

Importação das bibliotecas

```
In [464... import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
#Biblioteca para ignorar mensagens de warning (aviso) ao rodar uma célula de código
import warnings
warnings.filterwarnings('ignore')
```

```
In [465... # Bibliotecas de machine Learning
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from sklearn.preprocessing import LabelEncoder

# Métricas de avaliação dos modelos
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
```

Carregando os dados

```
In [466... # importação da base de dados
dados = pd.read_csv('precos_carros_brasil.csv')
#exibindo as primeiras linhas
dados.head()
```

Out[466...

	year_of_reference	month_of_reference	fipe_code	authentication	brand	model	fuel	gear	engine_size	year_mod
0	2021.0	January	004001-0	cfzlcztzfwrcp	GM - Chevrolet	Corsa Wind 1.0 MPFI / EFI 2p	Gasoline	manual	1	2002
1	2021.0	January	004001-0	cdqwxwpw3y2p	GM - Chevrolet	Corsa Wind 1.0 MPFI / EFI 2p	Gasoline	manual	1	2001
2	2021.0	January	004001-0	cb1t3xwwj1xp	GM - Chevrolet	Corsa Wind 1.0 MPFI / EFI 2p	Gasoline	manual	1	2000
3	2021.0	January	004001-0	cb9gct6j65r0	GM - Chevrolet	Corsa Wind 1.0 MPFI / EFI 2p	Alcohol	manual	1	2000
4	2021.0	January	004003-7	g15wg0gbz1fx	GM - Chevrolet	Corsa Pick-Up GL/Champ 1.6 MPFI / EFI	Gasoline	manual	1,6	2001

Remover a coluna "authentication" pôs este dado não será relevante, uma vez que ele é usado para autenticação no site da tabela File

```
In [467... dados.drop('authentication', axis=1, inplace=True)
dados.head()
```

Out[467...

	year_of_reference	month_of_reference	fipe_code	brand	model	fuel	gear	engine_size	year_model	avg_price_brl
0	2021.0	January	004001-0	GM - Chevrolet	Corsa Wind 1.0 MPFI / EFI 2p	Gasoline	manual	1	2002.0	9162.0
1	2021.0	January	004001-0	GM - Chevrolet	Corsa Wind 1.0 MPFI / EFI 2p	Gasoline	manual	1	2001.0	8832.0
2	2021.0	January	004001-0	GM - Chevrolet	Corsa Wind 1.0 MPFI / EFI 2p	Gasoline	manual	1	2000.0	8388.0
3	2021.0	January	004001-0	GM - Chevrolet	Corsa Wind 1.0 MPFI / EFI 2p	Alcohol	manual	1	2000.0	8453.0
4	2021.0	January	004003-7	GM - Chevrolet	Corsa Pick-Up GL/Champ 1.6 MPFI / EFI	Gasoline	manual	1,6	2001.0	12525.0

Verificar se ha dados invalidas

In [468...

```
### Verificar se há valores nulos
dados.isna()
```

Out[468...

	year_of_reference	month_of_reference	fipe_code	brand	model	fuel	gear	engine_size	year_model	avg_price_brl
0	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False
...
267537	True	True	True	True	True	True	True	True	True	True
267538	True	True	True	True	True	True	True	True	True	True
267539	True	True	True	True	True	True	True	True	True	True
267540	True	True	True	True	True	True	True	True	True	True
267541	True	True	True	True	True	True	True	True	True	True

267542 rows × 10 columns

Identificadas muitas linhas sem dados, estas linhas serão removidas.

In [469...

```
# Removendo linhas vazias
dados = dados.dropna()
# Verificando se existema mais alguma linha faltando dados
dados.isna().sum()
```

Out[469...

```
year_of_reference      0
month_of_reference     0
fipe_code              0
brand                 0
model                 0
fuel                 0
gear                 0
engine_size           0
year_model            0
avg_price_brl         0
dtype: int64
```

In [470...

```
dados.head()
```

Out[470...

	year_of_reference	month_of_reference	fipe_code	brand	model	fuel	gear	engine_size	year_model	avg_price_brl
0	2021.0	January	004001-0	GM - Chevrolet	Corsa Wind 1.0 MPFI / EFI 2p	Gasoline	manual	1	2002.0	9162.0
1	2021.0	January	004001-0	GM - Chevrolet	Corsa Wind 1.0 MPFI / EFI 2p	Gasoline	manual	1	2001.0	8832.0
2	2021.0	January	004001-0	GM - Chevrolet	Corsa Wind 1.0 MPFI / EFI 2p	Gasoline	manual	1	2000.0	8388.0
3	2021.0	January	004001-0	GM - Chevrolet	Corsa Wind 1.0 MPFI / EFI 2p	Alcohol	manual	1	2000.0	8453.0
4	2021.0	January	004003-7	GM - Chevrolet	Corsa Pick-Up GL/Champ 1.6 MPFI / EFI	Gasoline	manual	1,6	2001.0	12525.0

Verificando se há dados duplicados

In [471...

```
dados.duplicated().any()
```

Out[471...

np.True_

Removendo dados duplicados

In [472...

```
dados.drop_duplicates(inplace=True)
dados.duplicated().any()
```

Out[472...

np.False_

Separando colunas numéricas de categóricas

In [473...

```
dados.dtypes
numericas = dados.select_dtypes(include=['float64', 'int64'])
categoricas = dados.select_dtypes(include=['object'])
```

Resumo das variáveis numéricas

In [474...

```
numericas.describe()
```

Out[474...

	year_of_reference	year_model	avg_price_brl
count	202295.000000	202295.000000	202295.000000
mean	2021.564695	2011.271514	52756.765713
std	0.571904	6.376241	51628.912116
min	2021.000000	2000.000000	6647.000000
25%	2021.000000	2006.000000	22855.000000
50%	2022.000000	2012.000000	38027.000000
75%	2022.000000	2016.000000	64064.000000
max	2023.000000	2023.000000	979358.000000

Resumo das variáveis categóricas

In [475...

```
categoricas.describe()
```

Out[475...

	month_of_reference	fipe_code	brand	model	fuel	gear	engine_size
count	202295	202295	202295	202295	202295	202295	202295
unique	12	2091	6	2112	3	2	29
top	January	001216-5	Fiat	Palio Week. Adv/Adv TRYON 1.8 mpi Flex	Gasoline	manual	1,6
freq	24260	425	44962	425	168684	161883	47420

Imprimindo a contagem de valores por modelo e marca do carro

In [476...

```
dados[['brand','model']].value_counts()
```

Out[476...

brand	model	
Fiat	Palio Week. Adv/Adv TRYON 1.8 mpi Flex	425
Ford	Focus 1.6 S/SE/SE Plus Flex 8V/16V 5p	425
VW - Volkswagen	Saveiro 1.6 Mi/ 1.6 Mi Total Flex 8V	400
Ford	Focus 2.0 16V/SE/SE Plus Flex 5p Aut.	400
GM - Chevrolet	Corvette 5.7/ 6.0, 6.2 Targa/Stingray	375
...		
VW - Volkswagen	Polo Track 1.0 Flex 12V 5p	2
Renault	STEPWAY Zen Flex 1.0 12V Mec.	2
VW - Volkswagen	Saveiro Robust 1.6 Total Flex 16V CD	2
	Gol Last Edition 1.0 Flex 12V 5p	2
Nissan	KICKS Active 1.6 16V Flex Aut.	2
Name: count, Length: 2112, dtype: int64		

Conclusão:

Após a análise exploratória, removemos registros vazios e duplicados, resultando em um conjunto de dados limpo e consistente. As colunas numéricas e categóricas foram separadas para facilitar a análise. Observamos que a base de dados contém informações detalhadas sobre preços de carros no Brasil, com uma variedade de marcas e modelos.

Gráfico da distribuição da quantidade de carros por marca

In [477...

```
marcas = dados['brand'].value_counts().sort_values(ascending=False)
# Gráfico da distribuição da quantidade de carros por marca
plt.figure(figsize=(20,10)) # Aumentar o tamanho da figura na saída dos dados
grafico_qtd_branch = plt.bar(marcas.index, marcas.values) # Variavel Nivel Ensino no eixo X
plt.title('Distribuição da quantidade de carros por marca') # plt.title para inserir título no gráfico
plt.ylabel('Total de carros'); # # plt.ylabel para inserir título no gráfico
plt.bar_label(grafico_qtd_branch, size=10);
```

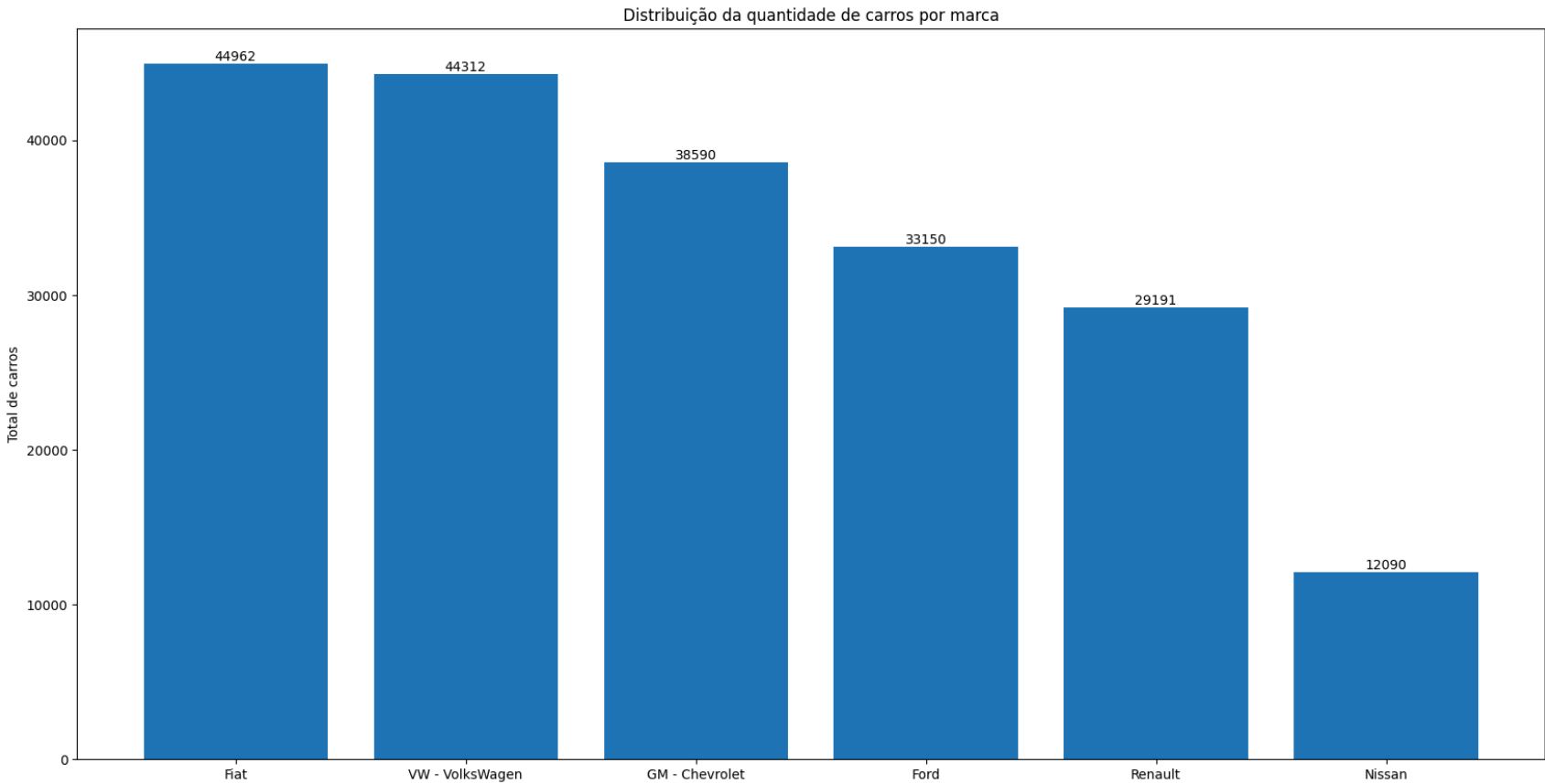


Gráfico da distribuição da quantidade de carros por tipo de engrenagem do carro

In [478...

```
marcas = dados['gear'].value_counts().sort_values(ascending=False)
# Gráfico da distribuição da quantidade de carros por tipo de cambio
plt.figure(figsize=(20,10)) # Aumentar o tamanho da figura na saída dos dados
grafico_engines = plt.bar(marcas.index, marcas.values) # Variavel Nivel Ensino no eixo X
plt.title('Distribuição da quantidade de carros por tipo de cambio') # plt.title para inserir título no gráfico
plt.ylabel('Total de carros'); # # plt.ylabel para inserir título no gráfico
plt.bar_label(grafico_engines, size=10);
```

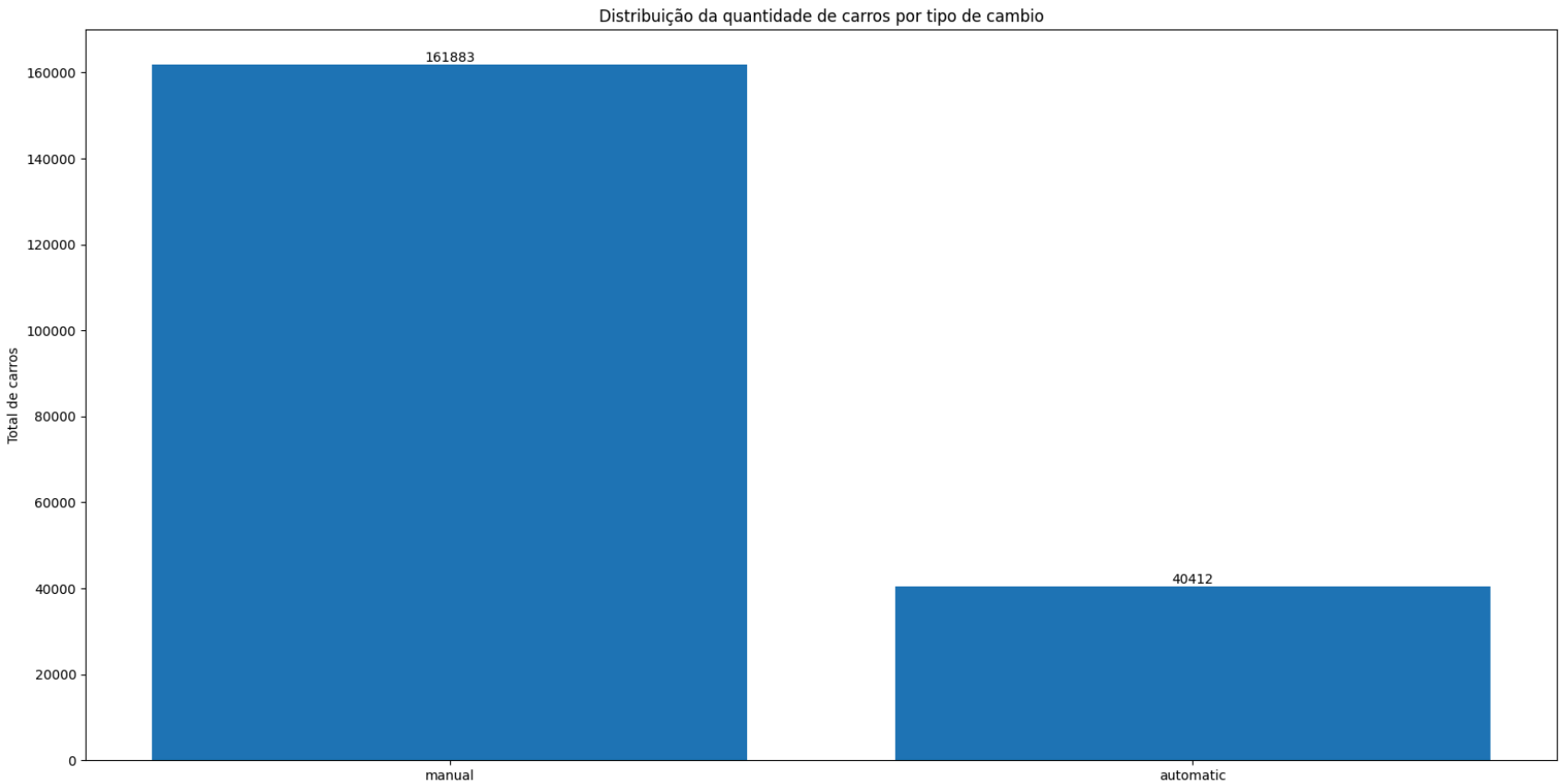


Gráfico da evolução da média de preço dos carros ao longo dos meses de 2022 (variável de tempo no eixo X)

```
In [479... # Criar um dicionário mapeando os meses para números
meses_ordem = {
    'January': 1, 'February': 2, 'March': 3, 'April': 4,
    'May': 5, 'June': 6, 'July': 7, 'August': 8,
    'September': 9, 'October': 10, 'November': 11, 'December': 12
}
# Adicionar os números correspondentes como índice auxiliar e reordenar corretamente
data_mean_price = (dados.where(dados['year_of_reference'] == 2022).dropna().groupby('month_of_reference')['avg_price_
    .mean().rename_axis('month_of_reference') # Garante que o índice tem nome correto
    .reset_index()) # Transforma em DataFrame para manipular

# Criar uma nova coluna numérica para ordenar corretamente
data_mean_price['month_number'] = data_mean_price['month_of_reference'].map(meses_ordem)

# Ordenar pelos números dos meses e restaurar o índice original
data_mean_price = (data_mean_price.sort_values(by='month_number')
    .drop(columns=['month_number']) # Remover a coluna auxiliar
    .set_index('month_of_reference') # Restaurar índice original
)
data_mean_price
```

Out[479...

	avg_price_brl
month_of_reference	
January	54840.270037
February	55824.519882
March	56848.951914
April	57150.037325
May	57799.763776
June	58065.611398
July	57893.997056
August	57923.544105
September	58198.936989
October	58227.410144
November	58215.626236
December	57997.243992

```
In [480... # Gráfico da distribuição da evolução média dos preços dos carros por mês no ano de 2022
plt.figure(figsize=(20,10)) # Aumentar o tamanho da figura

# Criar o gráfico de linha temporal
plt.plot(data_mean_price.index, data_mean_price.values, marker='o', linestyle='--', color='b')

# Adicionar título e rótulos
plt.title('Distribuição da quantidade de carros por tipo de câmbio')
plt.xlabel('Data')
plt.ylabel('Total de carros')

# Melhorar a visualização do eixo X (caso o índice seja uma data)
plt.xticks(rotation=45)
plt.show()
```

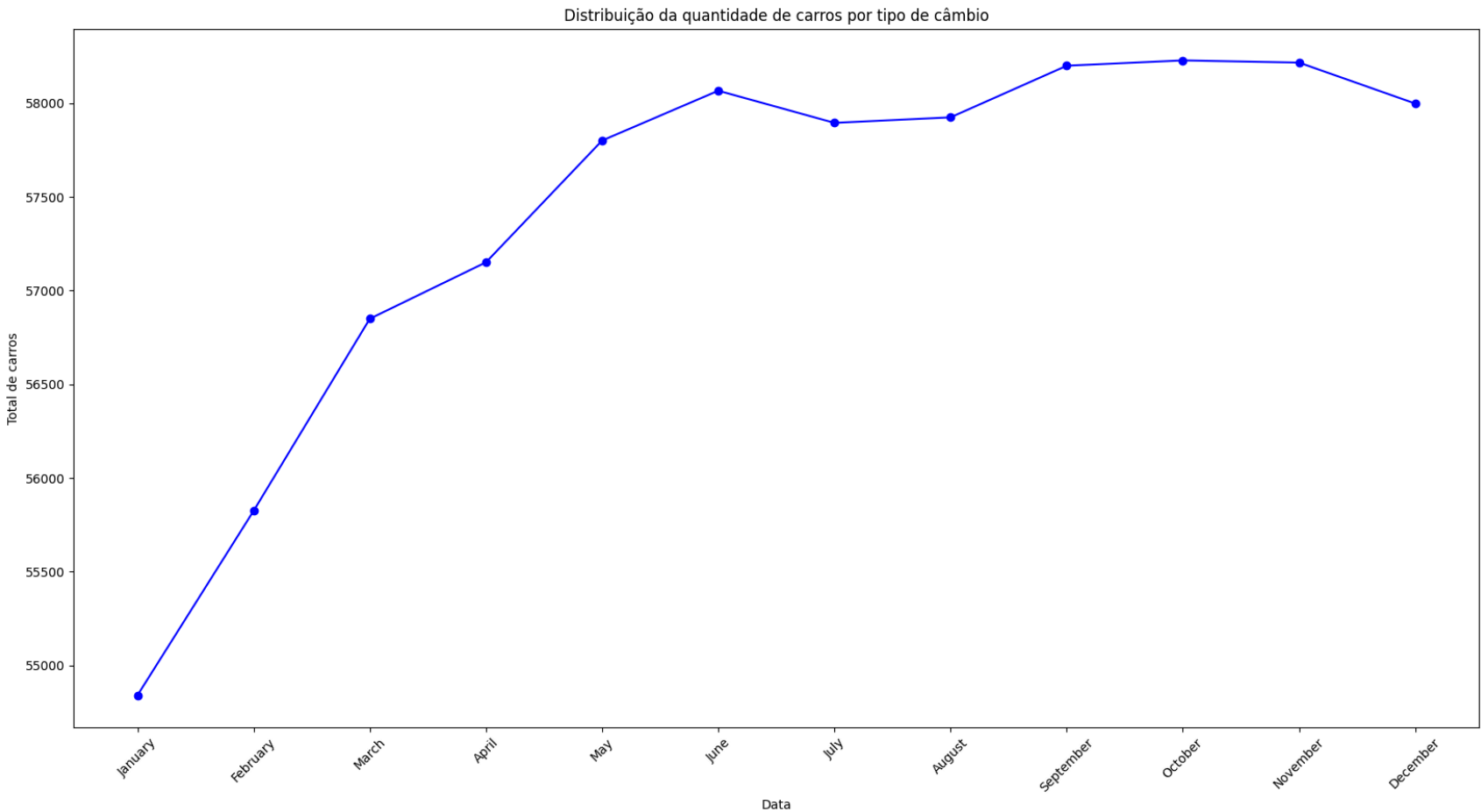


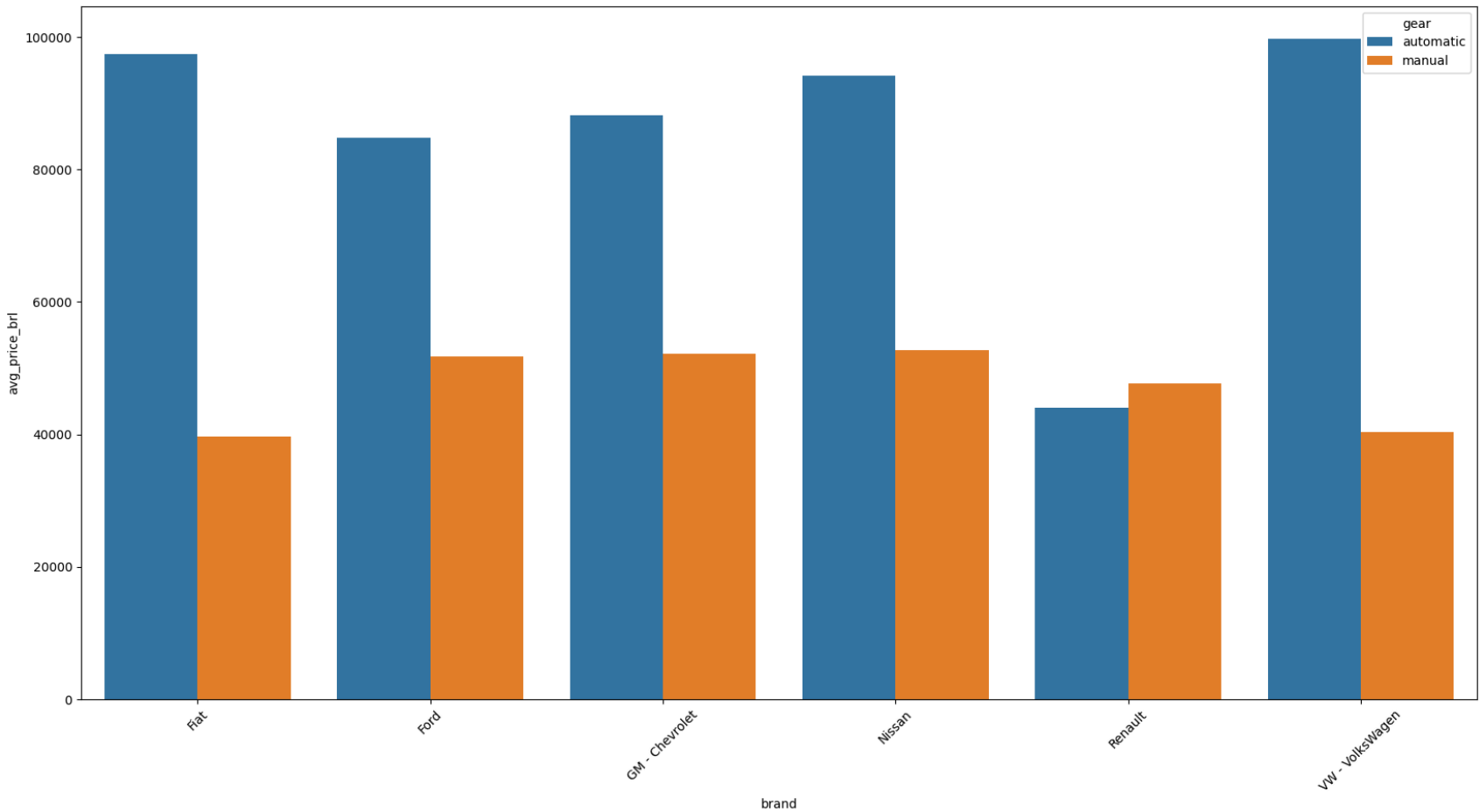
Gráfico da distribuição da média de preço dos carros por marca e tipo de engrenagem

```
In [481... # extração de dados
data_price_fuel = dados.groupby(['brand', 'gear'])['avg_price_brl'].mean().round(2)
data_price_fuel = pd.DataFrame(data_price_fuel).reset_index()
data_price_fuel = data_price_fuel.sort_values(by='brand')
data_price_fuel.head()
```

Out[481...

	brand	gear	avg_price_brl
0	Fiat	automatic	97396.80
1	Fiat	manual	39694.44
2	Ford	automatic	84769.11
3	Ford	manual	51784.33
4	GM - Chevrolet	automatic	88156.92

```
In [482... # Gráfico da distribuição da média de preços dos carros por marca e tipo de cambio
plt.figure(figsize=(20,10))
sns.barplot(x='brand', y='avg_price_brl', hue='gear', data=data_price_fuel, hue_order=['automatic', 'manual'])
plt.xticks(rotation=45) # Rotaciona os rótulos para melhor visualização
plt.show()
```



O gráfico da distribuição da média de preço dos carros por marca e tipo de engrenagem mostra a variação dos preços médios entre diferentes marcas e tipos de câmbio (manual e automático). Observa-se que, em geral, carros com câmbio automático tendem a ter preços médios mais elevados em comparação aos carros com câmbio manual. A marca VW - VolksWagen apresenta a maior diferença de preço entre os tipos de câmbio.

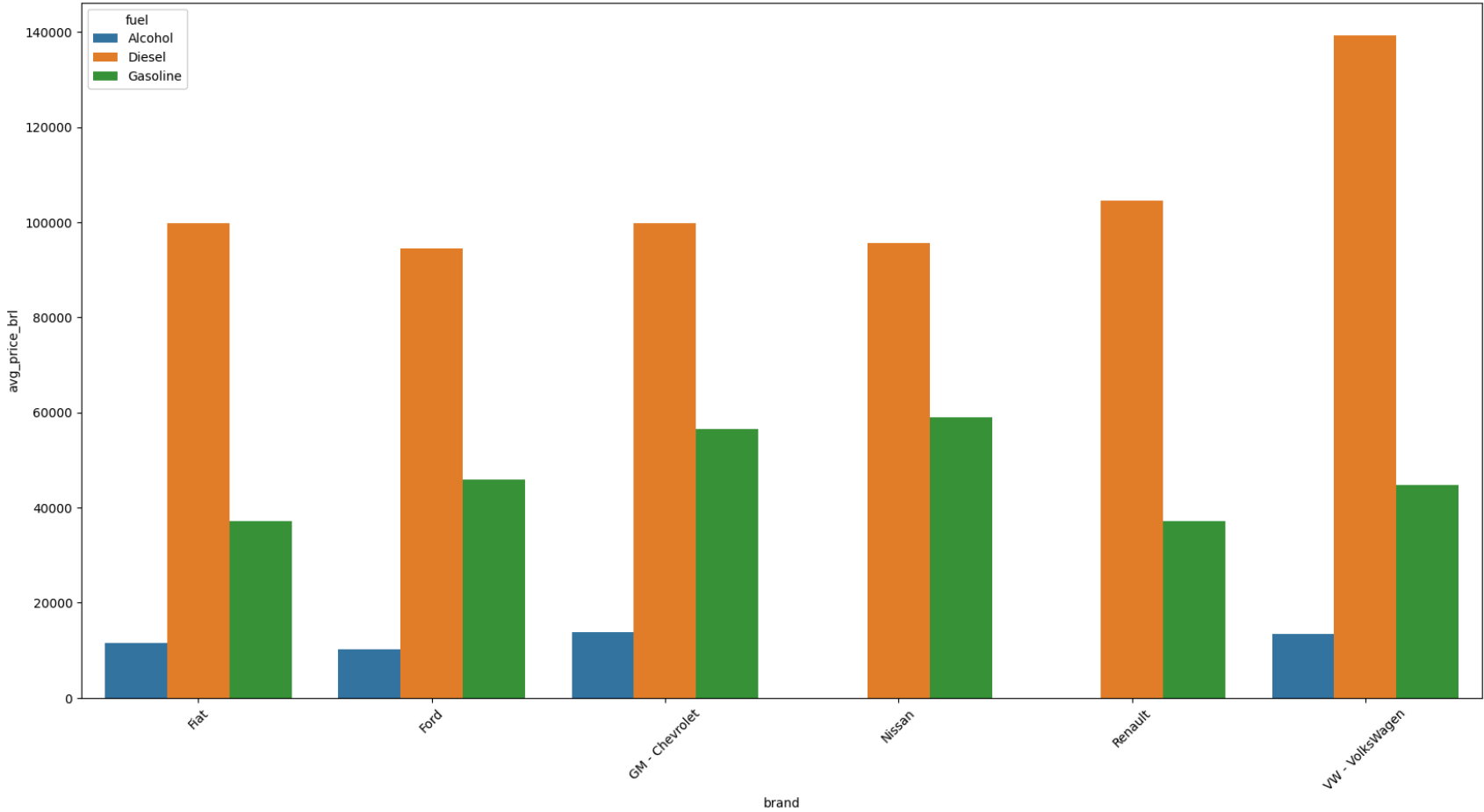
Gráfico da distribuição da média de preço dos carros por marca e tipo de combustível

```
In [483... data_price_fuel = dados.groupby(['brand', 'fuel'])['avg_price_brl'].mean().round(2)
data_price_fuel = pd.DataFrame(data_price_fuel).reset_index()
data_price_fuel = data_price_fuel.sort_values(by='brand')
data_price_fuel.head()
```

Out[483...

	brand	fuel	avg_price_brl
0	Fiat	Alcohol	11509.51
1	Fiat	Diesel	99814.45
2	Fiat	Gasoline	37197.29
3	Ford	Alcohol	10148.91
4	Ford	Diesel	94525.53

```
In [484... plt.figure(figsize=(20,10))
sns.barplot(x='brand', y='avg_price_brl', hue='fuel', data=data_price_fuel, hue_order=['Alcohol', 'Diesel', 'Gasoline'])
plt.xticks(rotation=45) # Rotaciona os rótulos para melhor visualização
plt.show()
```



O gráfico da distribuição da média de preço dos carros por marca e tipo de combustível mostra a variação dos preços médios entre diferentes marcas e tipos de combustível (álcool, diesel e gasolina). Observa-se que, em geral, carros movidos a diesel tendem a ter preços médios mais elevados em comparação aos carros movidos a álcool e gasolina. A marca VW - VolksWagen apresenta a maior diferença de preço entre os tipos de combustível, com os carros a diesel sendo significativamente mais caros. Este gráfico é útil para identificar tendências de preços com base no tipo de combustível utilizado pelos veículos.

Aplicação dos modelos de machine Learning

```
In [485... # Convertendo os valores de ano de referencia para inteiro
dados['year_of_reference'] = dados['year_of_reference'].astype(int)
# Convertendo os valores de ano do modelo para inteiro
dados['year_model'] = dados['year_model'].astype(int)
```

```
In [486... # Converter meses em numeros inteiros e vice-versa
Array_moth = dados['month_of_reference'].unique()
mapa_gear = {mes: i+1 for i, mes in enumerate(Array_moth)}
mapa_gear_inverso = {v: k for k, v in mapa_gear.items()}
# Função para converter mês para número
def moth_para_numero(df, coluna='month_of_reference'):
    df[coluna] = df[coluna].replace(mapa_gear)
    return df

# Função para converter número para mês
def numero_para_month(df, coluna='month_of_reference'):
    df[coluna] = df[coluna].replace(mapa_gear_inverso)
    return df

dados = moth_para_numero(dados)

dados.head()
```

Out[486...

	year_of_reference	month_of_reference	fipe_code	brand	model	fuel	gear	engine_size	year_model	avg_price_brl
0	2021	1	004001-0	GM - Chevrolet	Corsa Wind 1.0 MPFI / EFI 2p	Gasoline	manual	1	2002	9162.0
1	2021	1	004001-0	GM - Chevrolet	Corsa Wind 1.0 MPFI / EFI 2p	Gasoline	manual	1	2001	8832.0
2	2021	1	004001-0	GM - Chevrolet	Corsa Wind 1.0 MPFI / EFI 2p	Gasoline	manual	1	2000	8388.0
3	2021	1	004001-0	GM - Chevrolet	Corsa Wind 1.0 MPFI / EFI 2p	Alcohol	manual	1	2000	8453.0
4	2021	1	004003-7	GM - Chevrolet	Corsa Pick-Up GL/Champ 1.6 MPFI / EFI	Gasoline	manual	1,6	2001	12525.0

In [487...

```
# converter a marca para numeros e vice-versa
Array_brand = dados['brand'].unique()
mapa_brand = {mes: i+1 for i, mes in enumerate(Array_brand)}
mapa_brand_inverso = {v: k for k, v in mapa_brand.items()}
# Função para converter marca para número
def brand_para_numero(df, coluna='brand'):
    df[coluna] = df[coluna].replace(mapa_brand)
    return df

# Função para converter número para marca
def numero_para_brand(df, coluna='brand'):
    df[coluna] = df[coluna].replace(mapa_brand_inverso)
    return df
dados = brand_para_numero(dados)
dados.head()
```

Out[487...

	year_of_reference	month_of_reference	fipe_code	brand	model	fuel	gear	engine_size	year_model	avg_price_brl
0	2021	1	004001-0	1	Corsa Wind 1.0 MPFI / EFI 2p	Gasoline	manual	1	2002	9162.0
1	2021	1	004001-0	1	Corsa Wind 1.0 MPFI / EFI 2p	Gasoline	manual	1	2001	8832.0
2	2021	1	004001-0	1	Corsa Wind 1.0 MPFI / EFI 2p	Gasoline	manual	1	2000	8388.0
3	2021	1	004001-0	1	Corsa Wind 1.0 MPFI / EFI 2p	Alcohol	manual	1	2000	8453.0
4	2021	1	004003-7	1	Corsa Pick-Up GL/Champ 1.6 MPFI / EFI	Gasoline	manual	1,6	2001	12525.0

In [488...

```
# converter a modelo para numeros e vice-versa
Array_model = dados['model'].unique()
mapa_model = {mes: i+1 for i, mes in enumerate(Array_model)}
mapa_model_inverso = {v: k for k, v in mapa_model.items()}
# Função para converter modelo para número
```



```
def model_para_numero(df, coluna='model'):
    df[coluna] = df[coluna].replace(mapa_model)
    return df

# Função para converter número para modelo
def numero_para_model(df, coluna='model'):
    df[coluna] = df[coluna].replace(mapa_model_inverso)
    return df
dados = model_para_numero(dados)
dados.head()
```

Out[488...

	year_of_reference	month_of_reference	fipe_code	brand	model	fuel	gear	engine_size	year_model	avg_price_brl
0	2021	1	004001-0	1	1	Gasoline	manual	1	2002	9162.0
1	2021	1	004001-0	1	1	Gasoline	manual	1	2001	8832.0
2	2021	1	004001-0	1	1	Gasoline	manual	1	2000	8388.0
3	2021	1	004001-0	1	1	Alcohol	manual	1	2000	8453.0
4	2021	1	004003-7	1	2	Gasoline	manual	1,6	2001	12525.0

In [489...

```
# converter a fuel para numeros e vice-versa
Array_fuel = dados['fuel'].unique()
mapa_fuel = {mes: i+1 for i, mes in enumerate(Array_fuel)}
mapa_fuel_inverso = {v: k for k, v in mapa_fuel.items()}
# Função para converter fuel para número
def fuel_para_numero(df, coluna='fuel'):
    df[coluna] = df[coluna].replace(mapa_fuel)
    return df

# Função para converter número para fuel
def numero_para_fuel(df, coluna='fuel'):
    df[coluna] = df[coluna].replace(mapa_fuel_inverso)
    return df
dados = fuel_para_numero(dados)
dados.head()
```

Out[489...

	year_of_reference	month_of_reference	fipe_code	brand	model	fuel	gear	engine_size	year_model	avg_price_brl
0	2021	1	004001-0	1	1	1	manual	1	2002	9162.0
1	2021	1	004001-0	1	1	1	manual	1	2001	8832.0
2	2021	1	004001-0	1	1	1	manual	1	2000	8388.0
3	2021	1	004001-0	1	1	2	manual	1	2000	8453.0
4	2021	1	004003-7	1	2	1	manual	1,6	2001	12525.0

In [490...

```
# converter a gear para numeros e vice-versa
Array_gear = dados['gear'].unique()
mapa_gear = {mes: i+1 for i, mes in enumerate(Array_gear)}
mapa_gear_inverso = {v: k for k, v in mapa_gear.items()}
# Função para converter fuel para número
def gear_para_numero(df, coluna='gear'):
    df[coluna] = df[coluna].replace(mapa_gear)
    return df

# Função para converter número para fuel
def numero_para_gear(df, coluna='gear'):
    df[coluna] = df[coluna].replace(mapa_gear_inverso)
    return df
dados = gear_para_numero(dados)
dados.head()
```

Out[490...

	year_of_reference	month_of_reference	fipe_code	brand	model	fuel	gear	engine_size	year_model	avg_price_brl
0	2021		1	004001-0	1	1	1	1	2002	9162.0
1	2021		1	004001-0	1	1	1	1	2001	8832.0
2	2021		1	004001-0	1	1	1	1	2000	8388.0
3	2021		1	004001-0	1	1	2	1	2000	8453.0
4	2021		1	004003-7	1	2	1	1,6	2001	12525.0

In [491...

```
# verificando se existem valores nulos na coluna engine_size
dados['engine_size'].isna().any()
# Converter engine_size para float
# Verificar se a conversão foi realizada com sucesso
dados['engine_size'] = dados['engine_size'].replace(',', '.', regex=True).astype(float)
dados.dtypes
```

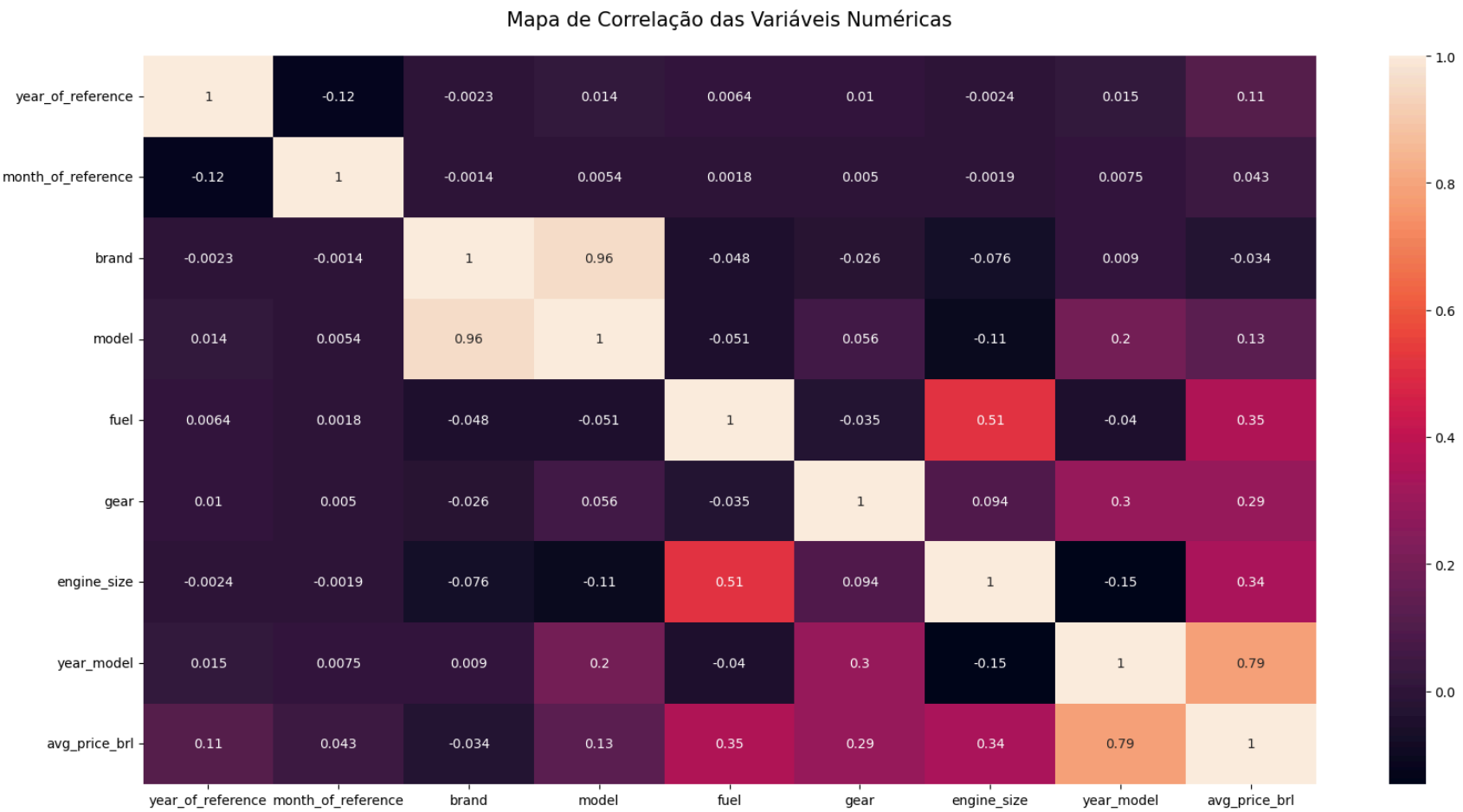
Out[491...
year_of_reference int64
month_of_reference int64
fipe_code object
brand int64
model int64
fuel int64
gear int64
engine_size float64
year_model int64
avg_price_br1 float64
dtype: object

```
In [492...  
# Variável dados_num contém apenas variáveis numéricas de interesse (exclui o restante)  
dados_num = dados.drop(['fipe_code'],axis = 1)  
dados_num.head()
```

Out[492...

	year_of_reference	month_of_reference	brand	model	fuel	gear	engine_size	year_model	avg_price_br1
0	2021		1	1	1	1	1.0	2002	9162.0
1	2021		1	1	1	1	1.0	2001	8832.0
2	2021		1	1	1	1	1.0	2000	8388.0
3	2021		1	1	2	1	1.0	2000	8453.0
4	2021		1	1	2	1	1.6	2001	12525.0

```
In [493...  
# Mapa de correlação das variáveis numéricas com variável Target  
plt.figure(figsize=(20,10))  
sns.heatmap(dados_num.corr("spearman"), annot = True)  
plt.title("Mapa de Correlação das Variáveis Numéricas\n", fontsize = 15)  
plt.show()
```



```
In [494...  
# Variável X contém apenas variáveis numéricas de interesse para a análise, excluindo a variável target  
X = dados_num.drop(['avg_price_br1'],axis = 1)  
X.head()
```

Out[494...

	year_of_reference	month_of_reference	brand	model	fuel	gear	engine_size	year_model
0	2021		1	1	1	1	1.0	2002
1	2021		1	1	1	1	1.0	2001
2	2021		1	1	1	1	1.0	2000
3	2021		1	1	2	1	1.0	2000
4	2021		1	1	2	1	1.6	2001

```
In [495...  
# Variável Y contém apenas a variável target - Faixa Salarial  
Y = dados_num['avg_price_br1']  
Y.head()
```

Out[495...
0 9162.0
1 8832.0
2 8388.0
3 8453.0
4 12525.0
Name: avg_price_br1, dtype: float64

```
In [496... # Divisão: 25% dos dados são de teste e 75% de treinamento
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.25, random_state = 42)
```

Random Forest

```
In [497... # Algoritmo Random Forest, sem especificar nenhum parâmetro (número de árvores, número de ramificações, etc)
model_rf = RandomForestRegressor()
model_rf.fit(X_train, Y_train)
valores_preditos_rf = model_rf.predict(X_test)
```

Analisando a importância das variáveis para estimar a variável target

```
In [498... model_rf.feature_importances_
feature_importances = pd.DataFrame(model_rf.feature_importances_, index = X_train.columns, columns=['importance']).sort_values(
feature_importances
```

Out[498...

	importance
engine_size	0.404537
year_model	0.379045
fuel	0.085556
model	0.070314
gear	0.033996
year_of_reference	0.012227
brand	0.008667
month_of_reference	0.005657

A análise de importância das variáveis revelou que o tamanho do motor (engine_size) e o ano do modelo (year_model) são os fatores mais influentes na determinação do preço médio dos carros (avg_price_br1). Variáveis como tipo de combustível (fuel) e modelo do carro (model) também têm impacto significativo, enquanto a marca (brand) e o mês de referência (month_of_reference) têm menor influência.

```
In [499... mse = mean_squared_error(Y_test, valores_preditos_rf)
mae = mean_absolute_error(Y_test, valores_preditos_rf)
r2s = r2_score(Y_test, valores_preditos_rf)
print("MSE: ", mse)
print("MAE: ", mae)
print("R2: ", r2s)
```

MSE: 6610948.74402515
MAE: 1236.6371045088888
R2: 0.9975435463984578

Os resultados da análise mostram que o modelo Random Forest tem um MSE de 6.593.390, um MAE de 1.236,54 e um R² de 0,998. Isso indica que o modelo tem um erro médio absoluto relativamente baixo e explica 99,75% da variabilidade dos dados, sugerindo uma boa precisão na previsão dos preços dos carros.

XGBoost

```
In [500... model_xgboost = XGBRegressor()
model_xgboost.fit(X_train, Y_train)
# Predição dos valores de salário com base nos dados de teste
valores_preditos_xgboost = model_xgboost.predict(X_test)
```

Analisando a importância das variáveis para estimar a variável target

```
In [501... model_xgboost.feature_importances_
feature_importances = pd.DataFrame(model_xgboost.feature_importances_, index = X_train.columns, columns=['importance']).sort_values(
feature_importances
```

Out[501...

	importance
fuel	0.301117
engine_size	0.294968
year_model	0.205772
gear	0.099849
brand	0.059079
model	0.017445
year_of_reference	0.015716
month_of_reference	0.006054

A análise de importância das variáveis revelou que o tipo de combustível (fuel) e o tamanho do motor (engine_size) são os fatores mais influentes na determinação do preço médio dos carros (avg_price_br1). Variáveis como o ano do modelo (year_model) e o tipo de câmbio (gear) também têm impacto significativo, enquanto a marca (brand) e o mês de referência (month_of_reference) têm menor influência.

In [503...

```
msexg = mean_squared_error(Y_test, valores_preditos_xgboost)
maexg = mean_absolute_error(Y_test, valores_preditos_xgboost)
r2sxo = r2_score(Y_test, valores_preditos_xgboost)
print("MSE XGBoost: ", msexg)
print("MAE XGBoost: ", maexg)
print("R2 XGBoost: ", r2sxo)
```

```
MSE XGBoost: 35962548.20807291
MAE XGBoost: 3665.9799093322717
R2 XGBoost: 0.9866372687965252
```