descomplica
faculdade digital

Previsão de quais clientes estimulados pelo(s) anúncio(s) realizarão ou não uma compra

Pitch | Avaliação do módulo 3

SUZANA DE ARAUJO GOMES - MATRICULA-140308

PÓS-GRADUAÇÃO EM DATA SCIENCE

PROBLEMA

IDENTIFICAR QUAIS CLIENTES ESTIMULADOS PELO(S) ANÚNCIO(S) REALIZARÃO OU NÃO UMA COMPRA.

PÚBLICO-ALVO: profissionais de marketing, analistas de dados e gestores de campanhas publicitárias

PROPOSTA DE SOLUÇÃO

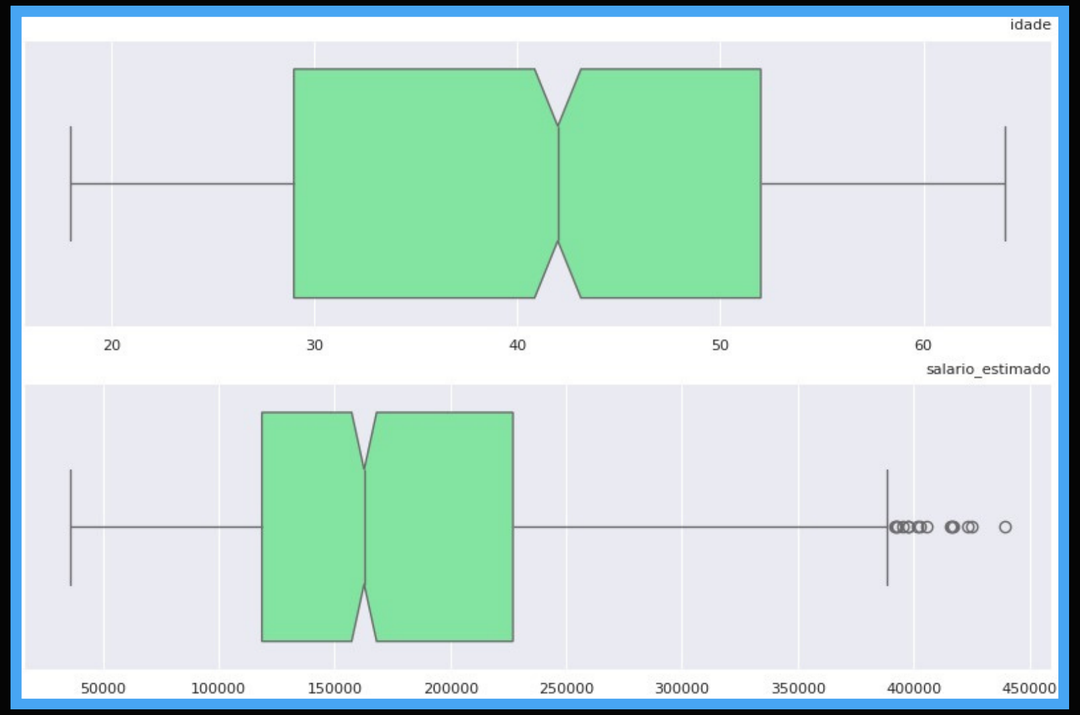
Ao identificar padrões de compra, criar um modelo preditivo do comportamento dos usuários em relação aos anúncios

1. **Correlação dos Dados dos Usuários**
2. **Geração de Modelo Preditivo**
3. **Simulação de Novas Entradas de Dados e Previsões**

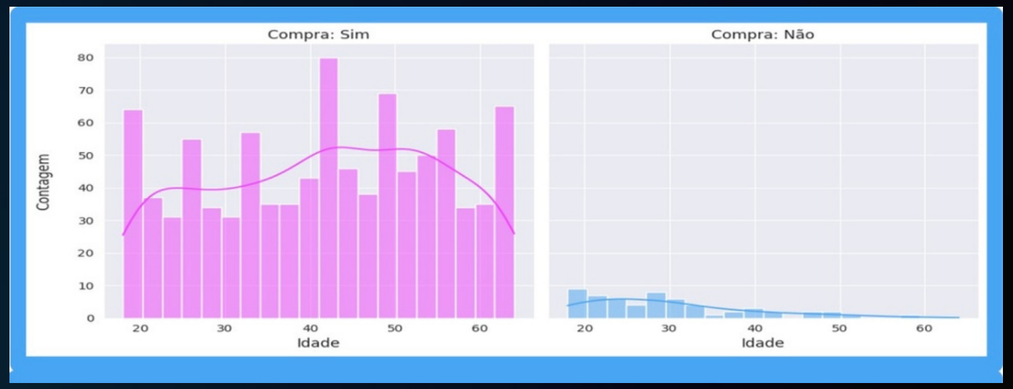
Um pouco do código python na prática

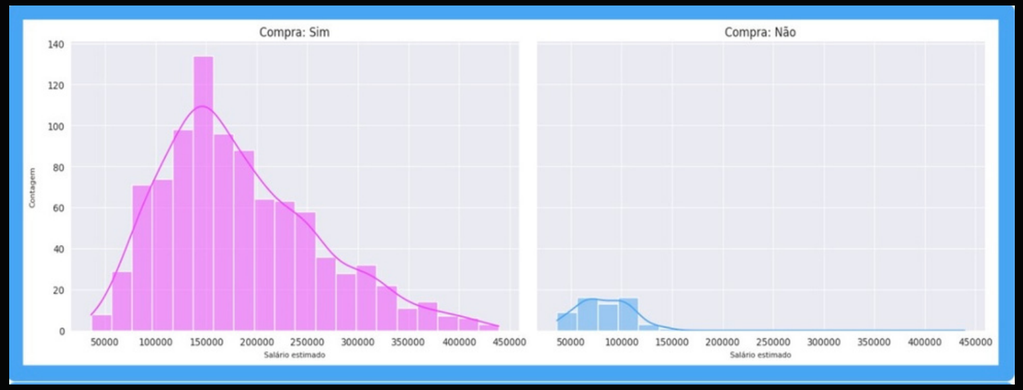
- Importação das bibliotecas e carregamento dos dados,verificação e ETL;
- Transformação dos valores da categoria gênero em categorias distintas usando o OneHotEnconder;
- Análise das medidas estatísticas dos dados

	genero	idade	salario_estimado	compra	feminino	masculino
0	Masculino	56	235755.137106	1	0.0	1.0
1	Masculino	46	158227.024343	1	0.0	1.0
2	Masculino	32	167283.053448	1	0.0	1.0
3	Feminino	60	252564.822559	1	1.0	0.0
4	Masculino	25	116576.501350	1	0.0	1.0

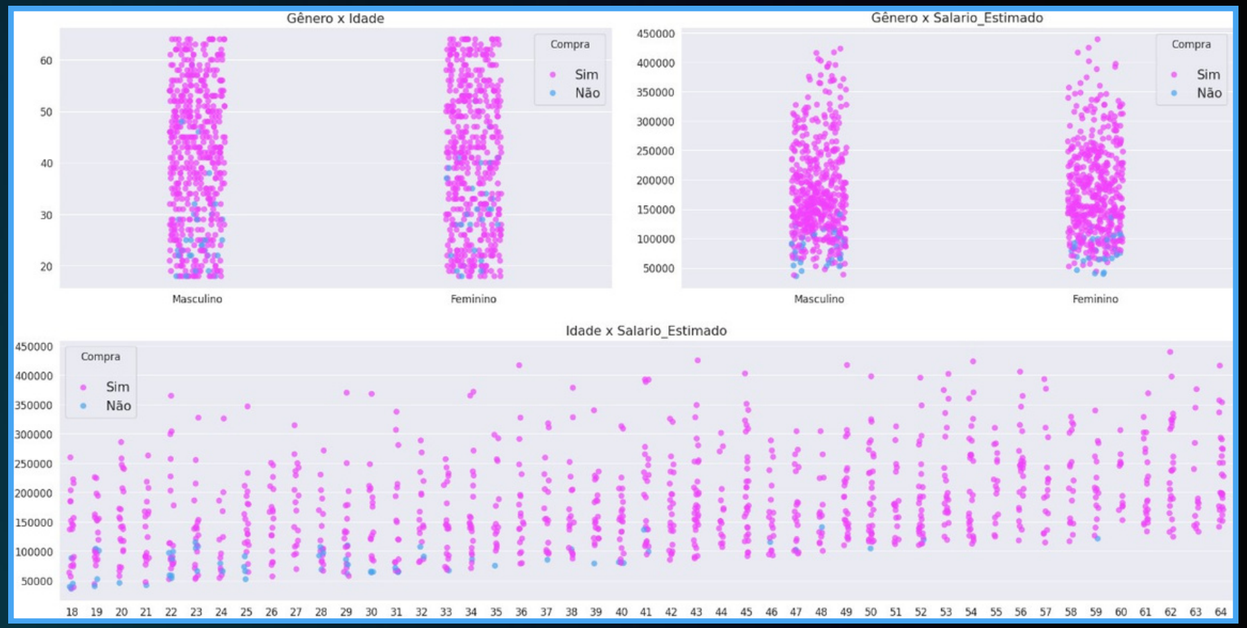


- Distribuição dos dados entre as variáveis





- Correlação entre as variáveis

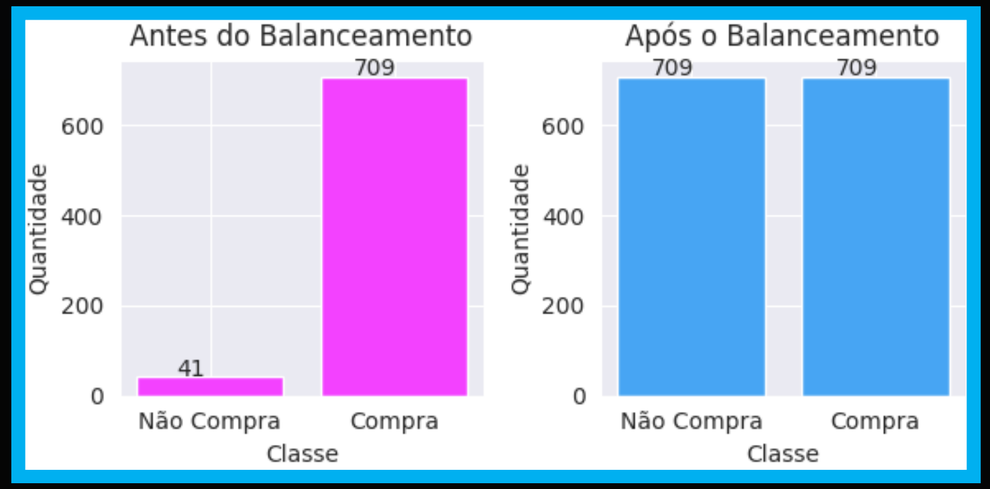


Correlação entre as variáveis

	idade	masculino	feminino	salario_estimado	compra
idade	1	0.0055	-0.0055	0.4	0.21
masculino	0.0055	1	-1	-0.011	-0.0011
feminino	-0.0055	-1	1	0.011	0.0011
salario_estimado	0.4	-0.011	0.011	1	0.3
compra	0.21	-0.0011	0.0011	0.3	1

- **Criação do modelo de regressão logística, aplicação do balanceamento, treinamento do modelo e acurácia do modelo**

```
# Features, variáveis independentes
X = df.drop(['compra', 'genero'], axis=1)

# Padronizando os dados das features
std = StandardScaler().fit(X)
X = std.transform(X)

# Target, variável dependente
y = df['compra'].copy()

# Definindo o percentual de dados para teste
perc = 0.25

# Definindo o random_state
seed = 10

# Separação dos dados de treino e teste
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=perc, random_state=seed)

print('\nQuantidade de dados divididos em treino e test
print('\nX_train:', X_train.shape)
print('y_train:', y_train.shape)
print('\nX_test:', X_test.shape)
print('y_test:', y_test.shape)
```



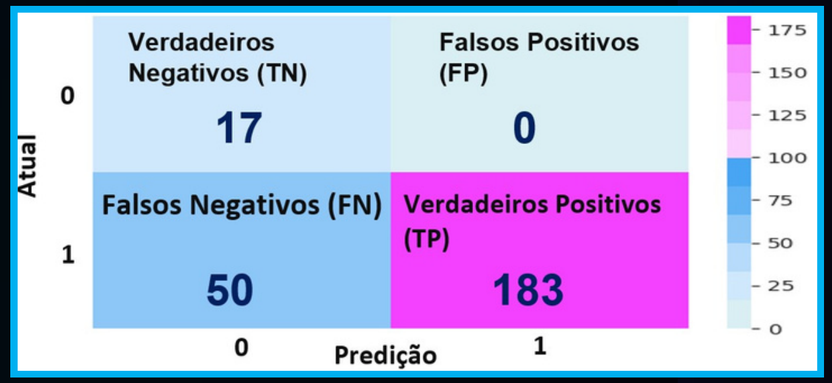

```
[27]  1 modelo_rl = LogisticRegression(solver='liblinear')
      2 print('\nModelo selecionado:', modelo_rl)

Modelo selecionado: LogisticRegression(solver='liblinear')
```

```
# treinamento do modelo
modelo_rl.fit(X_train_b, y_train_b)

# predição com dados de teste
pred_test = modelo_rl.predict(X_test)

pred_test #imprime as previsões feitas pelo modelo nos dados de tes

array([1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1,
       1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1,
       0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0,
       1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1,
       0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,
       1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0,
       1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1,
       0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1,
       1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1,
       0, 1, 1, 0, 1, 0, 1, 1])
```

750 amostras no conjunto de treinamento (X_train e
e 250 amostras no conjunto de teste (X_test e y_test
Cada amostra tem 4 features.

MÉTRICAS DO MODELO

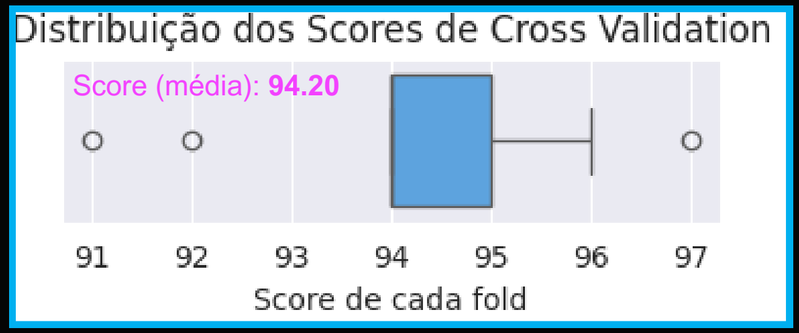
- Matriz Confusão



- Report Classification

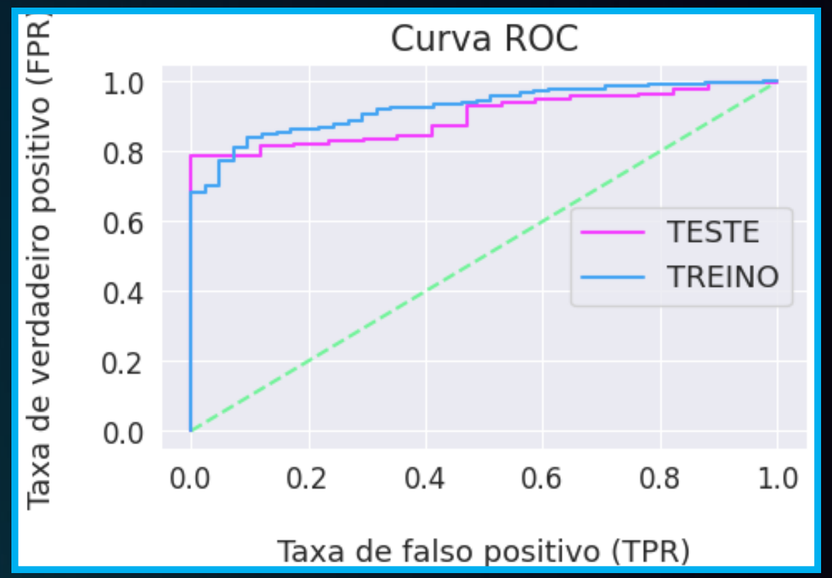
	precision	recall	f1-score	support
0	0.25	1.00	0.40	17
1	1.00	0.79	0.88	233
accuracy			0.80	250
macro avg	0.63	0.89	0.64	250
weighted avg	0.95	0.80	0.85	250

- Cross Validation (validação cruzada)

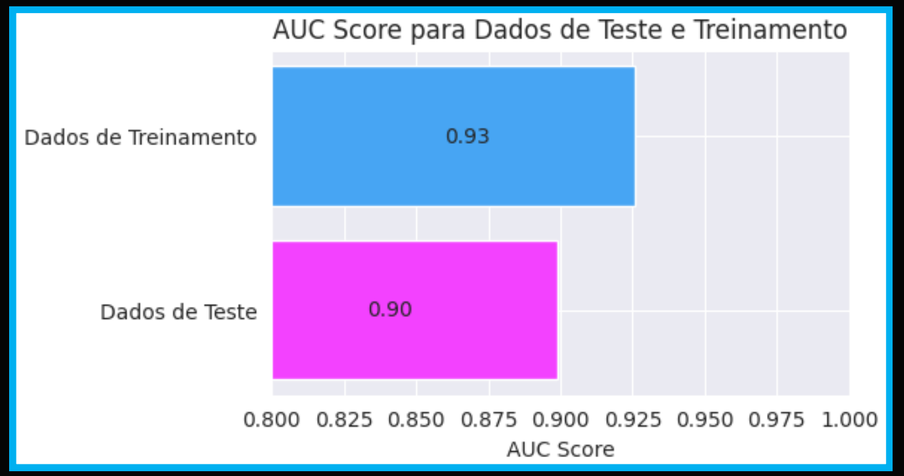


MÉTRICAS DO MODELO

- Curva ROC



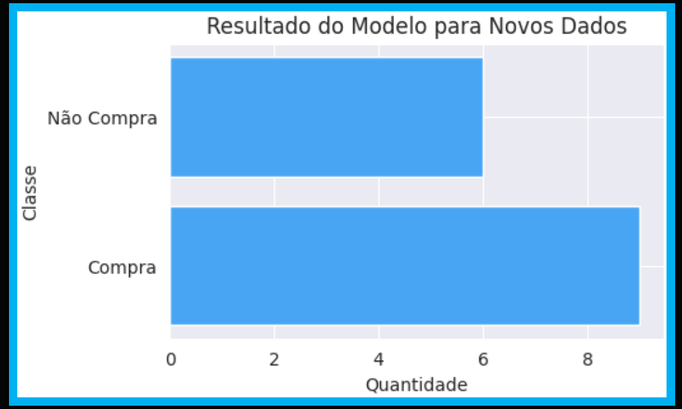
- Área sob a curva ROC (AUC)



- Aplicação do modelo de regressão logística a novos dados

```python
def teste_modelo(param1):

    arrx = []

    for i in range(param1):
        idade = np.random.randint(df['idade'].min(), df['idade'].max())
        masculino = np.random.randint(2)
        if masculino == 1:
            feminino = 0
        else:
            feminino = 1
        salario_estimado = np.random.randint(
            df['salario_estimado'].min(), df['salario_estimado'].max())
        arr = np.array([idade, masculino,
                        feminino, salario_estimado])
        arrx.append(arr)

    novos_X = np.array(arrx)
    std = StandardScaler().fit(novos_X)
    novos_X = std.transform(novos_X)
    novos_Y = modelo_rl.predict(novos_X)

    df_tab = pd.DataFrame(arrx)
    df_Y = pd.DataFrame(novos_Y)
    df_tab[4] = df_Y[0]

    print(tabulate(df_tab, headers=['idade', 'masculino',
                                    'feminino', 'salario_estimado', 'compra'],
                   tablefmt='fancy_grid',
                   floatfmt=('.0f', '.0f', '.0f', '.0f', '.2f', '.0f')))
```

	idade	masculino	feminino	salario_estimado	compra
0	51	1	0	175533.00	1
1	44	1	0	429900.00	1
2	24	1	0	257764.00	1
3	38	1	0	437838.00	1
4	40	0	1	264186.00	0
5	38	0	1	336586.00	0
6	18	1	0	382250.00	1
7	58	0	1	278835.00	0
8	62	0	1	190701.00	0
9	61	1	0	349293.00	1
10	23	1	0	369171.00	1
11	54	1	0	96651.00	1
12	58	0	1	316324.00	0
13	55	1	0	71513.00	1
14	18	1	0	190070.00	1



SOLUÇÃO NA PRÁTICA

- A solução prática consistiu na implementação do modelo de Regressão Logística que alcançou uma acurácia de **80%**, evidenciando sua eficácia na previsão de comportamentos de compra dos usuários. Além disso, a pontuação média de **94.20**, obtida por meio de Cross Validation, confirma a robustez do modelo.

- A aplicação do modelo não se limitou à **previsão de compras**, mas também permitiu realizar testes com novas entradas de dados. Ao simular **potenciais clientes** estimulados por um anúncio específico, o modelo destacou-se ao fornecer informações valiosas sobre usuários mais propensos a realizar uma compra.

- Essa abordagem contribui para uma **tomada de decisão** informada na alocação de recursos de marketing, maximizando o impacto das campanhas e aumentando a eficiência das estratégias publicitárias.

descomplica
faculdade digital

OBRIGADA !

Alguma dúvida?

SUZANA DE ARAUJO GOMES
suagomes@gmail.com

Pós-Graduação em Data Science

CÓDIGO PYTHON
TODO COMENTADO