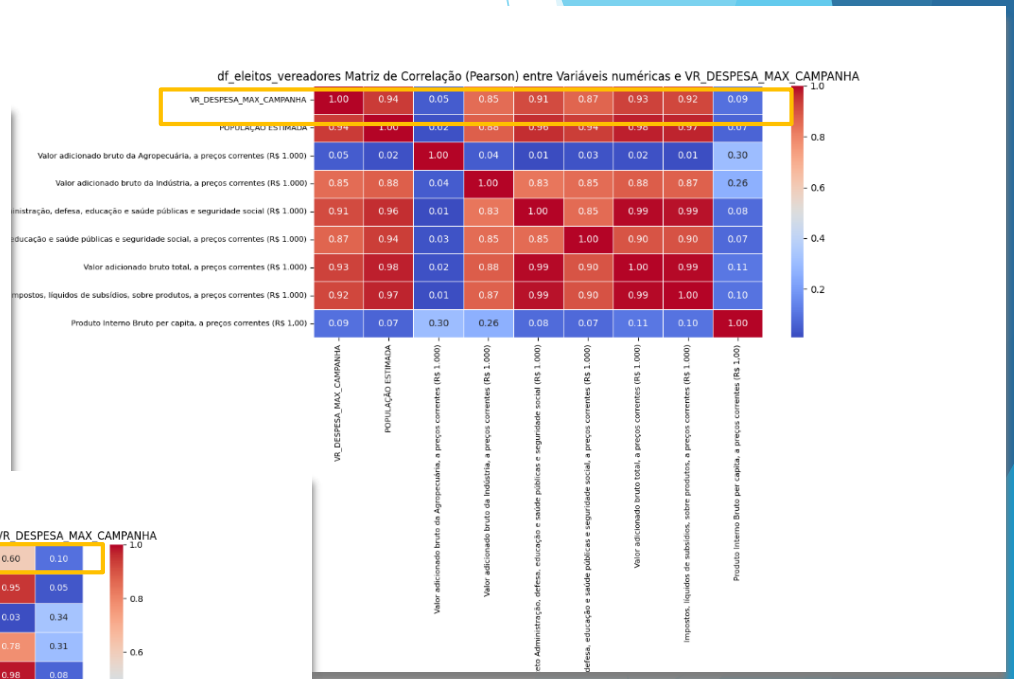
Resultados da Análise de Variância (ANOVA) para as colunas: DS_CARGO, TP_AGREMIACAO, SG_PARTIDO, NM_PARTIDO, ST_REELEICAO, capital, UF, Nome_da_Grande_Região

	sum_sq	df	F	PR(>F)
C(DS_CARGO)	267123218.3903	1.0000	0.0215	0.8836
C(TP_AGREMIACAO)	210413192.7098	1.0000	0.0169	0.8966
C(SG_PARTIDO)	45577407456.1106	28.0000	0.1307	1.0000
C(NM_PARTIDO)	45577407487.2736	28.0000	0.1307	1.0000
C(ST_REELEICAO)	54885941079.2792	1.0000	4.4078	0.0358
C(capital)	406064912374621.5000	1.0000	32610.2576	0.0000
C(UF)	50915599212.2717	25.0000	0.1636	0.9569
C(Nome_da_Grande_Região)	8146495874.0437	4.0000	0.1636	0.9569
Residual	720961871636022.1250	57899.0000	NaN	NaN

ANÁLISE E EXPLORAÇÃO DOS DADOS

ANÁLISE DE VARIÁVEIS NUMÉRICAS





ANÁLISE E EXPLORAÇÃO DOS DADOS

ESTATÍSTICAS DA COLUNA VR_DESPESA_MAX_CAMPANHA – VARIÁVEL ALVO

```
Estatisticas para VR_DESPESA_MAX_CAMPANHA em df_eleitos_dados_demograficos:
count      63448.0000
mean       58129.7489
std       256943.2441
min        12307.7500
25%        12307.7500
50%        12307.7500
75%        36847.9700
max     30413484.3800
Name: VR_DESPESA_MAX_CAMPANHA, dtype: float64
Estatisticas para VR_DESPESA_MAX_CAMPANHA em df_prefeitos:
count       5492.0000
mean      257676.0976
std       708644.7012
min       101190.8700
25%       123077.4200
50%       123077.4200
75%       189702.3725
max     30413484.3800
Name: VR_DESPESA_MAX_CAMPANHA, dtype: float64
Estatisticas para VR_DESPESA_MAX_CAMPANHA em df_vereadores:
count      57956.0000
mean       39220.4290
std       143408.1831
min        12307.7500
25%        12307.7500
50%        12307.7500
75%        26215.4800
max      3675197.1200
Name:                      VR_DESPESA_MAX_CAMPANHA,                      dtype:
```

Alguns insights e hipóteses que podem ser levantados a partir desses dados são:

- A diferença nas despesas máximas de campanha entre prefeitos e vereadores é significativa. Isso pode estar relacionado à maior visibilidade e custos associados às campanhas para o cargo de prefeito.

- A presença de valores extremamente altos em ambos os DataFrames sugere a presença de outliers, que podem ser investigados para determinar sua origem e relevância.

- A alta variabilidade nas despesas de campanha, como indicado pelo desvio padrão, pode ser explorada em relação a variáveis demográficas, partidárias ou geográficas para entender melhor os fatores que influenciam os gastos de campanha.

CRIAÇÃO DE MODELOS DE MACHINE LEARNING

GRADIENT BOOSTING REGRESSOR

Motivações de utiliza-lo no contexto:

Desempenho em problemas de regressão:
O Gradient Boosting Regressor é um algoritmo de ensemble que tem se mostrado eficaz em problemas de regressão, especialmente quando há interações complexas entre as variáveis independentes.

Lida bem com outliers: O Gradient Boosting Regressor é robusto em relação a outliers devido à sua natureza baseada em árvores. Árvores de decisão podem capturar padrões não lineares e são menos sensíveis a valores extremos em comparação com modelos lineares.

Flexibilidade:
Ele é flexível o suficiente para lidar com uma combinação de variáveis numéricas e categóricas, o que é relevante para o conjunto de dados estudado, que inclui ambas as categorias.

Melhora gradualmente o desempenho:
O algoritmo funciona construindo árvores de decisão sequencialmente, corrigindo os erros dos modelos anteriores. Isso permite que ele melhore gradualmente o desempenho, tornando-o adequado para ajustar-se a complexidades crescentes nos dados.

CRIAÇÃO DE MODELOS DE MACHINE LEARNING

analise_de_modelo_VR_DESPESA_MAX_CAMPANHA ()

```python
def analise_de_modelo_VR_DESPESA_MAX_CAMPANHA(df,nomedf,analise_resisduos):

    # Seleciona as colunas de recursos numéricos
    X_colunas_numericas = df[[
        'VR_DESPESA_MAX_CAMPANHA',
        'POPULAÇÃO ESTIMADA',
        'Valor adicionado bruto da Agropecuária, a preços correntes (R$ 1.000)',
        'Valor adicionado bruto da Indústria, a preços correntes (R$ 1.000)',
        'Valor adicionado bruto dos Serviços, a preços correntes - exceto Administração, defesa, educação e saúde públicas e seguridade social (R$ 1.000)',
        'Valor adicionado bruto da Administração, defesa, educação e saúde públicas e seguridade social, a preços correntes (R$ 1.000)',
        'Valor adicionado bruto total, a preços correntes (R$ 1.000)',
        'Impostos, líquidos de subsídios, sobre produtos, a preços correntes (R$ 1.000)',
        'Produto Interno Bruto per capita, a preços correntes (R$ 1,00)']]

    # Cria variáveis dummy para as colunas categóricas
    X_colunas_categoricas = pd.get_dummies(df[[
        'DS_CARGO',
        'TP_AGREMIACAO',
        'SG_PARTIDO',
        'ST_REELEICAO',
        'capital',
        'UF',
        'Nome_da_Grande_Região'


    ]], drop_first=True)

    # Combina as variáveis numéricas e categóricas codificadas
    X = pd.concat([X_colunas_numericas, X_colunas_categoricas], axis=1)


    # Define a variável alvo
    y = df['VR_DESPESA_MAX_CAMPANHA']

    # Divide o conjunto de dados em treinamento e teste
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    # Treina o modelo e obtém os melhores parâmetros
    modelo = calcular_melhores_parametros(X_train,y_train)

    # Ajusta o modelo aos dados de treinamento
    modelo[0].fit(X_train, y_train)
```

CRIAÇÃO DE MODELOS DE MACHINE LEARNING

calcular_melhores_parâmetros() - GridSearchCV

```python
def calcular_melhores_parametros(X,y):
    # Defina os hiperparâmetros que você deseja otimizar
    param_grid = {
        'n_estimators': [200,300],
        'learning_rate': [0.1],
        'max_depth': [4,6]
    }

    # Crie uma instância do regressor GradientBoosting
    regressor = GradientBoostingRegressor()

    # Crie um objeto GridSearchCV
    grid_search = GridSearchCV(estimator=regressor, param_grid=param_grid, cv=2, n_jobs=-1,verbose=4)

    # Realize a pesquisa em grade
    grid_search.fit(X, y)

    # Exiba os melhores hiperparâmetros encontrados
    print("Melhores hiperparâmetros encontrados:")
    print(grid_search.best_params_)

    # Exiba a melhor pontuação do modelo
    print("Melhor pontuação do modelo:")
    print(grid_search.best_score_)

    # Ajuste o modelo final com os melhores hiperparâmetros
    melhor_modelo = grid_search.best_estimator_
    return [melhor_modelo]
```

CRIAÇÃO DE MODELOS DE MACHINE LEARNING

remover_outliers_VR_DESPESA_MAX_CAMPANHA() – regra 3-sigma)

```python
def remover_outliers_VR_DESPESA_MAX_CAMPANHA(df,nome_df):
    # Calcula a média e o desvio padrão da coluna 'VR_DESPESA_MAX_CAMPANHA'
    media_despesa = df['VR_DESPESA_MAX_CAMPANHA'].mean()
    desvio_padrao_despesa = df['VR_DESPESA_MAX_CAMPANHA'].std()

    # Define o limiar para identificar outliers usando a regra 3-sigma
    limiar_superior = media_despesa + 3 * desvio_padrao_despesa
    limiar_inferior = media_despesa - 3 * desvio_padrao_despesa

    # Filtra os outliers em um novo DataFrame
    df_outliers = df[(df['VR_DESPESA_MAX_CAMPANHA'] < limiar_inferior) |
                     (df['VR_DESPESA_MAX_CAMPANHA'] > limiar_superior)]

    # Filtra os dados originais removendo os outliers
    df = df[(df['VR_DESPESA_MAX_CAMPANHA'] >= limiar_inferior) &
            (df['VR_DESPESA_MAX_CAMPANHA'] <= limiar_superior)]


    # Exiba informações sobre os outliers
    print(f"Número de outliers encontrados em {nome_df}: {len(df_outliers['VR_DESPESA_MAX_CAMPANHA'])}")
    print(f"Percentagem de outliers em {nome_df}:"
          f" {len(df_outliers['VR_DESPESA_MAX_CAMPANHA'])  / len(df['VR_DESPESA_MAX_CAMPANHA']) * 100:.2f}%")

    # Exibe estatísticas descritivas após remover os outliers
    print("Estatísticas descritivas após remover os outliers:")
    print(df['VR_DESPESA_MAX_CAMPANHA'].describe())

    # O DataFrame df agora contém os dados sem outliers e df_outliers apenas os outliers
    return [df,df_outliers]
```

Os outliers são considerados os valores que estão a mais de 3 desvios padrão da média.

CRIAÇÃO DE MODELOS DE MACHINE LEARNING

ANÁLISE DO MODELO DE REGRESSÃO

A parte final do código relacionada à Criação de Modelos de Machine Learning a aplica essas funções a três conjuntos de dados diferentes: df_eleitos_dados_demograficos, df_eleitos_prefeitos, e df_eleitos_vereadores.

Primeiro com outliers e, em seguida, sem outliers.

INTERPRETAÇÃO E APRESENTAÇÃO DOS RESULTADOS

ANÁLISE DO MODELO DE REGRESSÃO E OUTLIERS - df_eleitos_dados_demograficos

Algumas conclusões importantes:

- Em todos os cenários, os modelos de regressão mostraram um desempenho muito bom, com altos valores de coeficiente de determinação (R^2), indicando uma boa capacidade de previsão.

- A remoção de outliers teve um impacto significativo nos resultados. Sem outliers, os modelos obtiveram pontuações R^2 próximas a 1, indicando um ajuste quase perfeito aos dados.

- No cenário de análise conjunta de todos os dados (df_eleitos_dados_demograficos), o modelo com outliers ainda teve um desempenho sólido, mas a presença de outliers aumentou o erro médio absoluto (MAE) e o erro quadrático médio (MSE).

- **A separação dos dados em prefeitos (df_eleitos_prefeitos) e vereadores (df_eleitos_vereadores) mostrou que os modelos tiveram um melhor desempenho quando aplicados separadamente a cada grupo, tanto com quanto sem outliers.**

INTERPRETAÇÃO E APRESENTAÇÃO DOS RESULTADOS

ANÁLISE DO MODELO DE REGRESSÃO E OUTLIERS - df_eleitos_prefeitos

Conclusão:

A análise demonstra que a remoção de outliers teve um impacto significativo na qualidade do modelo de regressão para previsão do valor máximo de despesas de campanha.

O modelo sem outliers obteve resultados muito melhores em termos de precisão, ajuste aos dados e capacidade de explicar a variabilidade. Portanto, ao criar um modelo de previsão para esse cenário específico, **é altamente recomendável remover os outliers do conjunto de dados para obter previsões mais confiáveis e precisas**.

INTERPRETAÇÃO E APRESENTAÇÃO DOS RESULTADOS

ANÁLISE DO MODELO DE REGRESSÃO E OUTLIERS - df_eleitos_vereadores

Conclusão:

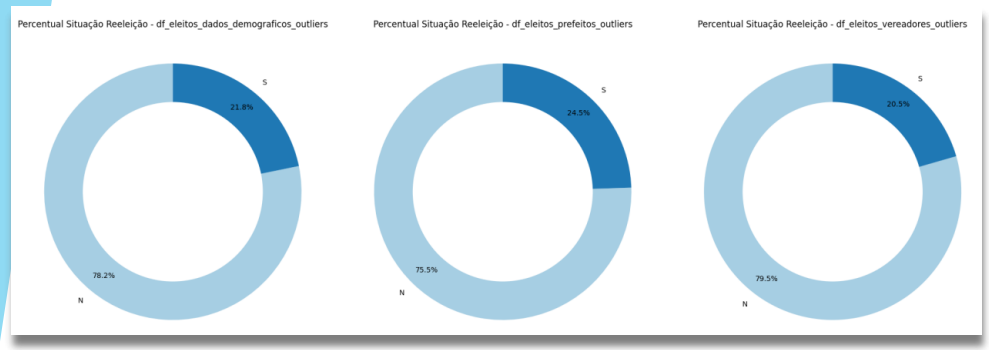
Comparando os modelos, **é evidente que a remoção dos outliers resultou em um modelo de regressão mais preciso, com menor MSE, MAE e RMSE**.

No entanto, é importante ressaltar que **ambos os modelos têm um desempenho geral excelente, com pontuações R^2 muito próximas de 1**, o que significa que eles explicam muito bem a variabilidade na variável alvo 'VR_DESPESA_MAX_CAMPANHA'.
Portanto, a remoção de outliers parece ter melhorado a capacidade de generalização do modelo.

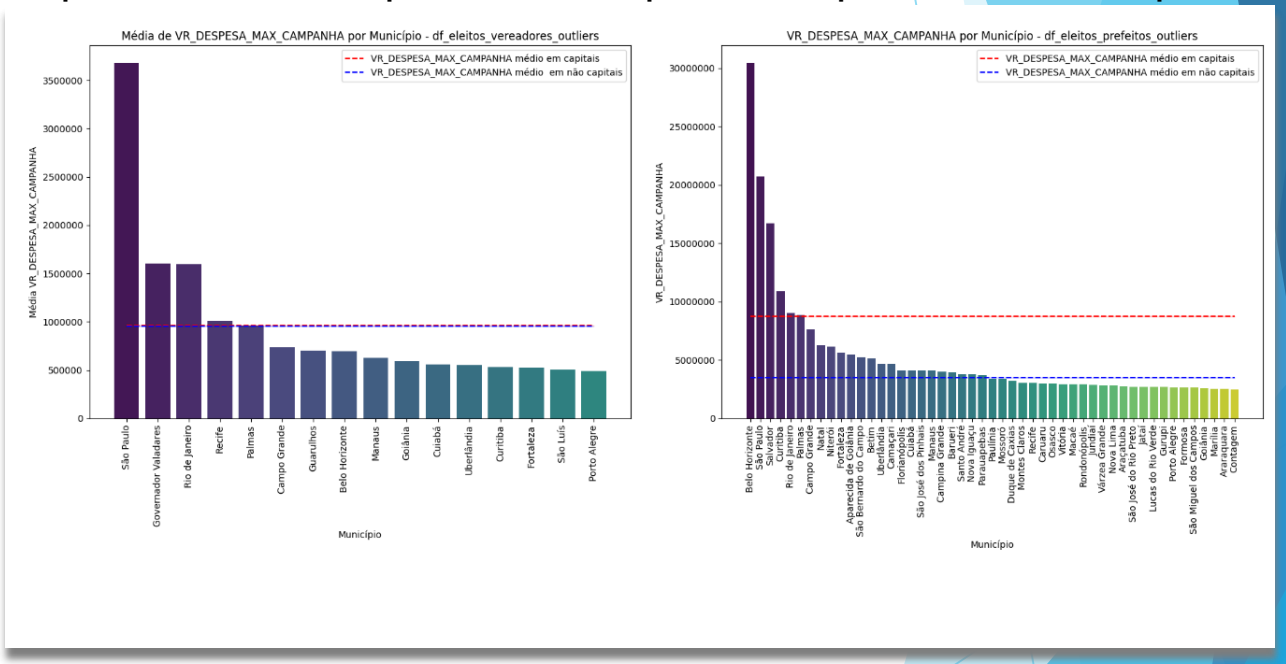
INTERPRETAÇÃO E APRESENTAÇÃO DOS RESULTADOS

ANÁLISE DOS OUTLIERS

Análise da Relação entre Reeleição e Outliers



Disparidade nas Despesas de Campanha: Capitais vs. Não Capitais



Analises:

Análise da Relação entre Reeleição e Outliers;
Distribuição dos Eleitos por Cargo e Agremiação ;
Disparidade nas Despesas de Campanha: Capitais vs. Não Capitais;
Relação Entre Média de Despesas de Campanha, PIB e UF;
Variação das Despesas de Campanha e População Estimada por UF;
Variação nas Médias de Gastos de Campanha por Partido;
Variação nas Despesas de Campanha de Candidatos Eleitos por Estado e Região.

INTERPRETAÇÃO E APRESENTAÇÃO DOS RESULTADOS

CONSIDERAÇÕES FINAIS

Contextualização Significativa: A contextualização inicial foi essencial para compreender a relevância desse estudo, considerando o contexto desafiador das eleições municipais de 2020 no Brasil, marcadas pela pandemia de COVID-19 e pela mudança no financiamento das campanhas políticas.

Impacto dos Outliers: O estudo demonstrou que a presença de outliers nos dados de despesas de campanha teve um impacto significativo nos modelos de regressão. A remoção desses valores atípicos resultou em modelos mais precisos e confiáveis, com melhor capacidade de previsão.

Modelos de Regressão Sólidos: Independentemente da presença de outliers, os modelos de regressão utilizados neste estudo mostraram um desempenho geral excelente, com altos valores de coeficiente de determinação (R^2), indicando uma boa capacidade de explicar a variabilidade nos gastos de campanha.

Influência de Fatores Econômicos e Demográficos: Ficou claro que fatores econômicos, representados pelo PIB municipal, e demográficos, como o porte populacional dos municípios, desempenharam um papel significativo na determinação das estratégias de campanha e dos resultados eleitorais em 2020.

Diferenças Regionais e Partidárias: O estudo revelou variações significativas nas despesas de campanha entre diferentes estados, regiões e partidos políticos. Essas diferenças podem ser valiosas para orientar estratégias políticas futuras e alocação de recursos de campanha.

Recomendações para Modelagem:
Com base nas análises, uma recomendação importante é a remoção de outliers ao criar modelos de previsão para gastos de campanha.
Isso pode melhorar a qualidade e a confiabilidade das previsões.

Contribuição para a Compreensão Eleitoral: O estudo contribuiu significativamente para a compreensão da dinâmica política brasileira, ao mostrar como fatores econômicos, demográficos e a presença de outliers afetaram as eleições municipais de 2020. Essas informações são valiosas para futuros estudos e estratégias políticas.

Recomendações Futuras: Para pesquisas futuras, seria interessante explorar mais a fundo as causas por trás das diferenças regionais e partidárias nas despesas de campanha, bem como considerar outros fatores que possam influenciar os resultados eleitorais.