

Análise das Vendas - Checkout 01

O gráfico compara as vendas de hoje, ontem e as médias da última semana e do último mês, mostrando padrões interessantes ao longo do dia.

Tendências Gerais

- **Média semanal e mensal:** As vendas começam a subir a partir das **05h** e atingem um pico estável por volta das **11h**. Depois disso, elas caem de forma gradual, com pequenos picos por volta das **14h-15h**. O comportamento é mais estável ao longo do dia, sem grandes flutuações.
- **Hoje e Ontem:** Em ambos os dias, vemos um comportamento atípico. Entre **07h e 08h**, há uma queda seguida por um pico acentuado às **09h** — muito mais brusco do que nas médias. Isso sugere que algum evento ou promoção pode estar afetando as vendas nesses dias.

Instabilidade Durante a Tarde

A partir das **10h**, as vendas de hoje e ontem mostraram **muita variação**, com picos e quedas ao longo do dia. Isso contrasta com as médias semanais e mensais, que são bem mais estáveis. Essas oscilações podem ser causadas por eventos externos ou até por algo relacionado ao sistema de vendas, como instabilidades e vendas represadas.

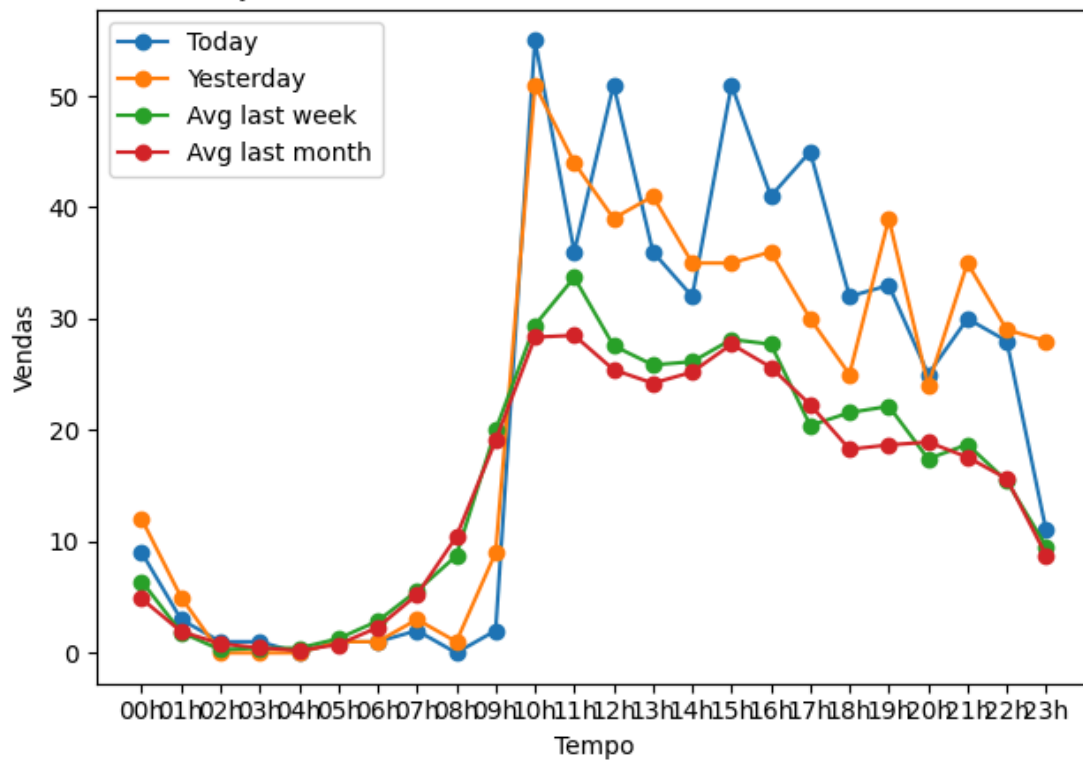
Queda no Final do Dia

Hoje, em particular, termina com uma queda nas vendas muito mais forte que nos outros períodos. Isso pode indicar o fim de algum evento especial que movimentou as vendas pela manhã e no dia de ontem.

Conclusão

Os melhores horários para vendas são entre **07h e 11h**, com uma boa performance à tarde, entre **14h e 16h**. A oscilação nos dias de hoje e ontem indica que algo fora do comum está afetando o padrão de vendas, e vale a pena investigar o que causou esse comportamento atípico.

Hoje x médias de última semana e mês (checkout01)



Análise das Vendas - Checkout 02

O gráfico mostra as vendas de hoje, ontem e as médias da semana e do mês. É possível notar alguns padrões e anomalias importantes.

Tendências Gerais

- **Médias semanais e mensais:**
 - As vendas começam a subir a partir das **05h**, com um pico estável às **10h**, e depois caem de forma constante até o final do dia. Esse é o comportamento mais típico em dias normais.
- **Ontem:**
 - As vendas começaram mais fortes entre **00h e 01h** do que nos outros dias.
 - O pico das **10h** foi maior que o de hoje e das médias. Durante o dia, há vários altos e baixos, o que pode indicar problemas ou flutuações causadas por eventos. Um ponto crítico foi às **08h**, quando as vendas zeraram — algo incomum e que pode indicar uma possível falha no sistema.

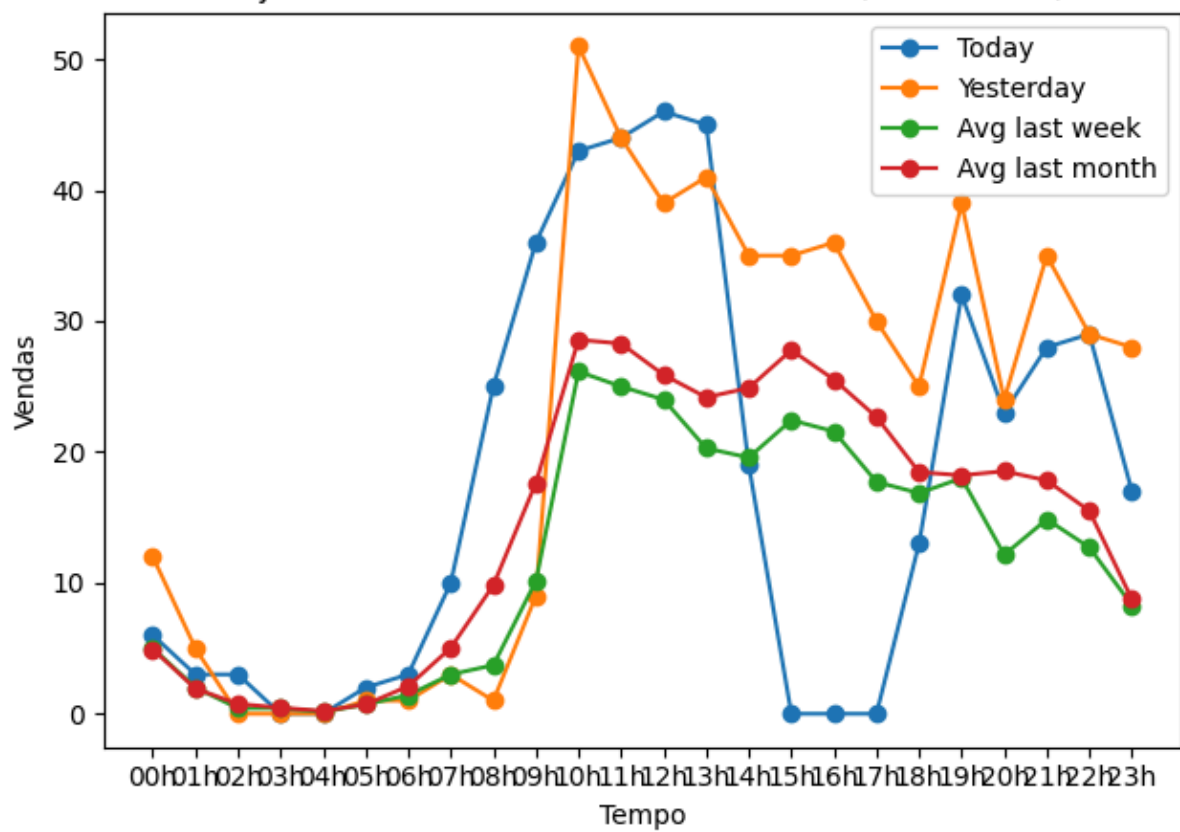
Comportamento de Hoje

- **Manhã:**
 - As vendas começaram fortes desde as **05h** e, embora o pico das **09h** tenha sido menor que o de ontem, as vendas se mantiveram boas no total, tendo vendas em grande volume com mais frequência e sem grandes picos.
- **Tarde (15h-17h):**
 - Um ponto de atenção foi entre **15h e 17h**, quando as vendas zeraram, sugerindo um problema sério no sistema. Depois disso, houve uma recuperação com um pico às **19h**, provavelmente com vendas represadas.

Conclusão

- As vendas de hoje e ontem foram instáveis, com quedas e picos acentuados, especialmente à tarde. A interrupção completa nas vendas entre **15h e 17h** precisa ser investigada com urgência, pois não é um comportamento normal.

Hoje x médias de última semana e mês (checkout02)



Contexto e Análise Técnica

Os dois conjuntos de dados (transactions_1 e transactions_2) representam transações categorizadas por status (aprovadas, negadas, revertidas, entre outros) ao longo de um período de tempo, com detalhes sobre o volume de cada status em determinados intervalos.

Análise dos Dados

1. Limpeza de Outliers:

- Antes de realizar qualquer análise, houve a necessidade de remover outliers dos dados para evitar que valores extremos distorçam os resultados.
- Isso foi feito utilizando o desvio padrão como base para identificar valores que fogem à média. Com essa abordagem, a análise se torna mais precisa, focando em valores que estão dentro do comportamento esperado.

```
mean_by_status = df1.groupby('status')['f0_'].mean()
std_by_status = df1.groupby('status')['f0_'].std()

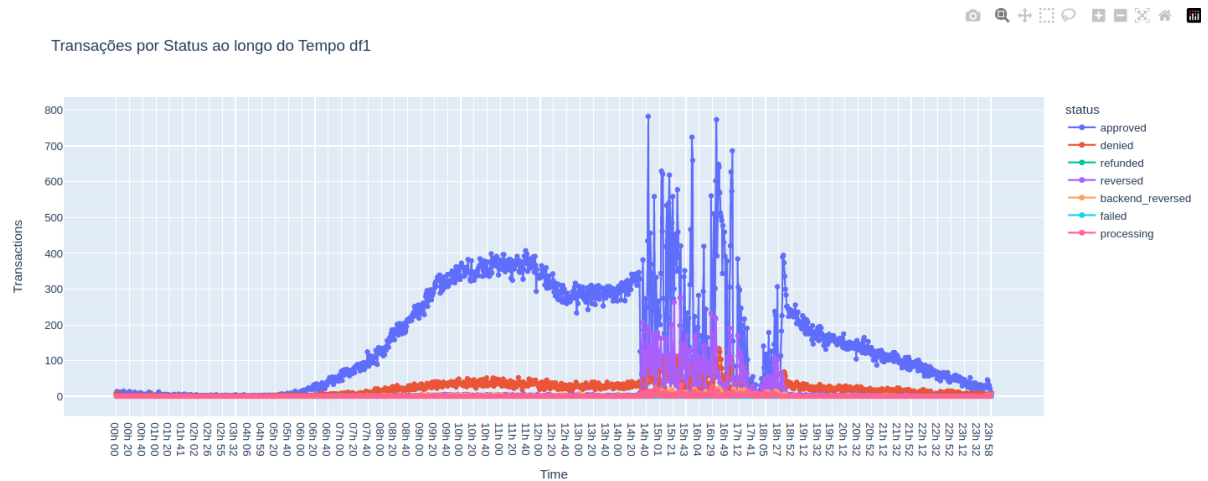
avg_without_outliers_df1 = df1.groupby('status').apply(
    lambda x: x[(x['f0_'] >= (mean_by_status[x.name] - 2 * std_by_status[x.name])) &
                (x['f0_'] <= (mean_by_status[x.name] + 2 * std_by_status[x.name]))]['f0_'].mean()
).round(2)
print('Excluindo outliers baseado no desvio padrão', avg_without_outliers_df1)
```

2. Tipo de gráfico:

- Como se trata de um gráfico com muitos pontos e diversos status, foi escolhido utilizar um gráfico mais dinâmico, utilizando então o plotly express por ter boa visualização, facilidade e praticidade na plotagem dos dados.
- Foi deixado um gráfico mais comum, plotado pelo matplotlib mesmo, para comparação.

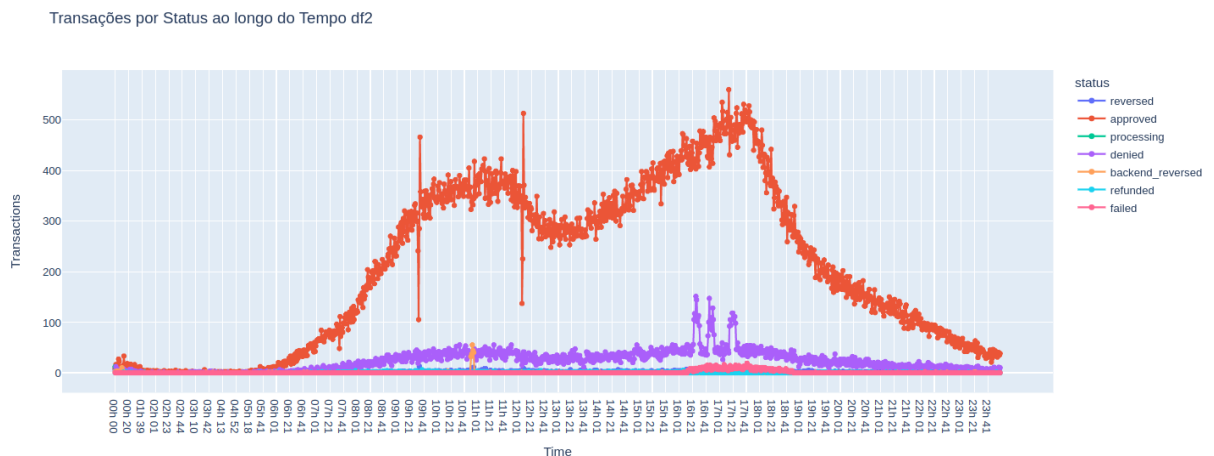
3. Incidente em transactions_1:

- **Pico de transações revertidas e negadas:** Entre **14h30 e 18h45**, houve um pico acentuado nas transações com status de reversed e denied. Isso indica um possível problema no sistema, pois esse comportamento é incomum.
- **Queda em transações aprovadas:** Durante o mesmo período, houve uma queda significativa nas transações aprovadas (approved), corroborando a existência de um incidente que afetou o fluxo normal de aprovações.
- Outros status, como refunded, failed, e backend_reversed, também apresentaram picos, indicando possíveis falhas no processamento das transações e longas esperas para concluir operações (processing).



4. Incidente em transactions_2:

- Em transactions_2, observam-se picos de transações negadas (denied) e falhas (failed) entre **16h23 e 17h25**. Esse intervalo foi identificado como crítico, exigindo alertas do sistema para notificar sobre o aumento inesperado dessas transações.
- Além disso, entre **10h51 e 10h54**, houve um pico de transações backend_reversed, outro sinal de um possível problema no sistema que deve ser investigado.



Sistema de Alerta

- O sistema de alerta foi implementado usando Python, SQLite, Flask e Ngrok para criar uma API que monitora as transações em tempo real.
- Critérios de Alerta:**
 - Quando o volume de transações de um determinado status ultrapassa o **dobro da média calculada** (excluindo outliers), o sistema envia uma mensagem de alerta. A mensagem inclui o status, o horário e a quantidade de transações que dispararam o alerta.

```
# Function about check if the set of transactions requires an alert
def check_anomaly(transaction_time, transaction_status, transaction_count, mean_values):
    if transaction_status == 'approved':
        return None
    if transaction_status in mean_values and (transaction_status == 'denied' or transaction_status == 'reversed' or transaction_status == 'failed' or transaction_status == 'suspicious'):
        mean_value = mean_values[transaction_status]
        # The rule for the alert is to be the twice of average
        if transaction_count > mean_value * 2:
            return f"{transaction_time} High number of {transaction_status} transactions ({transaction_count})"
    return None
```

- Isso garante que o sistema detecte anomalias e possa reagir rapidamente a qualquer comportamento inesperado.

Considerações Finais e Sugestões

- **Melhorias Potenciais:**
 - A sugestão de usar um modelo de aprendizado de máquina (LLM) para identificar padrões de anomalia poderia aprimorar a detecção de eventos fora do normal, tornando o sistema mais robusto.
 - O uso de bancos de dados mais robustos (como Prometheus ou InfluxDB) poderia ser explorado caso haja necessidade de auditoria ou consulta de grandes volumes de dados históricos, ou se o sistema crescer em complexidade.
 - Ações poderiam ser tomadas de forma automatizada com uma quantidade x de alertas em determinado tempo, como um alerta no slack ou abertura automática de incidente.