Desafio: Construindo um modelo de Regressão para Marketing

Contexto - Introdução

Uma empresa está investindo mensalmente em plataformas de publicidade online, como Youtube, Facebook e newspaper, para a prospecção de leads (pessoas interessadas em seus produtos). A fim de acompanhar o desempenho desses investimentos, a empresa registra todos os gastos com publicidade e todos os retornos de vendas gerados a partir desses investimentos.

Para **entender** melhor **a relação entre as variáveis** presentes nesses registros e **identificar os fatores que mais impactam** na geração de leads, a empresa solicitou a análise de um especialista em dados. **Além disso, a empresa busca criar um modelo de predição** de valores para estimar o retorno de vendas que pode ser gerado a partir de um determinado investimento em publicidade.

Sobre os dados

A tabela contém informações dos investimentos feitos pelo youtube, facebook, newspaper e também a quantidade de cada.

Coluna	Descrição
youtube	Investimento youtube
facebook	Investimento facebook
newspaper	Investimento newspaper
sales	Valor das vendas

Importando as Bibliotecas

```
#Manipulação de Dados
import pandas as pd
import numpy as np
import math

#Visualização
import plotly.express as px
import plotly.subplots as sp
import plotly.graph_objs as go
import plotly.figure_factory as ff
from plotly.subplots            import make_subplots

#Modelo
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

#Display
import warnings
from IPython.display            import Markdown, Image
```

Funções Auxiliares e Configurações

[ ] ↳ *1 célula oculta*

Importando o Dataset

```
df = pd.read_csv('MKT.csv')
df
```

	youtube	facebook	newspaper	sales
0	84.72	19.20	48.96	12.60
1	351.48	33.96	51.84	25.68
2	135.48	20.88	46.32	14.28
3	116.64	1.80	36.00	11.52
4	318.72	24.00	0.36	20.88
...	...	...	...	...
166	45.84	4.44	16.56	9.12

Etapas de Desenvolvimento

Para te ajudar nesse processo, detalhar o processo nas etapas a seguir:

170	278.52	10.32	10.44	16.08

Etapa 01) Análise Descritiva

Esta etapa consiste em explorar os dados do dataset para **compreender melhor as variáveis e identificar problemas**. Para isso, é recomendado utilizar a biblioteca **Pandas** para importar e manipular os dados e realizar cálculos estatísticos, além das bibliotecas de visualização.

É importante investigar o tipo de dado em cada variável, os valores e a distribuição dos dados. Ao final, espera-se ter uma interpretação sólida dos dados para avançar para a próxima etapa

Análise Descritiva do Dataset

```
1 df_informations(df)
```

Informações sobre o Dataset

Dataset tem 171 linhas e 4 colunas. Não possui linhas duplicadas. Sobre o dataset, temos:

	Not Null	Null	Perce Null	Dtype
youtube	171	0	0.00%	float64
facebook	171	0	0.00%	float64
newspaper	171	0	0.00%	float64
sales	171	0	0.00%	float64

Sobre Dtypes, temos:

	Dtype	Count	Perce
0	float64	4	100.00%

Estatística Descritiva

	count	mean	std	min	1%	25%	50%	75%	99%	max
youtube	171.00	178.02	102.45	0.84	6.01	91.08	179.76	262.98	351.73	355.68
facebook	171.00	27.67	17.91	0.00	0.44	11.70	26.76	43.68	59.28	59.52
newspaper	171.00	35.24	24.90	0.36	1.75	13.74	31.08	50.88	103.42	121.08
sales	171.00	16.92	6.31	1.92	5.60	12.54	15.48	20.82	30.85	32.40

As informações **.describe** podem ser úteis para entender a dispersão e a tendência central dos dados.

- Média: Representa a tendência central dos dados;
- Desvio Padrão: Indica a dispersão, sugerindo uma variabilidade dos valores;
- Quartis: Fornecem informações sobre a distribuição dos dados ao longo de diferentes partes; e
- Valor Mínimo e Máximo: Indicam a faixa em que os dados estão concentrados.

Logo, podemos observar que:

1. **Variável "youtube":**
    - A média de investimento do YouTube é de aproximadamente 178.02, o que nos mostra que é o que mais recebe invetimento.
    - O desvio padrão é relativamente alto (102.45), o que sugere que os dados do YouTube apresentam uma dispersão considerável em relação à média.

- Os quartis indicam que 25% dos dados estão abaixo de 91.08, 50% estão abaixo de 179.76 e 75% estão abaixo de 262.98.
- O valor mínimo é de 0.84 e o valor máximo é de 355.68, mostrando que a faixa de visualizações varia de forma considerável.

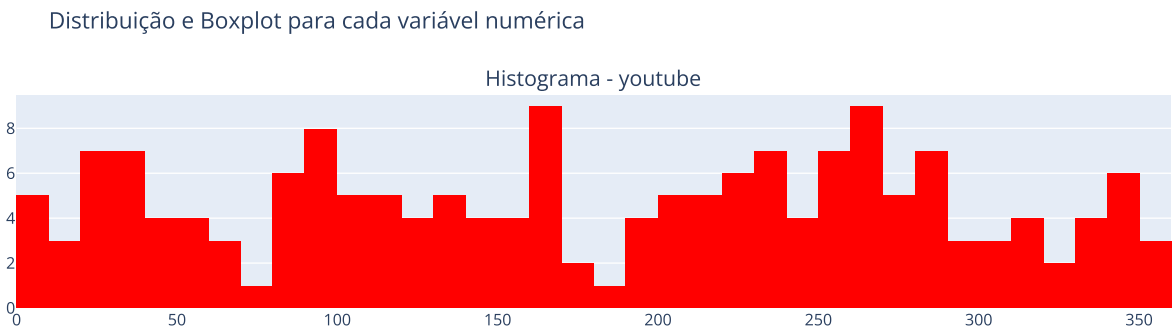
2. **Variável "facebook":**
   - A média (27.67) de investimento do Facebook é aproximadamente 6.5x menor em relação ao Youtube.
   - O desvio padrão é de 17.91, indicando uma dispersão alta (64.72%) em relação à média.
   - Os quartis mostram que 25% dos dados estão abaixo de 11.70, 50% estão abaixo de 26.76 e 75% estão abaixo de 43.68.
   - O valor mínimo é 0.00 e o valor máximo é 59.52, mostrando a faixa de valores para a métrica do Facebook.

3. **Variável "newspaper":**
   - A média (35.24) do newspaper é a segunda maior ficando atrás apenas do Youtube.
   - O desvio padrão é de 24.90, indicando uma dispersão relativamente alta (70.65%) em relação à média.
   - Os quartis mostram que 25% dos dados estão abaixo de 13.74, 50% estão abaixo de 31.08 e 75% estão abaixo de 50.88.
   - O valor mínimo é 0.36 e o valor máximo é 121.08, indicando uma ampla faixa de valores.

4. **Variável "sales":**
   - A média para os dados de vendas é de aproximadamente 16.92.
   - O desvio padrão é de 6.31, sugerindo uma dispersão moderada (37.29%) em relação à média.
   - Os quartis mostram que 25% dos dados estão abaixo de 12.54, 50% estão abaixo de 15.48 e 75% estão abaixo de 20.82.
   - O valor mínimo é 1.92 e o valor máximo é 32.40, mostrando a faixa de valores para as vendas.

Análise Gráfica

```
1 plot_distribution_and_boxplot(df)
```



Etapa 02) Análise Exploratória

Neta etapa iremos explorar mais a fundo os dados, **identificando relações entre as variáveis e descobrindo padrões relevantes**. Para isso, utilize técnicas de visualização de dados e análises estatísticas, buscando possíveis correlações e identificando possíveis outliers ou desvios da normalidade.

Correlação

Definição de Correlação

A correlação é uma medida estatística que *avalia a relação entre duas variáveis*. Ela indica a direção e a intensidade dessa relação, ou seja, se as variáveis se movem em conjunto (correlação positiva) ou de forma oposta (correlação negativa), e o quão forte essa relação é.

A força da correlação pode ser definida com base no valor do coeficiente de correlação, que varia de -1 a +1. Aqui estão as interpretações comuns para determinar a força da correlação:

1. **Correlação Fraca**:
    - Quando o coeficiente de correlação está próximo de 0, a correlação é considerada fraca.
    - Seja ela positiva ou negativa, a relação entre as variáveis é considerada fraca se o coeficiente de correlação estiver próximo de zero (por exemplo, entre -0,3 e 0,3).
    - Nesse caso, as variáveis têm uma associação limitada e seus movimentos não são consistentes.

2. **Correlação Média**:
    - Uma correlação é considerada média quando o coeficiente de correlação está em torno de -0,5 a -0,3 ou de 0,3 a 0,5.
    - A relação entre as variáveis é moderada e mostra algum grau de consistência em seus movimentos.
    - Uma correlação média indica que as variáveis têm alguma influência mútua, mas não é uma relação forte.

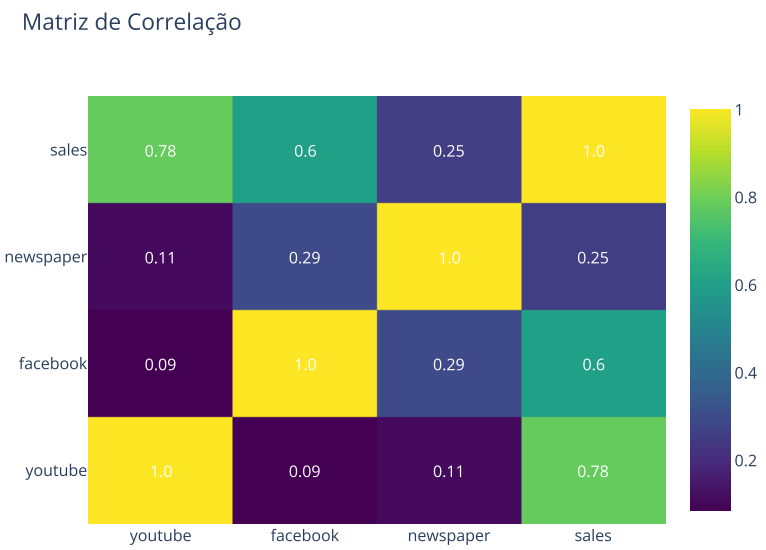
3. **Correlação Forte**:
    - A correlação é considerada forte quando o coeficiente de correlação está próximo de -1 ou 1.
    - Uma correlação positiva forte (próxima de +1) indica que as variáveis estão fortemente relacionadas e tendem a se mover na mesma direção.
    - Uma correlação negativa forte (próxima de -1) indica que as variáveis estão fortemente relacionadas, mas se movem em direções opostas.
    - Nesses casos, as variáveis têm uma associação consistente e os movimentos de uma variável estão altamente relacionados aos movimentos da outra.

É importante destacar que a força da correlação pode variar de acordo com o contexto e o domínio dos dados. Além disso, a correlação não implica causalidade direta, ou seja, apenas porque duas variáveis estão correlacionadas, não significa que uma causa a outra.

Matriz de Correlação

```
# Matriz de Correlação
correlation_matrix = df.corr()

# Cria o gráfico de heatmap
fig = go.Figure(data=go.Heatmap(
                   z=correlation_matrix,
```

```
                  x=correlation_matrix.columns,
                  y=correlation_matrix.columns,
                  colorscale='Viridis'))

# Adiciona anotações
annotations = []
for i, row in enumerate(correlation_matrix.values):
    for j, val in enumerate(row):
        annotations.append(go.layout.Annotation(text=str(round(val, 2)), x=correlation_matrix.columns[j], y=correlation_matrix.column
                                                showarrow=False, font=dict(color='white')))

# Atualiza layout
fig.update_layout(title='Matriz de Correlação',
                  annotations=annotations,
                  height=500, width=600)

fig.show()

```



Explicado o que é correlação, podemos observar que:

1. A correlação entre "youtube" e "sales" é de 0.78, o que indica uma correlação positiva forte. Isso sugere que há uma forte relação entre o investimento no YouTube e as vendas. Aumento nas vendas tendem a estar associados ao aumento de investimento no YouTube.
2. A correlação entre "facebook" e "sales" é de 0.60, o que indica uma correlação positiva forte. Isso sugere que há uma relação forte entre o investimento no Facebook e as vendas. Aumentos nas vendas tendem a estar associados ao aumento de investimento no Facebook.
3. A correlação entre "newspaper" e "sales" é de 0.25, o que indica uma correlação positiva fraca. Isso sugere que há uma relação fraca entre o investimento em newspaper e as vendas. A influência dos newspaper nas vendas é limitada, pois a correlação é relativamente baixa.

Scatter plot: Sales vs Investimento

```
# Reestrutura os dados para um formato longo
df_melted = df.melt(id_vars='sales', var_name='platform', value_name='investment')

# Cria um dicionário com os símbolos para cada plataforma
symbols = {'youtube': 'circle', 'facebook': 'diamond', 'newspaper': 'square'}

# Cria o gráfico de dispersão
fig = px.scatter(df_melted, x='investment', y='sales', color='platform',
                 symbol=df_melted['platform'].map(symbols),
                 title='Scatter plot: Sales vs Investimento',
                 labels={'investment':'Investimento', 'sales':'Sales'},
                 hover_data=['platform', 'investment', 'sales'])

fig.show()

```

Analisando o Scatter plot é possível observar que é necessário desprender ou investir muito mais dinheiro no Youtube para ter o mesmo retorno que as demais plataformas. Em contra partida o Facebook e Newspaper possuem uma distribuição semelhante, onde a relação Sales vs Investimento é mais vantajosa, sendo necessário despender menos dinheiro para um retorno maior.

Analisando a Eficiência da relação Sales vs Investimento

Cálculo do ROI individual

Uma forma de avaliar a eficiência dos investimentos em marketing é através do cálculo do Retorno sobre Investimento (ROI, em inglês Return on Investment). O ROI é uma métrica amplamente utilizada para avaliar a rentabilidade de um investimento. Ele é calculado da seguinte maneira:

ROI = (Retorno do Investimento - Custo do Investimento) / Custo do Investimento

Onde:

- Retorno do Investimento corresponde às Vendas (sales)
- Custo do Investimento é o valor investido nas plataformas (Youtube, Facebook e Newspaper)

Nossos dados indicam que as vendas (sales) estão correlacionadas com o valor investido em cada uma das plataformas. Ou seja, existe uma relação entre o valor investido e as vendas resultantes. Entretanto, é importante ressaltar que o valor total de vendas é influenciado pelo investimento em todas as plataformas, e não apenas em uma individualmente.

Embora calcular o ROI individual de cada plataforma possa não fornecer um quadro completamente preciso, pode ser uma maneira útil de começar a analisar a eficácia dos investimentos em marketing em cada plataforma.

Para simplificar, vamos fazer a suposição de que cada dólar investido em uma plataforma específica contribui igualmente para o total de vendas. Por exemplo, se investimos $84.72 no Youtube e 19.20 no Facebook, vamos considerar que cada um desses investimentos contribuiu igualmente para gerar as vendas de $12.60.

Com base nessas suposições, podemos prosseguir para analisar qual plataforma proporciona o melhor retorno sobre o investimento.

```
df['ROI_youtube'] = (df['sales'] - df['youtube']) / df['youtube']
df['ROI_facebook'] = (df['sales'] - df['facebook']) / df['facebook']
df['ROI_newspaper'] = (df['sales'] - df['newspaper']) / df['newspaper']
df
```

	youtube	facebook	newspaper	sales	ROI_youtube	ROI_facebook	ROI_newspaper
0	84.72	19.20	48.96	12.60	-0.85	-0.34	-0.74
1	351.48	33.96	51.84	25.68	-0.93	-0.24	-0.50
2	135.48	20.88	46.32	14.28	-0.89	-0.32	-0.69
3	116.64	1.80	36.00	11.52	-0.90	5.40	-0.68
4	318.72	24.00	0.36	20.88	-0.93	-0.13	57.00

```
1 df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
youtube	171.00	178.02	102.45	0.84	91.08	179.76	262.98	355.68
facebook	171.00	27.67	17.91	0.00	11.70	26.76	43.68	59.52
newspaper	171.00	35.24	24.90	0.36	13.74	31.08	50.88	121.08
sales	171.00	16.92	6.31	1.92	12.54	15.48	20.82	32.40
ROI_youtube	171.00	-0.84	0.22	-0.96	-0.92	-0.90	-0.85	1.29
ROI_facebook	171.00	inf	NaN	-0.96	-0.52	-0.32	0.32	inf
ROI_newspaper	171.00	0.41	4.70	-0.90	-0.68	-0.51	0.10	57.00

Corrigindo o inf do ROI_facebook

Após análise dos dados foi observado que na linha 98 do dataset a variável facebook é igual a 0, sendo assim quando calculamos o ROI, temos:

ROI = (10.56-0)/0

Matematicamente não é possível dividir um número por zero, então para corrigir esse problema vamos adotar uma premissa.

```
1 df.loc[98]
```

```
youtube         96.24
facebook         0.00
newspaper       11.04
sales           10.56
ROI_youtube     -0.89
ROI_facebook      inf
ROI_newspaper   -0.04
Name: 98, dtype: float64
```

Como premissa vamos adotar que o 0 será a média.

```
1 df['ROI_facebook'][98] = df['facebook'].mean()
```

```
1 df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
youtube	171.00	178.02	102.45	0.84	91.08	179.76	262.98	355.68
facebook	171.00	27.67	17.91	0.00	11.70	26.76	43.68	59.52
newspaper	171.00	35.24	24.90	0.36	13.74	31.08	50.88	121.08
sales	171.00	16.92	6.31	1.92	12.54	15.48	20.82	32.40
ROI_youtube	171.00	-0.84	0.22	-0.96	-0.92	-0.90	-0.85	1.29
ROI_facebook	171.00	0.62	3.48	-0.96	-0.52	-0.32	0.32	28.00
ROI_newspaper	171.00	0.41	4.70	-0.90	-0.68	-0.51	0.10	57.00

O ROI negativo, isso geralmente significa que o custo do investimento excede o retorno que você obteve. No contexto de uma campanha de marketing, isso poderia significar que o dinheiro que você gastou na campanha é maior do que a receita gerada por essa campanha.

Por exemplo, com um ROI médio de -0.90, se o valor investido foi de R$100, $aperdamédia, deacordocomesseROI, seriadeR$90.

- **ROI_youtube:** ROI médio de **-0.84** sugere que, em média, para cada unidade de moeda investida, a perda foi de 84% do valor investido. Essa é uma situação financeira **muito ruim**, indicando que o investimento está perdendo dinheiro em vez de gerar lucro.

- **ROI_facebook:** ROI médio de **0.62** sugere que, em média, para cada unidade de moeda investida, o ganho foi de 62% do valor investido. Essa é uma situação financeira **muito boa**, indicando que o investimento está gerando lucro.
- **ROI_newspaper:** ROI médio de **0.41** sugere que, em média, para cada unidade de moeda investida, o ganho foi de 41% do valor investido. Essa é uma situação financeira **muito boa**, indicando que o investimento está gerando lucro.

Cálculo do ROI consolidado

Primeiramente vamos precisar somar o valor investido em todas as plataformas.

```
df['plataformas'] = df['youtube'] + df['facebook'] + df['newspaper']
df
```

	youtube	facebook	newspaper	sales	ROI_youtube	ROI_facebook	ROI_newspaper
0	84.72	19.20	48.96	12.60	-0.85	-0.34	-0.74
1	351.48	33.96	51.84	25.68	-0.93	-0.24	-0.50
2	135.48	20.88	46.32	14.28	-0.89	-0.32	-0.69
3	116.64	1.80	36.00	11.52	-0.90	5.40	-0.68
4	318.72	24.00	0.36	20.88	-0.93	-0.13	57.00
...	...	...	...	...	...	...	...
166	45.84	4.44	16.56	9.12	-0.80	1.05	-0.45
167	113.04	5.88	9.72	11.64	-0.90	0.98	0.20
168	212.40	11.16	7.68	15.36	-0.93	0.38	1.00
169	340.32	50.40	79.44	30.60	-0.91	-0.39	-0.61
170	278.52	10.32	10.44	16.08	-0.94	0.56	0.54

171 rows × 8 columns

Depois, calcular o ROI consolidado.

```
df['ROI'] = (df['sales'] - df['plataformas']) / df['plataformas']
df
```

	youtube	facebook	newspaper	sales	ROI_youtube	ROI_facebook	ROI_newspaper
0	84.72	19.20	48.96	12.60	-0.85	-0.34	-0.74
1	351.48	33.96	51.84	25.68	-0.93	-0.24	-0.50
2	135.48	20.88	46.32	14.28	-0.89	-0.32	-0.69
3	116.64	1.80	36.00	11.52	-0.90	5.40	-0.68
4	318.72	24.00	0.36	20.88	-0.93	-0.13	57.00
...	...	...	...	...	...	...	...
166	45.84	4.44	16.56	9.12	-0.80	1.05	-0.45
167	113.04	5.88	9.72	11.64	-0.90	0.98	0.20
168	212.40	11.16	7.68	15.36	-0.93	0.38	1.00
169	340.32	50.40	79.44	30.60	-0.91	-0.39	-0.61
170	278.52	10.32	10.44	16.08	-0.94	0.56	0.54

171 rows × 9 columns

```
df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
youtube	171.00	178.02	102.45	0.84	91.08	179.76	262.98	355.68
facebook	171.00	27.67	17.91	0.00	11.70	26.76	43.68	59.52

ROI médio Consolidado de **-0.92** sugere que, em média, para cada unidade de moeda investida, a perda foi de 92% do valor investido. Essa é uma situação financeira **muito ruim**, indicando que o investimento está perdendo dinheiro em vez de gerar lucro.

Outliers

Definição de Outliers

Outliers são valores em um conjunto de dados que são significativamente diferentes dos outros. Eles são valores extremos que se desviam da média e das medidas padrão do conjunto de dados.

Para identificar outliers em um conjunto de dados, você pode usar vários métodos. Aqui estão algumas opções populares:

1. **Gráficos Boxplot**
2. **Z-Score**
3. **Desvio absoluto mediano (MAD)**
4. **Regra do IQR (Interquartile Range)**

A análise de outliers é crucial para uma análise de dados precisa, pois outliers podem distorcer os resultados. Boxplots são uma ferramenta gráfica comum para identificar outliers, baseando-se no Intervalo Interquartil (IQR).

Apesar da facilidade de identificar outliers visualmente em boxplots, elaborei uma função para calcular numericamente o IQR e os limites de outliers. Essa função nos permite entender melhor o processo de detecção de outliers.

Apesar de útil, não é necessário calcular manualmente esses valores sempre. Ferramentas como boxplots fazem isso automaticamente, facilitando a identificação de outliers. Na sequência, utilizaremos boxplots para visualizar a presença de outliers em nossos dados.

O IQR é a diferença entre o terceiro quartil (Q3) e o primeiro quartil (Q1) dos dados. Qualquer ponto de dados que esteja abaixo de **Q1-1.5IQR** ou acima de **Q3+1.5IQR** pode ser considerado um outlier.

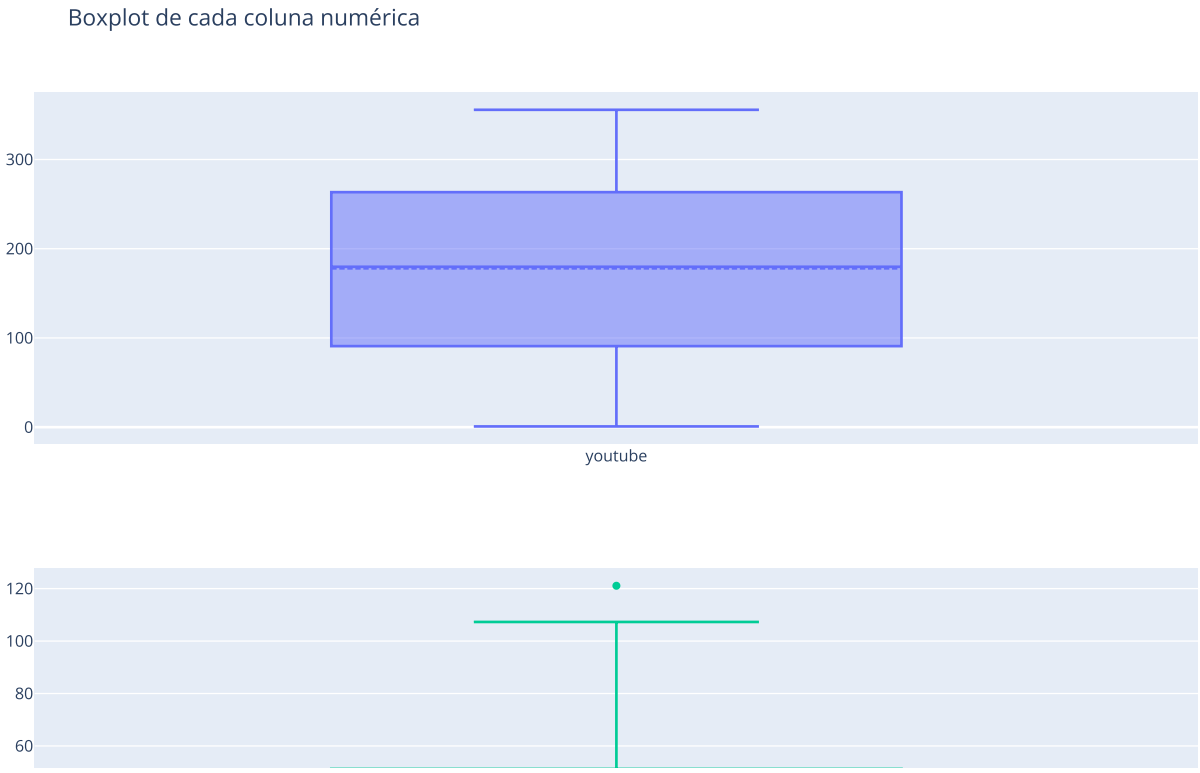
Cálculo dos Outliers

```
# Usando a função no DataFrame df
iqr_df = calculate_iqr(df[['youtube', 'facebook', 'newspaper', 'sales']])
iqr_df

```

	Coluna	Q1	Q3	IQR	Lower Bound	Upper Bound	Min	Max
0	youtube	91.08	262.98	171.90	-166.77	520.83	0.84	355.68
1	facebook	11.70	43.68	31.98	-36.27	91.65	0.00	59.52
2	newspaper	13.74	50.88	37.14	-41.97	106.59	0.36	121.08
3	sales	12.54	20.82	8.28	0.12	33.24	1.92	32.40

Os outliers são valores que desviam notavelmente do restante dos dados. No contexto do IQR, outliers são aqueles valores que estão **abaixo** do "Lower Bound" ou **acima** do "Upper Bound". Assim, na tabela IQR que geramos, qualquer valor da coluna "Min" menor que o respectivo "Lower Bound" e qualquer valor da coluna "Max" maior que o respectivo "Upper Bound" são considerados outliers.

```
# Usando a função com nosso DataFrame df
plot_boxplots(df[['youtube', 'facebook', 'newspaper', 'sales']])

```



Ao analisar os boxplots das nossas variáveis 'youtube', 'facebook', 'newspaper' e 'sales', notamos que apenas a variável 'newspaper' apresenta outliers. No entanto, essa quantidade de outliers parece ser pequena e, portanto, é improvável que cause um impacto significativo em nossa análise posterior ou distorça nossos resultados.

Conclusão Etapa 2 - Análise Exploratória

1. **Correlação:** A análise de correlação indicou que o investimento em YouTube e Facebook tem uma forte relação positiva com as vendas, com o YouTube mostrando a relação mais forte. Isso sugere que aumentar o investimento nessas plataformas poderia resultar em um aumento nas vendas. Por outro lado, o investimento em Newspaper mostrou uma relação mais fraca com as vendas, indicando que essa plataforma pode não ser tão eficaz para impulsionar as vendas.
2. **Sales vc Investimento:** Ao analisar a relação de investimento vs vendas através dos gráficos de dispersão, notamos que o investimento no YouTube requer mais recursos para alcançar o mesmo nível de vendas comparado ao Facebook e Newspaper. Isso indica que o Facebook e Newspaper podem oferecer um retorno maior por unidade de investimento.
3. **Eficiência - ROI Individual:** O cálculo do ROI por plataforma revelou uma situação preocupante para o YouTube, com um ROI médio de -0.84. Isso sugere que o investimento no YouTube está atualmente gerando uma perda em vez de lucro. Em contrapartida, o Facebook e o Newspaper mostraram um ROI positivo, indicando que esses investimentos estão gerando lucro.
4. **Eficiência - ROI Consolidado:** Entretanto, ao considerar o ROI consolidado de todos os investimentos, vemos que a situação geral é de perda, com um ROI de -0.92. Isso indica que o investimento total em marketing está gerando uma perda, o que pode requerer uma reavaliação da estratégia de marketing.
5. **Outliers:** Por último, a análise de outliers revelou a existência de alguns outliers apenas na variável Newspaper. Embora a presença de outliers possa impactar algumas análises estatísticas, a quantidade observada é pequena e, portamente, é pouco provável que altere significativamente nossas conclusões.

Etapa 03) Modelagem

Para esta etapa, deve-se **construir um modelo** simples de **regressão** que permita a previsão solicitada pela empresa, com base nos dados disponíveis. Para isto, importe as bibliotecas necessárias e carregue os conjuntos de dados para iniciar a sua construção!

Definição de Regressão Linear

Regressão Linear Simples

A regressão linear é uma técnica estatística que tenta modelar a relação entre uma variável dependente (também conhecida como variável de resposta) e uma ou mais variáveis independentes (também conhecidas como variáveis preditoras) por meio de uma equação linear. Na regressão linear simples (com uma variável independente), essa relação é modelada como uma linha reta (daí o termo "linear"). Em regressão linear múltipla (com mais de uma variável independente), essa relação é modelada como um hiperplano.

Vamos pensar na regressão linear simples por um momento. A equação básica que estamos tentando resolver é:

$$y = mx + b$$

Onde:

- ***y*** é a variável dependente;
- ***x*** é a variável independente;
- ***m*** é a inclinação da linha de regressão (representa o efeito de ***x*** sobre ***y***);
- ***b*** é a interceptação (representa o valor de ***y*** quando ***x*** é 0).

A regressão linear, em sua essência, é um método para encontrar os melhores valores para m e b. E quando dizemos "melhor", estamos falando em termos de minimizar a distância entre a linha de regressão (os valores previstos de y para qualquer valor de x) e os pontos de dados reais.

Abaixo eu criei um GIF que demonstra o comportamento da reta quando alteramos os valores de ***m*** e ***b*** de um Regressão Linear Simples (com **UMA** variável independente).

```
#Image(url='GIF Linear Regression.gif')
```

Regressão Linear Múltipla

Passando para a regressão linear múltipla, a ideia básica é a mesma, mas a matemática se torna um pouco mais complicada. Agora, em vez de termos apenas um m e um b, temos um coeficiente para cada variável independente e uma interceptação. Então, nossa equação se parece com:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \varepsilon$$

Onde:

- ***y*** é a variável dependente, ou seja, a variável que estamos tentando prever ou estimar;
- β_0 é o termo de interceptação. Ele representa o valor esperado de Y quando todas as variáveis independentes (Xs) são iguais a zero;
- $\beta_1, \beta_2, ..., \beta_n$ são os coeficientes de regressão. Eles representam a mudança esperada na variável dependente (Y) para cada mudança de uma unidade na respectiva variável independente, mantendo todas as outras variáveis independentes constantes;
- $x_1, x_2, ..., x_n$ são as variáveis independentes, ou seja, as variáveis que usamos para prever ou estimar Y.
- ε é o termo de erro, também conhecido como resíduos. Ele representa a diferença entre o valor real e o valor previsto de Y.

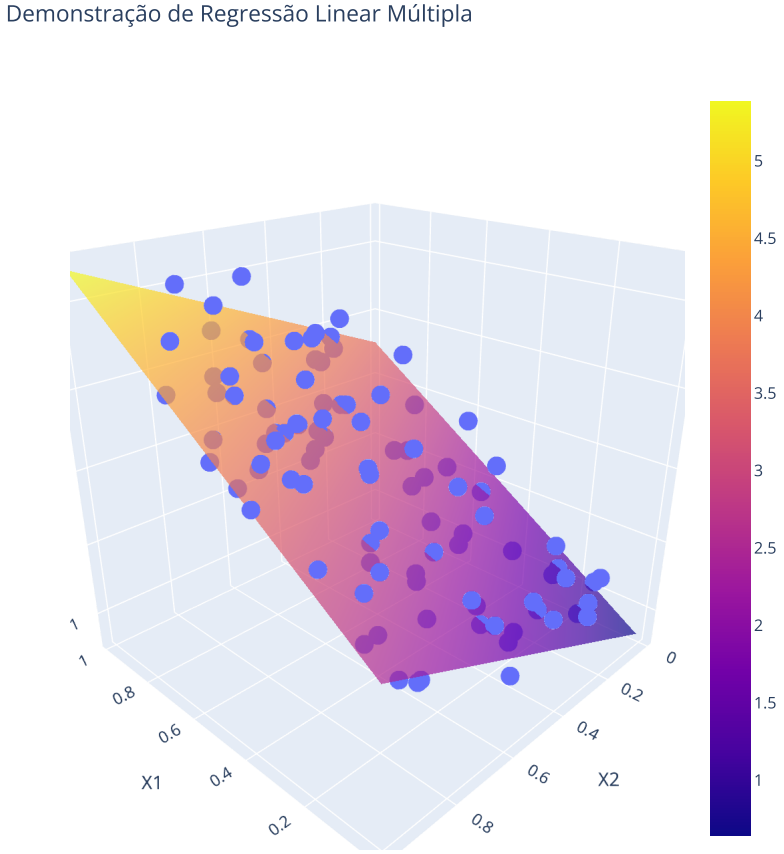
Quando estamos trabalhando com regressão linear múltipla, estamos tentando ajustar um modelo a um conjunto de dados que tem múltiplas variáveis independentes, que resultam em uma dimensão adicional para cada variável adicional.

Uma maneira de visualizar isso é imaginar que estamos tentando ajustar um plano em um espaço tridimensional em vez de uma linha em um espaço bidimensional. Se você tiver duas variáveis independentes, poderá visualizá-las ao longo de dois eixos (digamos x e y), e a variável dependente ao longo de um terceiro eixo (digamos z). Nesse caso, o "ajuste" mais próximo dos pontos de dados seria um plano que minimiza a distância entre os pontos de dados e o próprio plano.

Agora, se houvesse mais de duas variáveis independentes, o mesmo conceito se aplica, mas seria mais difícil visualizar porque estaríamos trabalhando em mais de três dimensões. No entanto, a matemática subjacente é a mesma: estamos tentando encontrar o "plano" (ou, mais tecnicamente, o hiperplano) que minimiza a distância entre os pontos de dados e o próprio hiperplano.

Abaixo eu criei um gráfico 3D para demonstrar o comportamento de uma Regressão Linear Múltipla (com **DUAS** variável independente).

```
# Gerando dados aleatórios para o exemplo
np.random.seed(0)
x1 = np.random.rand(100, 1)
x2 = np.random.rand(100, 1)
y2 = 3*x1 + 2*x2 + np.random.rand(100, 1)

# Modelo de regressão linear múltipla
model = LinearRegression()
model.fit(np.hstack([x1, x2]), y2)

# Você precisa chamar a função com os coeficientes obtidos do modelo.
update_graph2(model.coef_[0][0], model.coef_[0][1], model.intercept_[0])
```



Separando em conjuntos de Treino e Teste

```
# Crie um novo dataframe df_model
df_model = df[['youtube', 'facebook', 'newspaper', 'sales']].copy()

# Antes da divisão, criei uma nova coluna chamada 'split'
df_model['split'] = 'train'

# Dividindo o conjunto de dados
X = df_model[['youtube', 'facebook', 'newspaper']]
y = df_model['sales']

# Dividindo os dados em conjuntos de treino e teste
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=42)

# Indique que os dados em X_test estão na partição de teste
df_model.loc[X_test.index, 'split'] = 'test'

# Verificando os tamanhos dos conjuntos de treino e teste
print("Treino:", X_train.shape, y_train.shape)
print("Teste:", X_test.shape, y_test.shape)
```

```
Treino: (136, 3) (136,)
Teste: (35, 3) (35,)
```

```
df_model
```

	youtube	facebook	newspaper	sales	split
0	84.72	19.20	48.96	12.60	train
1	351.48	33.96	51.84	25.68	train

Etapa 04) Calculando Predição

Para concluirmos a demanda solicitada pela empresa, iremos **aplicar o modelo de regressão construído** nas etapas anteriores **para realizar as previsões** de retorno de vendas que pode ser gerado a partir de um determinado investimento em publicidade e assim, poderemos apresentá-lo a empresa.

Através dessas previsões, poderemos avaliar o impacto dos diferentes níveis de investimento em marketing nas vendas, auxiliando na tomada de decisões e na definição de estratégias de negócio.

Treinando o Modelo

O método ***.fit()*** é usado para treinar ou ajustar o modelo aos seus dados de treinamento. Em outras palavras, é aqui que o modelo aprende os padrões nos dados.

No caso de um modelo de regressão linear, o processo de ajuste envolve aprender os coeficientes que minimizam a soma dos erros quadrados entre os valores reais e os valores previstos pelo modelo. Para isso, o modelo usa o método dos mínimos quadrados.

Então, quando chamamos ***model.fit(X_train, y_train)***, estamos dizendo ao modelo para encontrar os melhores parâmetros (coeficientes de regressão) que mapeiam as variáveis independentes ***X_train*** para a variável dependente ***y_train*** com o menor erro.

Uma vez que o modelo foi treinado usando ***.fit()***, ele pode ser usado para fazer previsões em novos dados usando o método ***.predict()***. Nesse ponto, o modelo aplica os coeficientes de regressão que aprendeu durante o treinamento para prever a variável dependente a partir das variáveis independentes nos novos dados.

```
# Treinando o modelo
model = LinearRegression()
model.fit(X_train, y_train)

# Crie as previsões
y_train_pred = model.predict(X_train)
y_test_pred = model.predict(X_test)
```

Ao aplicar o método ***predict*** aos dados de treinamento **(X_train)**, estamos gerando previsões para os dados que foram usados para treinar o modelo. Essas previsões podem ser comparadas com os valores reais **(y_train)** para avaliar o quão bem o nosso modelo aprendeu os padrões nos dados de treinamento. Esta é uma forma de avaliar a precisão do treinamento do nosso modelo.

Por outro lado, quando aplicamos o método ***predict*** aos dados de teste **(X_test)**, estamos gerando previsões para novos dados que o modelo ainda não viu. Comparando essas previsões com os valores reais **(y_test)**, podemos avaliar a capacidade do nosso modelo de generalizar para novos dados. Esta é uma medida da precisão de teste do nosso modelo.

```
# Adicionando as previsões ao dataframe
df_model.loc[X_train.index, 'prediction'] = y_train_pred
df_model.loc[X_test.index, 'prediction'] = y_test_pred

df_model
```

	youtube	facebook	newspaper	sales	split	prediction
0	84.72	19.20	48.96	12.60	train	11.18
1	351.48	33.96	51.84	25.68	train	25.84
2	135.48	20.88	46.32	14.28	train	13.75
3	116.64	1.80	36.00	11.52	train	9.21
4	318.72	24.00	0.36	20.88	train	22.45
...	...	...	...	...	...	...
166	45.84	4.44	16.56	9.12	train	6.59
167	113.04	5.88	9.72	11.64	test	9.84
168	212.40	11.16	7.68	15.36	train	15.26
169	340.32	50.40	79.44	30.60	train	28.54
170	278.52	10.32	10.44	16.08	train	18.02

171 rows × 6 columns

```
tabela = pd.DataFrame()
tabela["y_teste"] = y_test.values
tabela["Previsoes Regressao Linear"] = y_test_pred

# Calcula os resíduos
tabela["Residuos"] = tabela["y_teste"] - tabela["Previsoes Regressao Linear"]

# Cria o objeto Figure
fig = go.Figure()

# Adiciona uma linha para os valores reais
fig.add_trace(go.Scatter(y=tabela['y_teste'], mode='lines', name='Valores Reais'))

# Adiciona uma linha para os valores previstos
fig.add_trace(go.Scatter(y=tabela['Previsoes Regressao Linear'], mode='lines', name='Previsoes Regressao Linear'))

# Adiciona uma linha para os resíduos
fig.add_trace(go.Scatter(y=tabela['Residuos'], mode='lines', name='Resíduos'))

# Adiciona título e rótulos de eixo
fig.update_layout(
    title='Valores Reais vs Previstos vs Resíduos',
    xaxis=dict(title='Índice'),
    yaxis=dict(title='Vendas')
)

# Mostra o gráfico
fig.show()

```

Interpretação do Gráfico

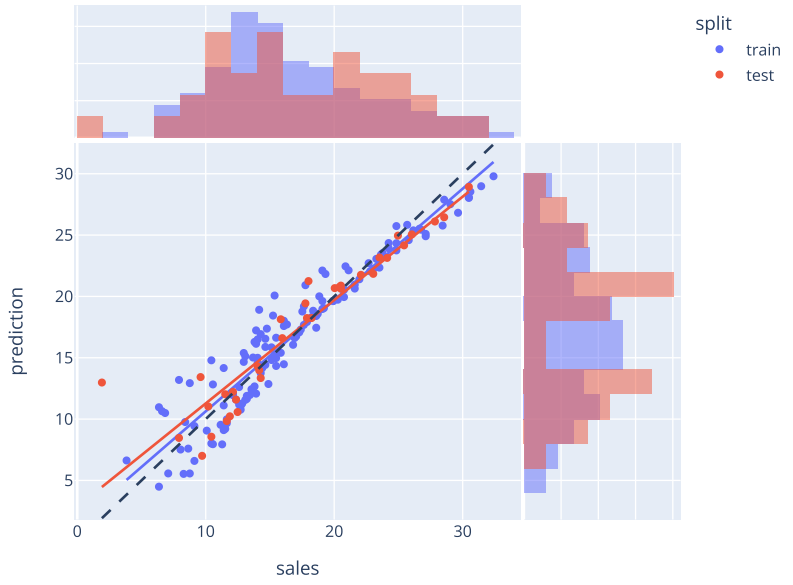
- A linha "Valores Reais" mostra os valores reais de venda que foram observados no conjunto de dados de teste.
- A linha "Previsões da Regressão Linear" mostra os valores de venda que foram previstos pelo nosso modelo de regressão linear a partir das variáveis independentes no conjunto de dados de teste.
- A linha "Resíduos" mostra a diferença entre os valores reais e os previstos, também conhecida como erro de previsão ou resíduo.

Comparando as linhas de "Valores Reais" e "Previsões da Regressão Linear", podemos ter uma ideia de quão bem o nosso modelo de regressão está funcionando. Idealmente, queremos que essas duas linhas estejam o mais próximas possível, o que indicaria que o nosso modelo está fazendo boas previsões.

A linha de "Resíduos" nos permite ver a magnitude dos erros de previsão. Valores próximos de zero indicam boas previsões, enquanto valores grandes (positivos ou negativos) indicam previsões ruins. Se vemos muitos valores grandes na linha de "Resíduos", isso pode ser um sinal de que o nosso modelo de regressão pode ser melhorado.

```
# Scatter plot
fig = px.scatter(
    df_model, x='sales', y='prediction',
```

```
    marginal_x='histogram', marginal_y='histogram',
    color='split', trendline='ols'
)
fig.update_traces(histnorm='probability', selector={'type':'histogram'})
fig.add_shape(
    type="line", line=dict(dash='dash'),
    x0=y.min(), y0=y.min(),
    x1=y.max(), y1=y.max()
)

fig.show()
```



Interpretação do Gráfico

- Os pontos representam as observações. Os pontos coloridos representam se a observação pertence ao conjunto de treinamento (azul) ou ao conjunto de teste (vermelho).
- O histograma na margem superior mostra a distribuição das vendas reais e o histograma na margem direita mostra a distribuição das vendas previstas.
- A linha tracejada preta representa a linha de perfeita igualdade (y = x), ou seja, os pontos que caem nessa linha são aqueles para os quais a venda real é igual à venda prevista. Idealmente, queremos que nossos pontos fiquem o mais próximos possível dessa linha.
- A linha de tendência azul (ols - ordinary least squares) é a linha de melhor ajuste para o conjunto de treinamento, minimizando a soma dos quadrados dos resíduos (ou erros).
- A linha de tendência vermelha é a linha de melhor ajuste para o conjunto de teste.

No geral, este gráfico nos permite ver quão bem o nosso modelo de regressão está prevendo as vendas. Se o modelo é bom, os pontos devem estar próximos da linha tracejada preta (linha de perfeita igualdade) e as linhas de tendência azul e vermelha devem ser semelhantes.

Avaliação do Modelo

O Root Mean Square Error (RMSE) e o coeficiente de determinação (R^2) são duas métricas comumente usadas para avaliar a performance de um modelo de regressão.

Root Mean Square Error (RMSE)

O RMSE é uma medida de erro que compara os valores previstos por um modelo com os valores reais. Ele é calculado ao se tomar a média dos quadrados dos erros, e em seguida tirando a raiz quadrada. Essa métrica dá uma ideia da quantidade de erro que o sistema normalmente comete em suas previsões, com um maior peso para erros maiores. Uma vantagem do RMSE é que o erro é expresso na mesma unidade que a variável de saída (y).

Passo a passo de como o RMSE é calculado:

1. **Erro de Previsão:** Para cada ponto de dados, você calcula a diferença entre a previsão do modelo e o valor real. Se o seu modelo previu um valor de 10 para uma determinada observação, mas o valor real é 12, o erro de previsão é -2 (10-12). Se a previsão fosse 14, o erro

seria 2 (14-12).

2. **Quadrado do Erro de Previsão:** Em seguida, você eleva ao quadrado cada erro de previsão. Fazemos isso por dois motivos:
    - Primeiro: para garantir que todos os erros sejam positivos (a diferença -2 e a diferença 2 têm o mesmo impacto).
    - Segundo: para dar mais peso a erros maiores. No exemplo anterior, o quadrado de -2 é 4 e o quadrado de 2 é 4.

3. **Média dos Quadrados dos Erros:** Em seguida, você calcula a média de todos os quadrados dos erros de previsão. Isso é conhecido como ***Mean Squared Error (MSE)***. Se você teve 5 previsões, e os quadrados dos erros foram [4, 4, 9, 4, 1], o MSE seria a soma desses números dividida por 5, que é 22/5 = 4.4.
4. **Raiz Quadrada da Média dos Quadrados dos Erros:** Finalmente, você tira a raiz quadrada do MSE para obter o RMSE e para trazer o erro de volta às unidades originais do output. Ao elevar ao quadrado os erros, como fazemos ao calcular o MSE, estamos efetivamente colocando os erros em termos de suas unidades ao quadrado. Isso pode ser um pouco abstrato e difícil de interpretar. No exemplo anterior, a raiz quadrada de 4.4 é aproximadamente 2.097.

Portanto, um RMSE de 2.097 significaria que, em média, as previsões do modelo estão cerca de 2.097 unidades distantes dos valores reais. Quanto menor o RMSE, melhor o modelo é capaz de prever os dados.

R^2

O coeficiente de determinação, ou R^2, é uma medida estatística que indica a proporção da variação na variável dependente que é previsível a partir da(s) variável(is) independente(s). Em outras palavras, R^2 é uma medida de quão bem as previsões do modelo se ajustam aos dados reais. O valor de R^2 varia entre 0 e 1, onde 1 indica que o modelo explica toda a variabilidade dos dados em torno da média.

Passo a passo de como o R^2 é calculado:

R^2 = 1 - (Soma dos Quadrados dos Resíduos / Soma Total dos Quadrados)

Vamos supor que temos os seguintes valores reais de Y: [3, -0.5, 2, 7] e suas previsões correspondentes do modelo são: [2.5, 0.0, 2, 8].

1. **Calcule a média de Y:** neste caso, a média é (3 - 0.5 + 2 + 7)/4 = 2.875.
2. **Calcule a soma total dos quadrados (SST):** que é a soma das diferenças quadradas entre cada valor de Y e a média de Y. Para nosso exemplo, a SST é $(3-2.875)^2 + (-0.5-2.875)^2 + (2-2.875)^2 + (7-2.875)^2 = 30.375$.
3. **Calcule a soma dos quadrados dos resíduos (SSR):** que é a soma das diferenças quadradas entre cada valor de Y e a previsão correspondente do modelo. Para nosso exemplo, a SSR é $(3-2.5)^2 + (-0.5-0.0)^2 + (2-2)^2 + (7-8)^2 = 1.5$.
4. **R^2:** é calculado como 1 - (SSR/SST). Então, nosso R^2 é 1 - (1.5/30.375) = 0.951.

Um R^2 de 0.951 indica que 95.1% da variação total em Y é explicada pelo modelo, o que é um bom ajuste.

Conclusão Etapa 4 - Avaliação do Modelo

```
# Calcular o RMSE
rmse_train = np.sqrt(mean_squared_error(y_train, y_train_pred))
rmse_test = np.sqrt(mean_squared_error(y_test, y_test_pred))

print(f"RMSE no conjunto de treinamento: {rmse_train:.2f}")
print(f"RMSE no conjunto de teste: {rmse_test:.2f}")
```

```
RMSE no conjunto de treinamento: 1.89
RMSE no conjunto de teste: 2.36
```

```
# Calcular o R²
r2_train = r2_score(y_train, y_train_pred)
r2_test = r2_score(y_test, y_test_pred)

print(f"R² no conjunto de treinamento: {r2_train:.2f}")
print(f"R² no conjunto de teste: {r2_test:.2f}")
```

```
R² no conjunto de treinamento: 0.91
R² no conjunto de teste: 0.87
```

```
# Obtendo os coeficientes
coef = model.coef_

# Transformando em porcentagem
coef_percent = coef * 100

# Formatando os coeficientes para remover a notação científica
formatted_coef = ['{:.2f}%'.format(i) for i in coef_percent]

```

```
# Associando cada coeficiente à sua feature correspondente
features = ['youtube', 'facebook', 'newspaper']
results = dict(zip(features, formatted_coef))

# Imprimindo os resultados
for feature, coef in results.items():
    print(f'{feature} - {coef}')

```

```
youtube - 4.42%
facebook - 19.45%
newspaper - -0.00%
```

Com base nos resultados, podemos concluir que:

1. O modelo apresentou bom desempenho, com RMSE de 1.89 no treino e 2.36 no teste, indicando pequena diferença entre os valores previstos e reais.
2. O alto R^2 (0.91 treino e 0.87 teste) sugere que o modelo explica uma grande parte da variação nos dados.
3. Em relação aos coeficientes:
    - Para cada aumento de uma unidade em youtube, espera-se que as sales aumentem, em média, 4.42% (assumindo que todas as outras variáveis permaneçam constantes).
    - Para cada aumento de uma unidade em facebook, espera-se que as sales aumentem, em média, 19.45% (assumindo que todas as outras variáveis permaneçam constantes).
    - Para cada aumento de uma unidade em newspaper, espera-se que as sales diminuam, em média, 0.49% (assumindo que todas as outras variáveis permaneçam constantes).

Em suma, o modelo de regressão linear é eficaz em prever as vendas baseado nos investimentos em publicidade e os resultados da análise sugerem que a plataforma do **Facebook** pode ser a mais eficaz para gerar vendas a partir do investimento em publicidade, dada a sua maior correlação com as vendas (representada pelo coeficiente de 19.45%). Isso sugere que cada unidade de aumento no investimento em publicidade no Facebook poderia resultar, em média, em um aumento de 19.45% nas vendas, supondo que todas as outras variáveis permaneçam constantes.