

Introdução

Este projeto documenta o processo de análise de um conjunto de registros de vendas de um restaurante. O objetivo central foi transformar este grande volume de dados brutos em inteligência de negócio acionável.

Para isso, utilizamos técnicas de Machine Learning para segmentar automaticamente os perfis de negócio, construir um modelo preditivo com alta precisão e, finalmente, simular o impacto financeiro de estratégias de marketing.

A finalidade é demonstrar como uma aplicação orientada a dados pode capacitar um comerciante a abandonar a intuição e tomar decisões que comprovadamente aumentam sua lucratividade.

O motivo das análises

No competitivo mercado de restaurantes, decisões sobre cupons e promoções são frequentemente tomadas com base no achismo. A análise foi motivada pela necessidade de responder, com dados, a perguntas críticas para qualquer comerciante:

- Meus cupons de fidelidade estão valendo a pena ou só comendo minha margem?"
- Qual estratégia de venda realmente traz mais lucro no final do mês?
- Como posso saber se uma nova promoção vai funcionar antes de gastar dinheiro com ela?

O objetivo era substituir a incerteza por evidências concretas, provando que é possível encontrar a estratégia de lucro ótima analisando o próprio histórico de vendas do negócio.

Algoritmos utilizados

Para construir essa inteligência, combinamos duas técnicas de Machine Learning, cada uma com um papel fundamental no projeto:

Agrupamento K-Means Clustering: Primeiramente, usamos este algoritmo de aprendizado não supervisionado para analisar as características de todas as vendas e descobrir, de forma automática, 3 segmentos de negócio distintos. Ele nos permitiu

organizar o caos e entender que existiam diferentes tipos de "jogadas" estratégicas ocorrendo.

Régressão Linear: Após a segmentação, aplicamos este algoritmo de aprendizado supervisionado para criar um modelo preditivo. Ele foi treinado com 30.000 exemplos de vendas para aprender a equação do lucro mais próxima da resposta do negócio e alcançou uma precisão de 90.43%

```
--- Avaliação do Desempenho do Modelo ---
Erro Médio Absoluto (MAE): R$ 5.18
Coeficiente de Determinação (R2): 90.43%
```

Tornando-se nosso confiável simulador de estratégias que torna as dúvidas dos comerciantes sem precisar quebrar a cabeça, e ficar no achismo se daria certo ou não.

```
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, r2_score

arquivo = 'vendas.csv'

try:
    df = pd.read_csv(arquivo)
    print("Arquivo lido")

except FileNotFoundError:
    print(f"Erro: Arquivo '{arquivo}' não encontrado.")
```

```
#K-Means
from sklearn.preprocessing import StandardScaler
import pandas as pd
from sklearn.cluster import KMeans

# Selecionar as colunas
features = df[['valor_venda', 'tipo_cupom', 'plataforma']]
features_encoded = pd.get_dummies(features, columns=['tipo_cupom', 'plataforma'])
scaler = StandardScaler()
features_scaled = scaler.fit_transform(features_encoded)

print("Dados convertidos")
```

```

numero_de_grupos = 3

kmeans = KMeans(n_clusters=numero_de_grupos, n_init='auto', random_state=42)
kmeans.fit(features_scaled)

df['Grupo'] = kmeans.labels_
print(df.head())

print("Análise Detalhada de Cada Grupo Encontrado")

características_grupo = df.groupby('Grupo').agg({
    'valor_venda': 'mean',
    'tipo_cupom': lambda x: str(x.mode().iat[0]) if not x.mode().empty else None,
    'plataforma': lambda x: str(x.mode().iat[0]) if not x.mode().empty else None
}).rename(columns={'valor_venda': 'Valor Médio da Venda', 'tipo_cupom': 'Cupom Mais Usado', 'plataforma': 'Plataforma Dominante'})

características_grupo['Valor Médio da Venda'] = características_grupo['Valor Médio da Venda'].round(2)

print(características_grupo)

print("\nVisualização Gráfica dos Grupos")
fig, axes = plt.subplots(1, 3, figsize=(20, 6))
fig.suptitle('Características Visuais dos Grupos Encontrados', fontsize=16)

# Valor Médio da Venda
sns.barplot(data=df, x='Grupo', y='valor_venda', ax=axes[0], palette='viridis')
axes[0].set_title('Valor Médio da Venda por Grupo')

# Plataforma mais usada
sns.countplot(data=df, x='Grupo', hue='plataforma', ax=axes[1])
axes[1].set_title('Distribuição de Plataformas por Grupo')

# Tipo de cupom mais usado
sns.countplot(data=df, x='Grupo', hue='tipo_cupom', ax=axes[2])
axes[2].set_title('Distribuição de Cupons por Grupo')

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()

```

```

mapa_de_perfis = {
    0: 'Nicho Premium',
    1: 'Campeão de Volume',
    2: 'Equilíbrio Local'
}

df['Perfil do Restaurante'] = df['Grupo'].map(mapa_de_perfis)
df.drop('Grupo', axis=1, inplace=True)

print("Nomes dos perfis aplicados com sucesso")
print(df.head())

```

```
features = df[['Perfil do Restaurante', 'tipo_cupom', 'plataforma', 'valor_venda']]
target = df['lucro']
X = pd.get_dummies(features)
y = target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)

print("Dados preparados e divididos para o modelo de previsão")
print(f"São {X_train.shape[0]} dados devendas para treinar e {X_test.shape[0]} para testar.")

modelo_preditivo = LinearRegression()
modelo_preditivo.fit(X_train, y_train)
previsoes = modelo_preditivo.predict(X_test)

mae = mean_absolute_error(y_test, previsoes)
r2 = r2_score(y_test, previsoes)

print("\nAvaliação do Desempenho do Modelo")
print(f"Erro Médio Absoluto (MAE): R$ {mae:.2f}")
print(f"Coeficiente de Determinação (R²): {r2:.2%}")

df_comparacao = pd.DataFrame({'Lucro Real': y_test.head(), 'Lucro Previsto': previsoes[:5]})
print("\nComparando Lucro Real vs. Lucro Previsto")
print(df_comparacao.round(2))
```

```

def simular_cenario(perfil, cupom, plataforma, valor_venda):
    dados_simulacao = pd.DataFrame({
        'Perfil do Restaurante': [perfil],
        'tipo_cupom': [cupom],
        'plataforma': [plataforma],
        'valor_venda': [valor_venda]
    })
    simulacao_encoded = pd.get_dummies(dados_simulacao)
    simulacao_final = simulacao_encoded.reindex(columns=X_train.columns, fill_value=0)
    lucro_previsto = modelo_preditivo.predict(simulacao_final)
    return lucro_previsto[0]

print("Simulando Estratégia")

lucro_cenario_1 = simular_cenario('Equilíbrio Local', 'nenhum', 'Rappi', 120.00)
print(f"\nCenário sem cupom:")
print(f"Lucro Previsto por venda: R$ {lucro_cenario_1:.2f}")

lucro_cenario_2 = simular_cenario('Equilíbrio Local', 'fidelidade_20_reais', 'Rappi', 120.00)
print(f"\nCenário com cupom:")
print(f"Lucro Previsto por venda: R$ {lucro_cenario_2:.2f}")

diferenca = lucro_cenario_2 - lucro_cenario_1
print("\nConclusão da Simulação")
if diferenca > 0:
    print(f"Implementar o cupom de fidelidade pode AUMENTAR o lucro por venda em aproximadamente R$ {diferenca:.2f}!")
else:
    print(f"Implementar o cupom de fidelidade pode REDUZIR o lucro por venda em aproximadamente R$ {abs(diferenca):.2f}.")

dados_grafico3 = {
    'Tipo de Cliente': ['Cliente Comum', 'Cliente Fidelizado'],
    'Valor Total Gerado ($)': [88.59, 201.70]
}
df_grafico3 = pd.DataFrame(dados_grafico3)

# Criar o gráfico
plt.figure(figsize=(10, 7))
sns.set_style("whitegrid")
bars = sns.barplot(x='Tipo de Cliente', y='Valor Total Gerado (R$)', data=df_grafico3, palette=['#EA4335', '#34A853'])

plt.title('Gráfico 3: O Veredito - Qual Cliente Vale Mais de R$200?', fontsize=18, fontweight='bold')
plt.xlabel('')
plt.ylabel('Valor Total Gerado por Cliente (LTV)', fontsize=12)
plt.ylim(0, 250)

for bar in bars.patches:
    bars.annotate(f'R$ {bar.get_height():.2f}', 
                  (bar.get_x() + bar.get_width() / 2, bar.get_height()),
                  ha='center', va='center', size=14, xytext=(0, 8),
                  textcoords='offset points', fontweight='bold', color='black')

plt.suptitle('Nós transformamos seus dados de venda em lucro.', fontsize=12, y=0.98)
plt.figtext(0.5, 0.05, "E é EXATAMENTE isso que nosso aplicativo faz por você: analisa seus dados 24/7 para encontrar essas oportunidades preditivas.\nEle te diz qual estratégia usar, para qual perfil de cliente, para maximizar seu lucro total.", 
            ha="center", fontsize=12, style='italic')

plt.subplots_adjust(top=0.88, bottom=0.25)
plt.show()

```

Resultados da análise

A aplicação dos modelos gerou um insight estratégico fundamental. Ao simular o impacto de um cupom de fidelidade, obtivemos a seguinte descoberta:

Lucro Previsto por venda (Sem Cupom): R\$ 47,89

Lucro Previsto por venda (Com Cupom de Fidelidade): R\$ 39,94

```
Cenário sem cupom:  
-> Lucro Previsto por venda: R$ 47.89
```

```
Cenário com cupom:  
-> Lucro Previsto por venda: R$ 39.94
```

Isso mostra que, em uma venda isolada, o cupom de fato reduz o lucro imediato em R\$ 7,95. Contudo, a análise do comportamento histórico revelou o outro lado da moeda:

```
Conclusão da Simulação  
Implementar o cupom de fidelidade pode REDUZIR o lucro por venda em aproximadamente R$ 7.95.
```

Média de Compras (Clientes Comuns): 1.85 compras

Média de Compras (Clientes Fidelizados): 5.05 compras

```
Número médio de compras feitas por tipo de cliente:  
Grupo de Cliente  
Nunca Usou 1.85  
Usa Fidelidade 5.05
```

Fica evidente que o que se perde em uma única venda é massivamente compensado pela frequência. O cliente fidelizado compra quase 3 vezes mais.

Conclusão sobre a finalidade do projeto

A finalidade deste projeto foi alcançada com sucesso: provar que é possível e altamente lucrativo usar os próprios dados de um negócio para guiar suas estratégias de crescimento.

Concluímos que a estratégia de cupons de fidelidade é um investimento inteligente. A análise demonstra que a perda de margem no curto prazo é um pequeno preço a se pagar para adquirir um cliente que, a longo prazo, é exponencialmente mais valioso para o negócio.

O projeto valida, portanto, o valor de uma aplicação que automatiza essa análise. A ferramenta não é apenas um relatório, mas um guia estratégico contínuo que permite ao comerciante simular o futuro, entender o valor real de suas ações e, finalmente, tomar decisões que maximizam não apenas a venda de hoje, mas a lucratividade de amanhã.

