Relatório Técnico de Estágio em IA
VExpenses

Yvens Almeida Girão

CEARÁ - JANEIRO DE 2025

1. Introdução

Neste projeto, eu tive o desafio de prever se usuários de um site imobiliário comprariam ou não uma casa, com base em um conjunto de dados fornecido. Para isso, apliquei técnicas de análise de dados e machine learning, tratando o problema como uma tarefa de classificação binária.

Ao longo do trabalho, explorei o comportamento das variáveis, preparei os dados para o modelo e utilizei a Regressão Logística como abordagem inicial. Escolhi ferramentas como Pandas, NumPy e Scikit-learn para o processamento e construção do modelo, além de Matplotlib e Seaborn para criar visualizações que ajudam a interpretar os dados e os resultados.

Aqui, compartilho as etapas que segui, os principais resultados e também as reflexões sobre o que poderia ser melhorado. Este foi um aprendizado significativo e uma ótima oportunidade para aplicar conhecimentos em Machine Learning de forma prática.

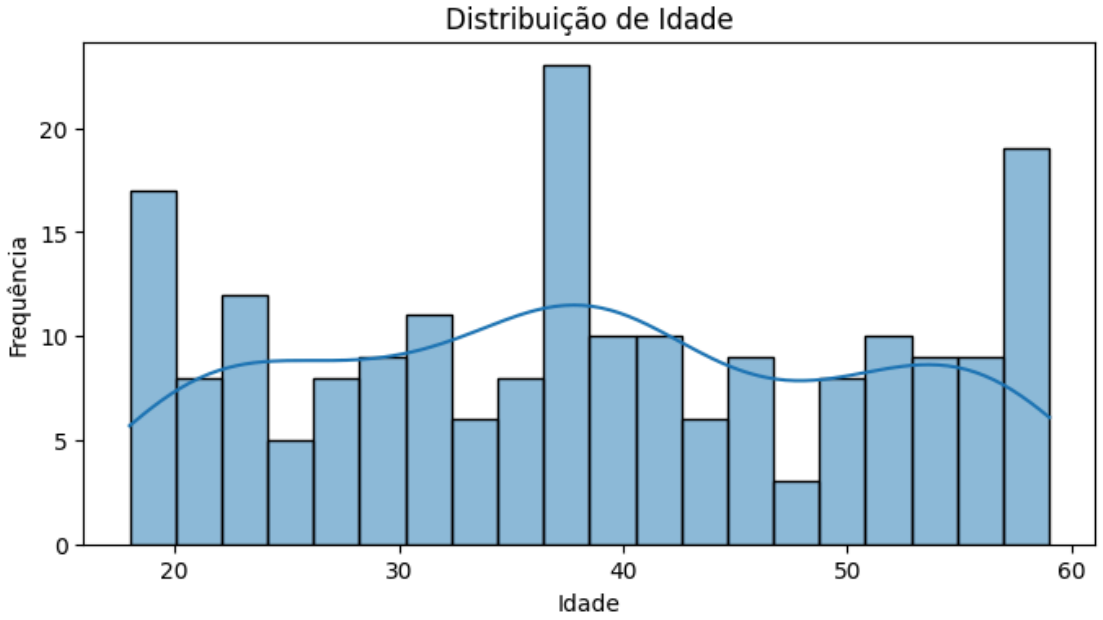
2. Análise exploratória dos Dados

Antes de construir o modelo, dediquei um tempo para entender melhor o conjunto de dados fornecido. Essa etapa foi essencial para identificar padrões, possíveis relações entre as variáveis e problemas que poderiam impactar a qualidade do modelo.

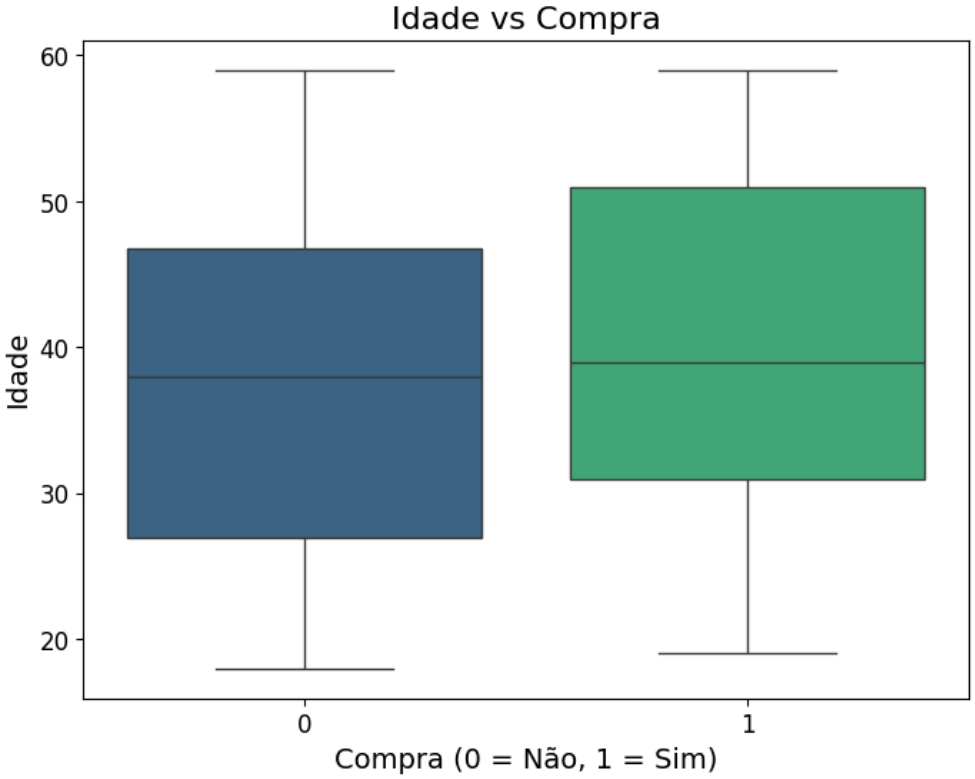
2.1 Análise das Variáveis Numéricas

Primeiramente, examinei as distribuições das variáveis numéricas:

- **Idade:** A maioria dos usuários está concentrada em uma faixa específica, o que pode influenciar o perfil de compradores potenciais.

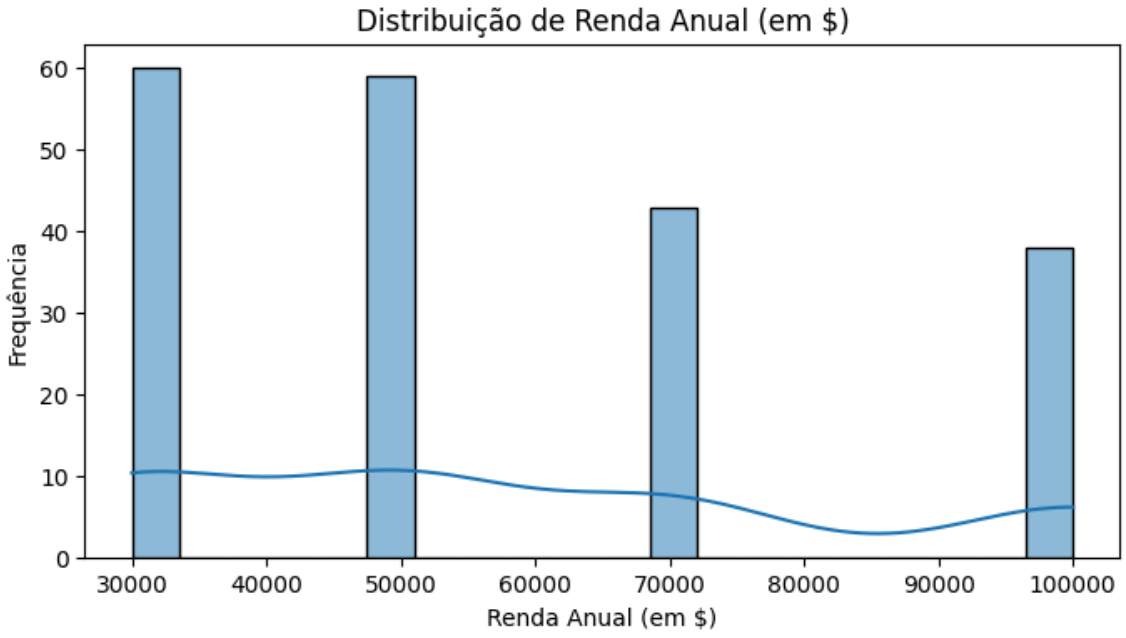


Nota-se que no **gráfico** da distribuição de Idade, possui valores variados. A linha no gráfico representa a **curva de densidade kernel (Kernel Density Estimation - KDE)**, A linha ajuda a visualizar a tendência geral dos dados. Por exemplo, é possível perceber os picos e vales que mostram onde há maior ou menor concentração de idades.

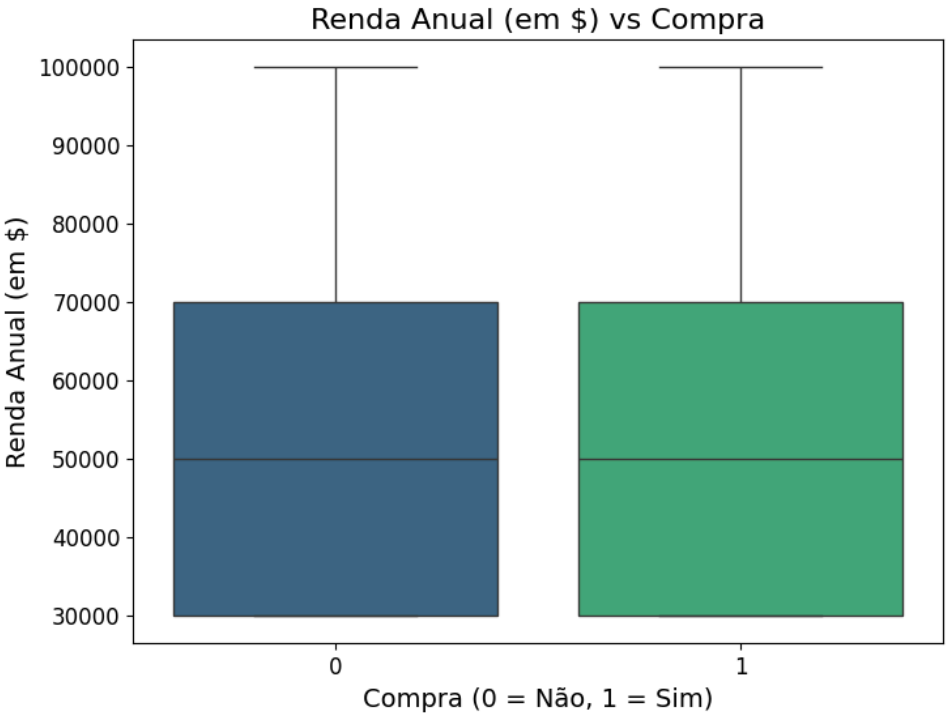


O gráfico acima é um **boxplot**, entre a variável Idade e a variável Compra, não existe muita diferença significativa na distribuição da faixa etária entre os dois grupos (quem comprou ou quem não comprou).

- **Renda Anual:** Foi possível observar uma dispersão considerável, sugerindo a necessidade de normalização.

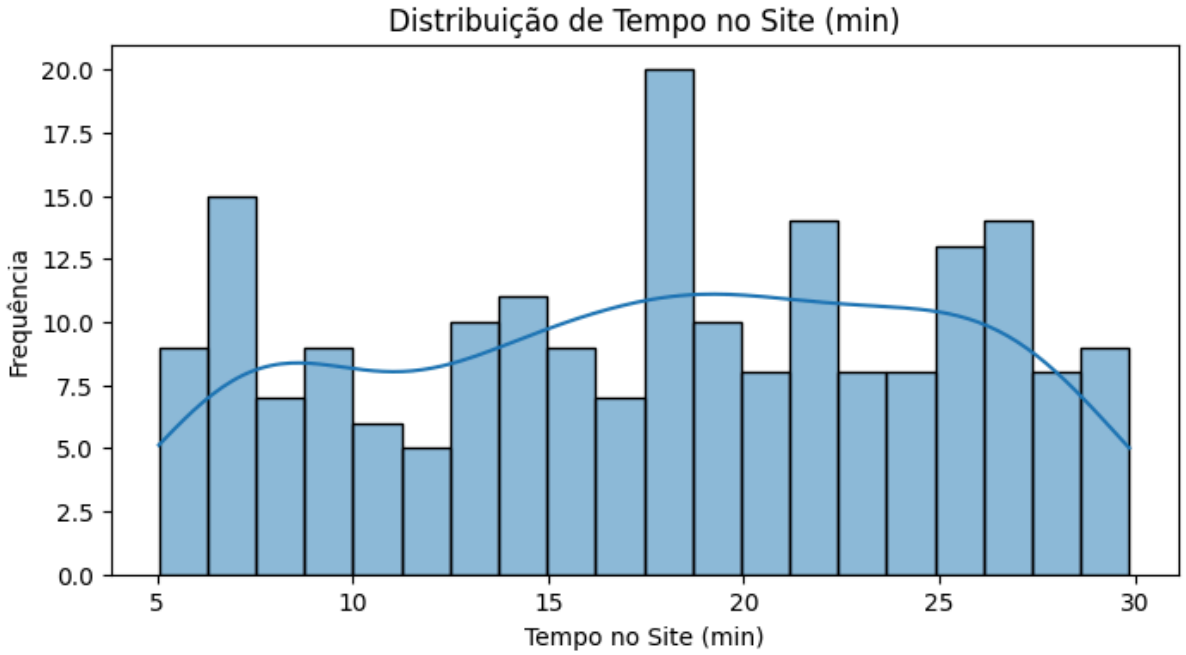


Nota-se que o **histograma** da Distribuição de Renda, nota-se que há uma ausência de dados ou baixa frequência em faixas intermediárias, como entre 60.000 e 90.000 dólares.

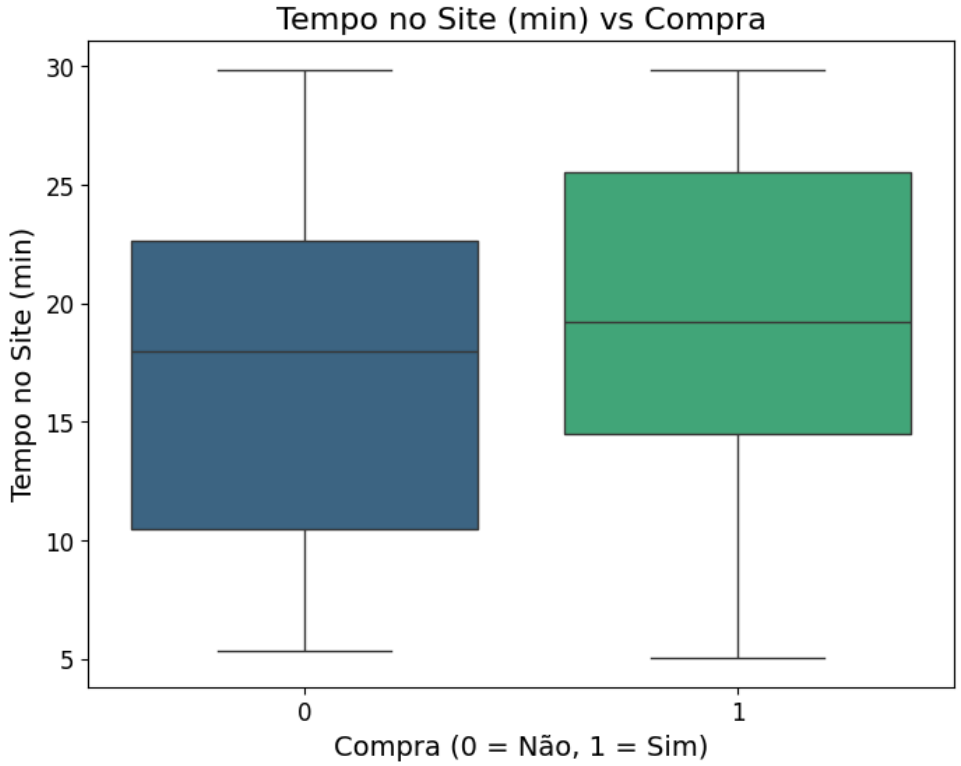


Com base neste gráfico, parece que a **renda anual** não apresenta grandes diferenças entre os grupos de quem comprou e quem não comprou, já que o objetivo é identificar fatores que influenciam a compra.

- **Tempo no Site:** A análise mostrou variações interessantes, com alguns usuários dedicando mais tempo navegando.



O gráfico acima mostra a distribuição do tempo no site, nota-se que entre 17 e 20 minutos, teve uma maior frequência do que outros.



Este **boxplot** compara o tempo no site (em minutos) entre dois grupos: usuários que não realizaram a compra (0) e os que realizaram a compra (1**)**. Este gráfico sugere uma relação positiva entre o tempo no site e a compra, mesmo a diferença não sendo muito grande.

2.2 Relações entre as Variáveis

- **Gênero e Compra:** Analisei como o gênero estava relacionado à decisão de compra, utilizando gráficos de barras.

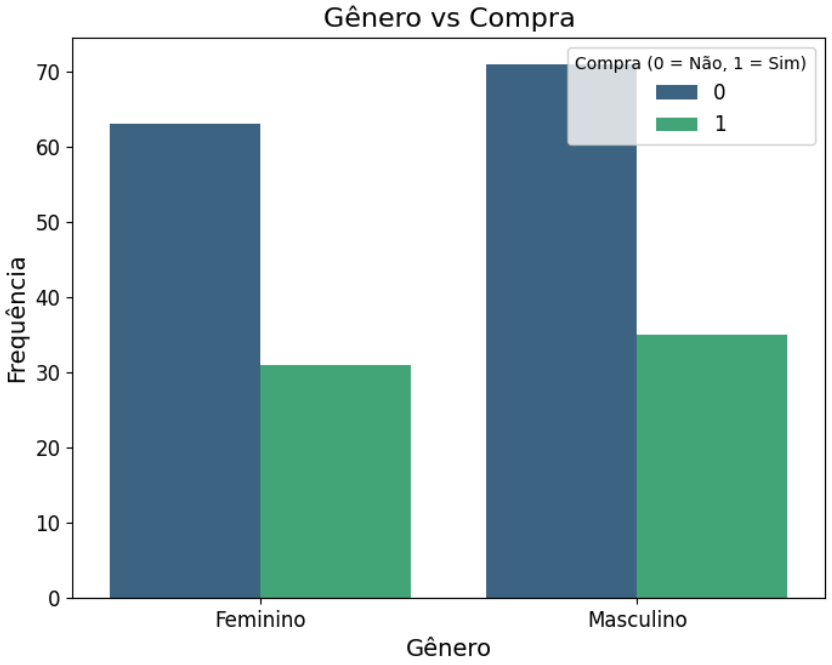
- **Tempo no Site e Compra:** Usuários que passaram mais tempo no site demonstraram uma maior tendência a comprar, o que ficou evidente nos gráficos de dispersão.

- **Renda e Compra:** A relação entre a renda anual e a compra foi explorada, mas os resultados não indicaram uma tendência muito forte.

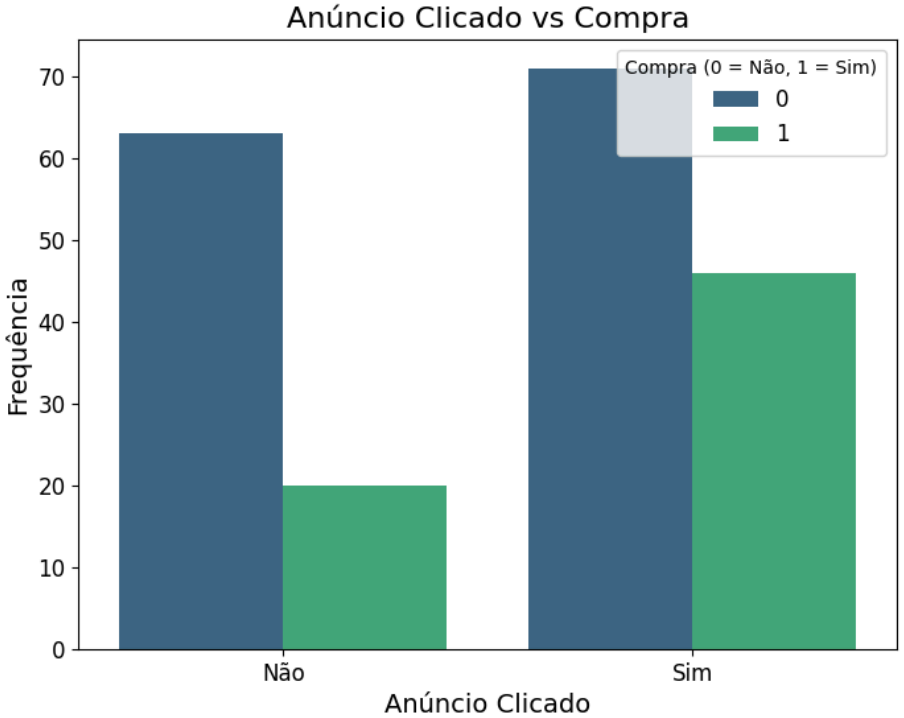
2.3 Correlação Entre Variáveis

Tempo no Site tem a correlação mais significativa com a variável alvo (**Compra**), o que era esperado dado o comportamento de navegação.

2.4 Análise das Variáveis Categóricas



O gráfico acima não apresenta uma diferença marcante entre os grupos. Fatores comportamentais ou econômicos (como tempo no site ou renda) podem ser mais influentes. Observa-se que a quantidade de usuários que não compraram (Mulheres) é maior do que os que compraram (Homens).



Este gráfico de barras mostra a relação entre o fato de o usuário ter clicado em um anúncio e a decisão de compra (compra = 1 ou não compra = 0). A análise revela o seguinte:

- **Usuários que não clicaram no anúncio:** A maioria deles não realizou a compra (classe 0), mas ainda há um número pequeno que efetuou a compra.

- **Usuários que clicaram no anúncio:** A proporção de compras (classe 1) aumenta significativamente, indicando que o clique em um anúncio está fortemente relacionado à probabilidade de compra.

Este gráfico sugere que o comportamento de clicar em anúncios pode ser um indicador importante para prever a decisão de compra. Isso reforça a relevância de campanhas publicitárias direcionadas para estimular interações no site.

2.4 Tratamento de valores ausentes e inconsistentes

Nas variáveis numéricas: idade e renda anual, apresentava valores faltantes, e isso poderia ter um grande impacto no treinamento do modelo. Então peguei a mediana de todos os dados presentes, e preenchi nos valores ausentes do dataset.

```
df['Idade'].fillna(df['Idade'].median(), inplace=True)
df['Renda Anual (em $)'].fillna(df['Renda Anual (em $)'].median(), inplace=True)
```

Nas variáveis categóricas: Gênero e Anúncio clicado, também possuía valores faltantes, então utilizei a moda (o que mais se repete), e preenchi nos valores ausentes do dataset.

```
df['Anúncio Clicado'].fillna(df['Anúncio Clicado'].mode()[0], inplace=True)
df['Gênero'].fillna(df['Gênero'].mode()[0], inplace=True)
```

2. Pré processamento dos dados

- **Normalização de Variáveis Numéricas:** Variáveis como **Idade**, **Renda Anual** e **Tempo no Site** foram normalizadas para garantir que estivessem na mesma escala.

```python
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
df[variaveis_numericas] = scaler.fit_transform(df[variaveis_numericas])
df[variaveis_numericas]
```

- **Codificação de Variáveis Categóricas:** Transformei as variáveis categóricas (**Gênero** e **Anúncio Clicado**) em valores numéricos utilizando codificação binária.

```python
from sklearn.preprocessing import LabelEncoder

label_encoder_genero = LabelEncoder()
df['Gênero'] = label_encoder_genero.fit_transform(df['Gênero'])

label_encoder_anuncio = LabelEncoder()
df['Anúncio Clicado'] = label_encoder_anuncio.fit_transform(df['Anúncio Clicado'])
```

- **Divisão do Conjunto de Dados:** Dividi os dados em conjuntos de treino e teste para avaliar o desempenho do modelo de maneira independente.

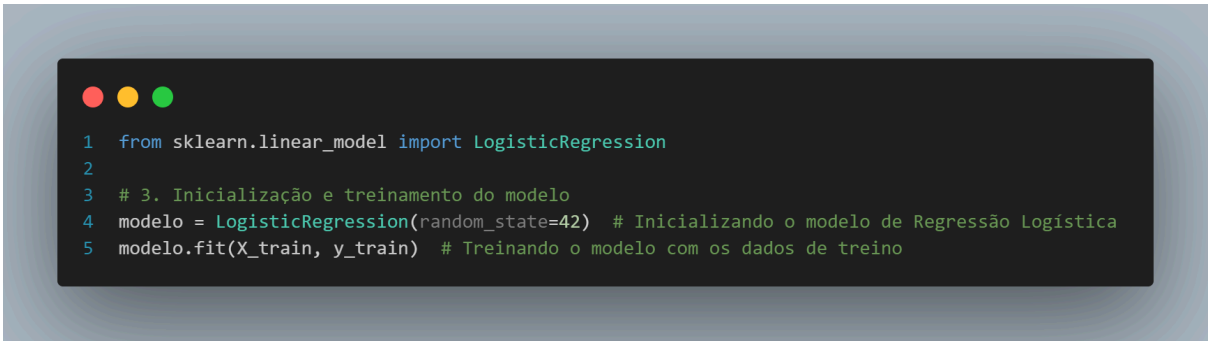
```python
from sklearn.model_selection import train_test_split

target = 'Compra (0 ou 1)'

X = df.drop(columns=['Compra (0 ou 1)'])
y = df[target]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)


X_train.shape, X_test.shape, y_train.shape, y_test.shape

```

3. Construção do Modelo

Escolhi a **Regressão Logística** como modelo inicial, devido à sua simplicidade e interpretabilidade. As principais etapas foram:

- Treinei o modelo no conjunto de treino utilizando as variáveis independentes normalizadas.
- Avaliei o desempenho no conjunto de teste para verificar sua capacidade de generalização.

3.1 Construção do Modelo sem Smote (balanceamento de dados)



```python
from sklearn.linear_model import LogisticRegression

# 3. Inicialização e treinamento do modelo
modelo = LogisticRegression(random_state=42)  # Inicializando o modelo de Regressão Logística
modelo.fit(X_train, y_train)  # Treinando o modelo com os dados de treino
```

Porém não tive bons resultados, aparentemente o modelo se adaptou bastante na variável alvo , causando overfitting. E por isso, gerou péssimos resultados.

```
Acurácia: 0.58
Matriz de Confusão:
[[35  0]
 [25  0]]
Relatório de Classificação:
              precision    recall  f1-score   support

           0       0.58      1.00      0.74        35
           1       0.00      0.00      0.00        25

    accuracy                           0.58        60
   macro avg       0.29      0.50      0.37        60
weighted avg       0.34      0.58      0.43        60
```

3.2 Construção do Modelo com Smote (balanceamento de dados)

Uma etapa importante para melhorar o desempenho do modelo foi o balanceamento dos dados utilizando a técnica SMOTE (Synthetic Minority Oversampling Technique). Como o dataset inicial apresentava um desequilíbrio entre as classes (muito mais registros de não compradores do que compradores), o SMOTE foi aplicado para gerar amostras sintéticas da classe minoritária e, assim, equilibrar o conjunto de dados.

Passos realizados:

- Apliquei o SMOTE ao conjunto de treino para balancear as classes.
- Reentrei o modelo utilizando os dados balanceados.

O balanceamento resultou em uma melhoria significativa na acurácia do modelo e em uma distribuição mais justa das predições entre as classes.

```python
from imblearn.over_sampling import SMOTE

smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X, y)

print(f"Distribuição após SMOTE: {pd.Series(y_resampled).value_counts()}")

```

```
Acurácia: 0.69
Matriz de Confusão:
[[17 14]
 [ 3 20]]
Relatório de Classificação:
              precision    recall  f1-score   support

           0       0.85      0.55      0.67        31
           1       0.59      0.87      0.70        23

    accuracy                           0.69        54
   macro avg       0.72      0.71      0.68        54
weighted avg       0.74      0.69      0.68        54
```

4. Extras: Implementações Adicionais e Insights Inovadores

Depois da utilização do Smote, o tempo que o usuário fica no site é o que teve maior coeficiente e quem clica no anúncio, também foi uma variável significativa

	Feature	Coefficient	Abs_Coefficient
3	Tempo no Site (min)	1.010686	1.010686
4	Anúncio Clicado	0.608446	0.608446
0	Idade	0.369455	0.369455
1	Renda Anual (em $)	-0.229817	0.229817
2	Gênero	-0.094900	0.094900

Nesta etapa, explorei algumas ideias para complementar a análise e melhorar os resultados:

- **Visualizações Dinâmicas:** Para facilitar a compreensão dos dados e resultados, gerei gráficos interativos usando Plotly, permitindo uma exploração mais detalhada das relações entre variáveis.

Essas implementações adicionais contribuíram para uma análise mais robusta e resultados mais confiáveis.

5. Resultados e Conclusões

O modelo apresentou desempenho razoável com uma acurácia de **69%** após o balanceamento com SMOTE. Observou-se que as variáveis relacionadas ao comportamento do usuário no site (é o caso do **Tempo no Site**) têm maior impacto na probabilidade de compra.

Pontos Positivos:

- Boa separação entre compradores e não-compradores.
- Identificação clara de variáveis-chave para futuras estratégia