



Análise Case Técnico

Teste A/B Cupom de Desconto

Contextualização



Este documento descreve análise de dados realizada sobre Teste A/B e Cupons de Desconto. A Seguir serão descritas as ferramentas e metodologias utilizadas.

Ferramentas e Metodologia utilizadas



Para análise do case técnico proposto foi utilizado Databricks e os datasets foram lidos do S3 conforme links compartilhados com exceção para o dataset `ab_test_ref` que por limitações do spark precisou ser descompactado localmente e armazenado no DBFS File Browser. No notebook compartilhado os primeiros comandos referem-se à leituras destes datasets.

Com o objetivo de tornar a manipulação desses dados mais ágil, foram escritas as tabelas `orders`, `consumers`, `merchants` e `ab_test_users` no database também criado (`analytics`).

Os dados foram manipulados utilizando funções PySpark e bibliotecas python específicas.

A partir da célula de comando 20 tem-se as análises realizadas.

1. Principais Indicadores

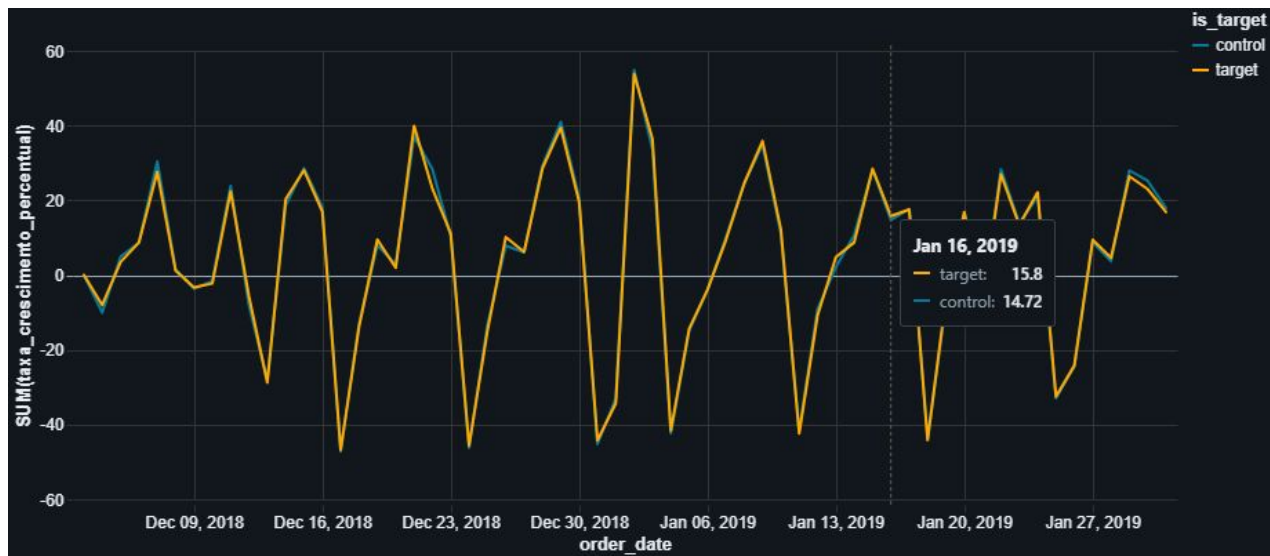
Como indicadores relevantes para análise do teste A/B referente a campanha de cupom de desconto foram escolhidos:

- $\% \text{ Pedidos} = \text{Total de pedidos de clientes com cupom} / \text{total de pedidos}$;
- Crescimento diário de pedidos (comparativo entre as segmentações);
- Valor médio de pedido = Valor médio de pedido de clientes que utilizam cupom;
- Mediana diária do valor total de pedidos (comparativos entre as segmentações);
- Análise de Significância Estatística

1.1 - Percentual de Pedidos e Percentual de Crescimento Diários

Analisando a quantidade distinta de pedidos para o segmento target, ou seja, que receberam cupom de desconto, tem-se que este segmento possui 58,23% dos pedidos, o que pode levar à um bom indício em relação à adesão à campanha de cupons.

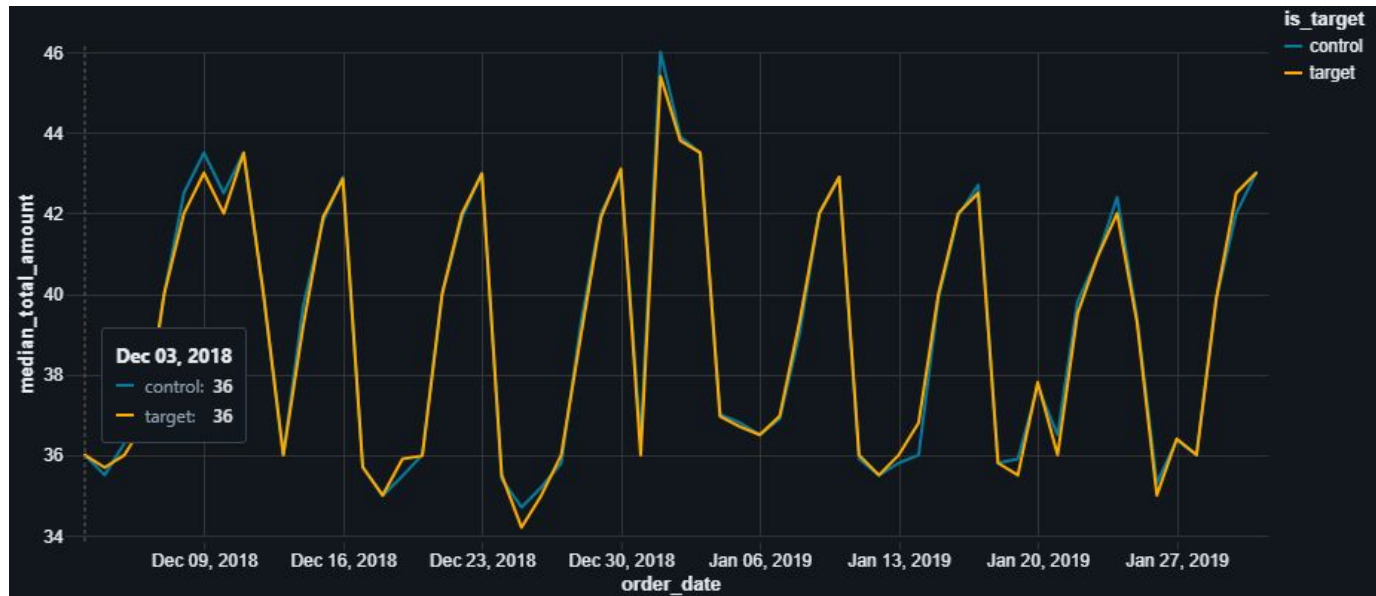
Em relação ao percentual de crescimento diário de pedidos, quando comparados os dois segmentos é possível observar que apresentam valores muito próximos e oscilam entre percentuais crescimento e quedas, conforme gráfico a seguir.



1.2 - Valor Médio de Pedidos e Mediana Diária do Valor dos Pedidos

Em relação ao valor médio de pedidos (`order_total_amount`), para o segmento target este valor é de 47.81, já para o control é de 47.92, ou seja, embora o segmento target tenha o maior percentual de pedidos, em termos de valores, os valores médios são muito próximos para os segmentos.

A mediana diária dos valores de pedidos por segmento foi analisada e em geral, os valores são próximos conforme gráfico a seguir.



1.3 - Significância Estatística e Média de `amount_order` por cliente e análise de performance do segmento com cupom

Para assegurar que os resultados do teste A/B refletem de fato o melhor desempenho de um dos segmentos e não apenas variação natural dos dados, aplica-se o cálculo de Significância Estatística.

Logo, considera-se nível de confiança de 95% e significância (p) de 5%, ou seja, a Significância pode ser entendida como a probabilidade dos resultados apresentados serem por acaso.

Desta forma:

p-valor - Baixo (< 0.05): A diferença observada provavelmente não é por acaso

p-valor - Alto (≥ 0.05): A diferença pode ter ocorrido por variação natural

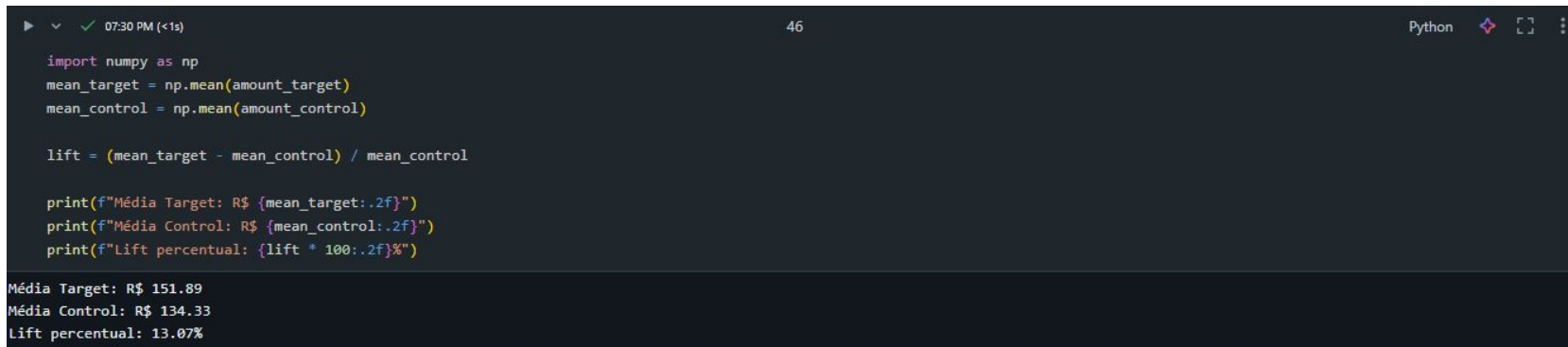
Para calcular a Significância Estatística para os dados referentes ao teste A/B deste case técnico foram obtidos conjuntos de valores para pedidos por clientes e `order_amount` por clientes. Estes conjuntos de valores foram transformados em listas utilizando a biblioteca Pandas e testes estatísticos da biblioteca SciPy (`ttest_ind()` e `mannwhitneyu()`). O módulo `ttest_ind()` é utilizado para dados próximos com variância parecida e por isso foi utilizado para analisar quantidade de pedidos, já `mannwhitneyu()` é um módulo utilizado para conjunto de dados com outliers e por isso foi aplicado para `order_amount`.

Os valores obtidos foram próximos de zero indicando que o desempenho para clientes com cupom de fato reflete desempenho real e não apenas um resultado aleatório.

1.3 - Significância Estatística e Média de amount_order por cliente e análise de performance do segmento com cupom

Utilizando a biblioteca Pandas foi calculado a média de order_amount por cliente para ambos segmentos e o Lift percentual, ou seja, o percentual de performance do segmento com cupom em relação ao segmento sem cupom.

A imagem a seguir ilustra o resultado obtido através da execução do script disponibilizado no notebook compartilhado para análise deste case.



```
import numpy as np
mean_target = np.mean(amount_target)
mean_control = np.mean(amount_control)

lift = (mean_target - mean_control) / mean_control

print(f"Média Target: R$ {mean_target:.2f}")
print(f"Média Control: R$ {mean_control:.2f}")
print(f"Lift percentual: {lift * 100:.2f}%")
```

Média Target: R\$ 151.89
Média Control: R\$ 134.33
Lift percentual: 13.07%

O segmento com cupom apresenta percentual de 13.07% acima em relação ao segmento que não possui cupom em relação a média de valor de pedido por cliente.

2 - Segmentação, aplicação de novo Teste de Significância e Análise de Performance

Para nova segmentação foram utilizadas variáveis numéricas atreladas a cada cliente (customer_id) que encontra-se ativo e possui pelo menos um pedido atrelado a um restaurante também ativo.

As variáveis numéricas utilizadas foram:

- Quantidade de pedidos (quant_order);
- Valor total dos pedidos (total_order);
- Média do valor mínimo dos restaurantes atrelados aos pedidos (minimum_order_value);
- Ticket médio do restaurante (average_ticket)

Foi utilizada a biblioteca scikit-learn e o módulo de K-Means para clusterização com base nas variáveis numéricas escolhidas. Foram obtidas as quantidade de clientes por cluster (segmento) e as características destes clusters em relação às variáveis escolhidas. A imagem a seguir reflete a aplicação de K-Means:

2 - Segmentação, aplicação de novo Teste de Significância e Análise de Performance

```
▶ 08:38 PM (3s) 58

#Quantidade de clientes por cluster
print(df_clusters["cluster"].value_counts())

0    641930
1     61544
Name: cluster, dtype: int64

▶ 08:27 PM (3s) 59

# Características por cluster
print(df_clusters.groupby("cluster").mean())
```

	quant_order	total_order	minimum_order_value	average_ticket
cluster				
0	1.758701	80.527619	88.602909	57.987138
1	9.144157	455.132043	94.654348	59.504087

K-Means aplicou a segmentação com base em maior impacto em termos de variáveis escolhidas, logo a segmentação possui maior número de clientes em segmento de menor impacto.

O cluster 1 foi considerado como segmento alvo para a aplicação de cupons e foram recuperados os IDs (customer_id) referentes à clientes deste cluster. Em seguida foi realizado cruzamento com a tabela de pedidos para verificar o impacto desta segmentação.

2 - Segmentação, aplicação de novo Teste de Significância e Análise de Performance

O novo teste de significância foi aplicado e o resultado obtido foi positivo, indicando que resultados obtidos não aleatórios.

Em seguida, foram calculados nova Média do valor total de pedido por cliente e Lift Percentual para compreender o quanto o grupo com cupons performou melhor em termos de receita em relação ao grupo sem cupons. A imagem a seguir descreve os resultados obtidos:

2 - Segmentação, aplicação de novo Teste de Significância e Análise de Performance

```
▶ 09:04 PM (3s) 67 Python

stat, p_value = mannwhitneyu(amount_target, amount_control, alternative='two-sided')

print(f"Valor-p: {p_value:.4f}")
if p_value < 0.05:
    print("Diferença estatisticamente significativa na receita por cliente.")
else:
    print("Sem diferença significativa na receita por cliente.")

Valor-p: 0.0000
Diferença estatisticamente significativa na receita por cliente.
```

Show preview B I <> ↺ ↻ H1 H2 H3 ☰ ☲

Markdown

%md
Nova Media do valor total de pedido por cliente e Lift Percentual para compreender o quanto o grupo B performou melhor em termos de receita em relação ao grupo A

```
▶ 09:07 PM (3s) 69

mean_new_target = np.mean(amount_new_target)
mean_new_control = np.mean(amount_new_control)

lift = (mean_new_target - mean_new_control) / mean_control

print(f"Média novo Target: R$ {mean_new_target:.2f}")
print(f"Média novo Control: R$ {mean_new_control:.2f}")
print(f"Lift percentual: {lift * 100:.2f}%")

Média novo Target: R$ 624.01
Média novo Control: R$ 135.78
Lift percentual: 363.45%
```

3 - Próximos Passos

Com base nos resultados obtidos e com objetivo de que sejam implementadas soluções semelhantes deve-se:

- Verificar a base de clientes e realizar limpeza dos dados removendo clientes que não estão ativos e restaurantes inativos;
- Aplicar técnicas como K-means com base análise de variáveis numéricas do conjunto de dados para nova clusterização;
- Após nova clusterização definir as métricas chave que serão acompanhadas durante a realização de novo Teste A/B;
- Conforme demonstrado, utilizando K-Means e segmentando de forma correta a performance em termos de receita de clientes com cupom de desconto, pode alcançar bons percentuais. é importante mensurar o ROI para esta implementação.

Referências



A/B Testing: A Complete Guide to Statistical Testing, disponível em
<<https://medium.com/data-science/a-b-testing-a-complete-guide-to-statistical-testing-e3f1db140499>>;

O que é teste A/B, o que você pode testar e como começar a fazer, disponível em
<<https://www.rdstation.com/blog/marketing/o-que-e-teste-ab/>>.