

Análise Conformal de Previsões vs. Valores Reais (Detecção de Deriva Temporal)

Neste notebook, vamos avaliar as previsões do modelo em comparação aos valores reais das variáveis de numerário de uma agência bancária, utilizando o **método conformal** para detectar possíveis **derivas temporais** (mudanças de comportamento, como as causadas pela pandemia). Faremos tanto análises **univariadas** (cada variável separadamente) quanto **multivariadas** (considerando todas as variáveis conjuntamente). As etapas gerais são:

1. **Preparação dos Dados:** Carregar os arquivos de previsões do modelo e de valores reais, mesclar pelas colunas de `AGENCIA` e `DATA`, e filtrar os dados para a agência de interesse. Em seguida, ordenar por data e separar um período de **calibração (pré-pandemia)** e outro de **teste (pós-pandemia)** para análise de deriva.
2. **Análise Univariada:** Para cada variável target (SAQ, DEP, SAQCEI, DEPCEI), calcular os **erros de previsão** e aplicar o método conformal para obter intervalos de predição com nível de confiança determinado (ex.: 90%). Avaliar a *cobertura* desses intervalos (percentual de pontos em que o valor real caiu dentro do intervalo previsto) no período de calibração e no período de teste, comparando com o esperado. Calcular métricas de erro como RMSE para cada variável em ambos os períodos.
3. **Visualização Univariada:** Plotar, para cada variável, os valores preditos vs. reais ao longo do tempo, juntamente com o **intervalo de predição conformal** calibrado no período pré-pandemia. Destacar pontos fora do intervalo no período pós-pandemia, evidenciando possíveis quebras de desempenho do modelo.
4. **Análise Multivariada:** Calcular um **escore de não-conformidade conjunto** (por exemplo, o *máximo* dos erros absolutos das 4 variáveis em cada dia) no período de calibração. Definir um limite conformal multivariado (quantil) e avaliar quantos dias no período de teste excedem esse limite (ou seja, pelo menos uma variável fora do intervalo esperado), indicando deriva conjunta.
5. **Visualização Multivariada:** Plotar o escore conjunto de erro ao longo do tempo com o limite conformal, destacando períodos em que o erro combinado excede o esperado.

Vamos iniciar carregando e preparando os dados conforme descrito.

Preparação dos Dados

- **Carregar dados de previsão do modelo:** arquivo CSV `previsoes_numerario_pre_pos_pandemia.csv` contendo as colunas:
 - `AGENCIA` – código da agência (identificador, texto)
 - `DATA` – data da previsão (tipo data)
 - `DEPOSITO` – volume de depósito (em milhares) previsto pelo modelo (corresponde à coluna real `DEP` no features)
 - `SAQUE` – volume de saque previsto (corresponde à coluna real `SAQ`)
 - `DEP_CEI` – volume de depósito em caixa eletrônico previsto (corresponde à coluna real `DEPCEI`)
 - `SAQUE_CEI` – volume de saque em caixa eletrônico previsto (corresponde à coluna real `SAQCEI`)

- **Carregar dados de valores reais:** arquivo Parquet `features_numerario.parquet` contendo pelo menos as colunas:
 - `AGENCIA`, `DATA` – identificadores de agência e data
 - `DEP`, `SAQ`, `DEPCEI`, `SAQCEI` – volumes reais correspondentes às variáveis acima.
- **Mesclar as tabelas** de previsões e reais usando as colunas `AGENCIA` e `DATA`, obtendo assim um único *DataFrame* com as previsões e valores reais lado a lado para cada variável.
- **Selecionar uma agência** para análise: como o dataset contém múltiplas agências, escolheremos apenas uma (identificada pelo código `AGENCIA`) para focar a análise. Iremos filtrar o *DataFrame* mesclado para essa agência.
- **Ordenar por data** para ter a série temporal em ordem cronológica.
- **Definir período de calibração vs. teste:** estabelecer uma data de corte que separe o período **pré-pandemia** (antes da data de corte, usado para calibrar o método conformal) do período **pós-pandemia** (após a data de corte, onde avaliaremos a possível deriva). Aqui vamos usar, por exemplo, **1º de março de 2020** como início aproximado dos impactos da pandemia. (Adapte a data conforme necessário.)

Vamos executar esses passos:

```
import pandas as pd

# Carregar dados de previsão e valores reais
df_pred = pd.read_csv('previsoes_numerario_pre_pos_pandemia.csv',
                      parse_dates=['DATA'])
df_real = pd.read_parquet('features_numerario.parquet',
                          columns=['AGENCIA', 'DATA', 'DEP', 'SAQ', 'DEPCEI', 'SAQCEI'])

# Mesclar previsões com valores reais (inner join em AGENCIA e DATA)
df_merged = pd.merge(df_pred, df_real, on=['AGENCIA', 'DATA'], how='inner')

# Listar agências disponíveis e selecionar uma para análise
agencias = sorted(df_merged['AGENCIA'].unique())
print(f"Agências disponíveis: {agencias}")
# Definir manualmente ou via input a agência desejada:
selected_agencia = input("Digite o código da agência a ser analisada: ")
selected_agencia = selected_agencia.strip()
# Se não for fornecida entrada, podemos usar a primeira como padrão (opcional)
if selected_agencia == "" and len(agencias) > 0:
    selected_agencia = agencias[0]
    print(f"Nenhuma agência informada. Usando padrão: {selected_agencia}")

# Filtrar os dados para a agência selecionada e ordenar por data
data_agencia = df_merged[df_merged['AGENCIA'] ==
                          selected_agencia].sort_values('DATA').reset_index(drop=True)

# Definir período de calibração (pré-pandemia) e teste (pós-pandemia)
corte_data = pd.Timestamp('2020-03-01') # data aproximada do início da
pandemia
cal_data = data_agencia[data_agencia['DATA'] < corte_data].copy()
test_data = data_agencia[data_agencia['DATA'] >= corte_data].copy()
```

```
print(f"Agência selecionada: {selected_agencia}")
print(f"Período de calibração: {cal_data['DATA'].min().date()} até {cal_data['DATA'].max().date()} (total de {len(cal_data)} dias)")
print(f"Período de teste: {test_data['DATA'].min().date()} até {test_data['DATA'].max().date()} (total de {len(test_data)} dias)")
```

Acima, exibimos a lista de agências disponíveis e permitimos selecionar a agência a ser analisada. Em seguida, dividimos os dados dessa agência em dois períodos (antes e depois de `corte_data`). Esses conjuntos serão usados para calibrar e testar o método conformal.

Análise Univariada (por Variável)

Nesta seção, analisamos cada variável de forma independente:

- Calculamos os **erros de previsão** para cada variável target (diferença entre valor real e previsto) no conjunto de calibração.
- Aplicamos o **método conformal** para determinar um **intervalo de predição** para cada variável: aqui utilizaremos um intervalo bilateral fixo determinado pelo quantil dos **erros absolutos** no período de calibração. Por exemplo, para um nível de confiança de 90% ($\alpha = 0,10$), encontramos o quantil de 90% dos erros absolutos de calibração. Esse valor `q` servirá de **limite conformal**: espera-se que aproximadamente 90% dos futuros erros absolutos estejam **abaixo ou iguais a** `q` se não houver mudança no comportamento do erro.
- Avaliamos a **cobertura** no período de calibração (deve ser próximo de 90%, dado como calibramos) e no período de teste (pós-pandemia). Se ocorrer **deriva** na série temporal, esperamos que a cobertura no período de teste caia significativamente abaixo do esperado (muitos pontos reais ficaram fora do intervalo previsto).
- Calculamos métricas de erro como **RMSE** (Root Mean Square Error) para cada variável nos períodos de calibração e teste, para quantificar o aumento do erro após a mudança.

Vamos definir o nível de significância α e calcular os intervalos conformais univariados e métricas para cada variável:

```
import numpy as np

alpha = 0.10 # nível de significância para o intervalo conformal (ex.: 10% -
> intervalo de 90%)

# Lista das variáveis de previsão vs. variáveis reais correspondentes
vars_pred_real = [
    ("DEPOSITO", "DEP"),      # Depósito (previsto vs real)
    ("SAQUE", "SAQ"),         # Saque (previsto vs real)
    ("DEP_CEI", "DEPCEI"),    # Depósito em CAIXA ELETRÔNICO (previsto vs
real)
    ("SAQUE_CEI", "SAQCEI")   # Saque em CAIXA ELETRÔNICO (previsto vs real)
]

metrics = [] # para armazenar métricas de cada variável

for var_pred, var_real in vars_pred_real:
    # Erros absolutos no período de calibração
```

```

cal_errors = (cal_data[var_real] - cal_data[var_pred]).abs()
# Limite conformal (quantil 1-alpha dos erros absolutos de calibração)
q = cal_errors.quantile(1 - alpha)
# Cobertura na calibração: proporção de erros dentro do limite q
if len(cal_errors) > 0:
    coverage_cal = (cal_errors <= q).mean()
else:
    coverage_cal = np.nan

# Erros absolutos no período de teste
test_errors = (test_data[var_real] - test_data[var_pred]).abs()
# Cobertura no teste: proporção de erros de teste dentro do mesmo limite
q
if len(test_errors) > 0:
    coverage_test = (test_errors <= q).mean()
else:
    coverage_test = np.nan

# Cálculo de RMSE nos períodos de calibração e teste
rmse_cal = np.sqrt(((cal_data[var_real] - cal_data[var_pred])**2).mean())
rmse_test = np.sqrt(((test_data[var_real] -
test_data[var_pred])**2).mean())

# Armazenar métricas
metrics.append({
    "Variável": var_real,
    f"Limite({int((1-alpha)*100)}%)": float(q) if q == q else np.nan, #
convertendo para float simples se não for NaN
    "Cobertura_Calibração": coverage_cal,
    "Cobertura_Teste": coverage_test,
    "RMSE_Calibração": rmse_cal,
    "RMSE_Teste": rmse_test
})

# Exibir tabela de métricas por variável
metrics_df = pd.DataFrame(metrics).set_index("Variável")
metrics_df.style.format({
    f"Limite({int((1-alpha)*100)}%)": "{:.3f}",
    "Cobertura_Calibração": "{:.2%}",
    "Cobertura_Teste": "{:.2%}",
    "RMSE_Calibração": "{:.3f}",
    "RMSE_Teste": "{:.3f}"
})

```

Após executar o código acima, teremos uma tabela (`metrics_df`) com as seguintes colunas para cada variável target:

- **Limite(90%):** o valor do limite conformal `q` (diferença absoluta) correspondente a 90% de cobertura nos dados de calibração.
- **Cobertura_Calibração:** porcentagem de dias no período de calibração em que o erro ficou dentro do limite `q` (deveria ser próximo de 90%).

- **Cobertura_Teste:** porcentagem de dias no período de teste com erro dentro do limite q . Valores significativamente menores que 90% indicam quebra de suposição (deriva) – ou seja, muitos erros no pós-pandemia excedem o limite estimado na fase pré, mostrando que o modelo perdeu acurácia/confiança.
- **RMSE_Calibração e RMSE_Teste:** erro quadrático médio nos dois períodos, ilustrando o aumento de erro médio após a mudança.

Você deve observar que para cada variável, a cobertura no período pós-pandemia caiu, e o RMSE aumentou, evidenciando a deterioração do desempenho do modelo após a pandemia (caso de forte deriva).

Visualização dos Intervalos Conformais Univariados

Para entender visualmente a magnitude dos erros e onde o modelo passou a falhar, vamos plotar cada variável mostrando:

- A série **real** versus **prevista** ao longo do tempo.
- O **intervalo de predição** conformal de 90% (sombras cinza) em torno do valor previsto, calculado a partir do período de calibração (valor constante $\pm q$ para cada variável, aplicado em torno da previsão).
- Uma linha vertical indicativa do ponto de corte (início da pandemia).
- Destaque em vermelho (x) para os pontos **fora do intervalo** no período de teste, que representam casos onde o valor real ficou além do esperado pelo modelo/calibração.

Esses gráficos permitem ver após qual momento e com que frequência as previsões passaram a falhar em cada variável.

```
import matplotlib.pyplot as plt

# Configurar tamanho de figura padrão
plt.rcParams['figure.figsize'] = (8, 4)

# Iterar por cada variável para plotar separadamente
for var_pred, var_real in vars_pred_real:
    # Obter o limite q previamente calculado (do DataFrame de métricas)
    q = metrics_df.loc[var_real, f"Limite({int((1-alpha)*100)}%)"]
    # Plotar valores previsto vs real
    plt.figure()
    plt.plot(data_agencia['DATA'], data_agencia[var_pred], label=f"Previsto - {var_real}")
    plt.plot(data_agencia['DATA'], data_agencia[var_real], label=f"Real - {var_real}")
    # Preencher intervalo de predição [previsto - q, previsto + q]
    plt.fill_between(data_agencia['DATA'],
                     data_agencia[var_pred] - q,
                     data_agencia[var_pred] + q,
                     color='gray', alpha=0.3, label=f"Intervalo {int((1-alpha)*100)}%")
    # Linha vertical no início da pandemia
    plt.axvline(corte_data, color='red', linestyle='--', label='Início')
```

```
Pandemia')
    # Destacar pontos fora do intervalo no período de teste
    out_of_bound = test_data[((test_data[var_real] -
test_data[var_pred]).abs() > q)]
    plt.scatter(out_of_bound['DATA'], out_of_bound[var_real], color='red',
marker='x', label='Fora do Intervalo')
    # Títulos e legendas
    plt.title(f"{var_real}: Valor Real vs Previsto com Intervalo Conformal")
    plt.xlabel("Data")
    plt.ylabel(f"Valor {var_real} (milhares)")
    plt.legend(loc='best')
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()
```

Interpretação dos gráficos univariados: Cada gráfico mostra se e quando os valores reais (linha contínua) passam a ficar frequentemente **fora da faixa cinza** (intervalo previsto). Antes do ponto de corte (pré-pandemia), por construção ~90% dos pontos reais deveriam estar dentro da faixa prevista. Após a pandemia, se houve deriva, veremos vários marcadores x vermelhos indicando reais fora do intervalo – sinal de que o modelo **não conseguiu prever corretamente** essas magnitudes dentro da incerteza esperada. Cada variável pode apresentar níveis de deriva distintos (por exemplo, talvez saques em caixa eletrônico tenham mudado mais drasticamente que depósitos em balcão, etc.), mas todas com perda de performance.

Análise Multivariada (Conjunto de Variáveis)

Na análise multivariada, consideramos o **vetor de previsões** vs **vetor de valores reais** de **todas as 4 variáveis simultaneamente** em cada dia. O método conformal nos permite avaliar se uma nova observação multivariada está ou não consistente com a distribuição de referências (calibração). Aqui faremos o seguinte:

- Para cada dia, calculamos um **escore de não-conformidade conjunto**. Uma escolha simples é usar o **máximo dos erros absolutos** das 4 variáveis naquele dia. Assim, um dia será considerado dentro do intervalo previsto conjunto se *todas os erros* estiverem dentro dos limites univariados, o que equivale a dizer que o máximo dos erros está abaixo do limite.
- Usando os dados de calibração, calculamos o quantil de 90% desses máximos de erro (ou seja, um valor de escore Q_multi tal que 90% dos dias de calibração têm erro máximo \leq Q_multi). Esse será o **limite conformal multivariado** para 90% de confiança.
- Verificamos a **cobertura conjunta**: a porcentagem de dias em que *todas as variáveis* ficaram dentro dos intervalos previstos (equivalentemente, erro máximo \leq Q_multi). No período de calibração deve ser ~90%. No período de teste, uma queda nesse percentual indica que em muitos dias pelo menos uma variável “falhou” (saiu fora do esperado), caracterizando deriva no comportamento **conjunto** do sistema.
- Essa análise conjunta é mais rigorosa: exige que o modelo acerte **simultaneamente** todas as variáveis em um dia para contar como “dentro do intervalo”.

Vamos calcular o escore conjunto e o limite conformal multivariado, e avaliar a cobertura:

```
# Calcular o erro máximo absoluto por dia no período de calibração e teste
if len(cal_data) > 0:
```

```

    cal_max_error = cal_data.apply(lambda row: max(abs(row['DEP'] -
row['DEPOSITO']),
abs(row['SAQ'] -
row['SAQUE'])),
abs(row['DEPCEI'] -
abs(row['SAQCEI'] -
row['SAQUE_CEI'])), axis=1)
else:
    cal_max_error = pd.Series(dtype=float)
if len(test_data) > 0:
    test_max_error = test_data.apply(lambda row: max(abs(row['DEP'] -
row['DEPOSITO']),
abs(row['SAQ'] -
row['SAQUE'])),
abs(row['DEPCEI'] -
abs(row['SAQCEI'] -
row['SAQUE_CEI'])), axis=1)
else:
    test_max_error = pd.Series(dtype=float)

# Limite conformal multivariado (quantil 1-alpha dos máximos de erro de
calibração)
Q_multi = cal_max_error.quantile(1 - alpha)

# Cobertura conjunta nos períodos
coverage_multi_cal = None
coverage_multi_test = None
if len(cal_max_error) > 0:
    coverage_multi_cal = (cal_max_error <= Q_multi).mean()
if len(test_max_error) > 0:
    coverage_multi_test = (test_max_error <= Q_multi).mean()

print(f"Limite conformal conjunto (90%): {Q_multi:.3f}")
print(f"Cobertura conjunta na calibração: {coverage_multi_cal:.2%} (esperado
~{(1-alpha):.0%})")
print(f"Cobertura conjunta no teste: {coverage_multi_test:.2%}")

```

Os resultados impressos mostram o valor do limite `Q_multi` e as coberturas observadas. Em caso de deriva severa, a **cobertura conjunta no teste** cairá muito abaixo de 90%. Por exemplo, pode mostrar algo como 30% ou 0% (caso extremo onde quase todos os dias no pós-pandemia tiveram pelo menos uma variável fora do intervalo). Isso indica que praticamente todos os dias do período pós-pandemia apresentaram um padrão de valores que o modelo (calibrado no pré-pandemia) não conseguiu prever simultaneamente dentro do intervalo de confiança – evidência clara de **quebra do regime anterior**.

Visualização do Desvio Conjunto

Para visualizar a deriva conjunta, vamos plotar o **máximo dos erros absolutos diários** ao longo do tempo, comparando com o **limite conformal conjunto** calculado:

- O eixo y será o *máximo erro absoluto diário* entre as 4 variáveis.
- Traçamos uma linha horizontal indicando o valor `Q_multi` (limite de 90% da calibração).
- Marcamos a linha vertical do início da pandemia.
- Pontos acima da linha de limite no pós-pandemia indicam dias onde o modelo falhou em pelo menos uma variável além do esperado.

```
plt.figure(figsize=(8,4))
plt.plot(cal_data['DATA'], cal_max_error, label='Máx Erro Abs (Calibração)',
color='blue')
plt.plot(test_data['DATA'], test_max_error, label='Máx Erro Abs (Teste)',
color='orange')
plt.axhline(Q_multi, color='green', linestyle='--', label=f'Limite 90%
(Q_multi={Q_multi:.2f})')
plt.axvline(corte_data, color='red', linestyle='--', label='Início Pandemia')
plt.title('Erro Máximo Absoluto Diário vs. Limite Conformal Conjunto')
plt.xlabel('Data')
plt.ylabel('Máx |Erro| das 4 variáveis')
plt.legend(loc='best')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

Nos resultados deste gráfico, vemos claramente o comportamento do erro máximo diário:

- **Antes da pandemia**, quase todos os pontos azuis (calibração) devem ficar abaixo da linha verde `Limite 90%` por construção.
- **Após a pandemia**, se há deriva, muitos pontos laranja ultrapassam a linha verde, indicando que nesses dias o erro de pelo menos uma variável excedeu o que era considerado “normal” no passado.

Em um cenário de forte mudança (por exemplo, mudança drástica nos padrões de saques e depósitos devido a lockdowns), é possível que quase todos os dias pós-pandemia estejam acima do limite – como indicado pela cobertura conjunta muito baixa.

Conclusões da Análise

Aplicando o método conformal aos dados de previsões vs. reais da agência selecionada, observamos evidências claras de **deriva de série temporal** causada pela pandemia:

- As **análises univariadas** mostraram que cada variável apresentou queda na cobertura dos intervalos de predição no período pós-pandemia (menor que os 90% esperados) e aumentos significativos nos erros (RMSE), indicando que o modelo subestimou ou superestimou consistentemente os novos valores.
- A **análise multivariada** reforçou esse resultado, mostrando que em grande parte dos dias pós-pandemia o conjunto de valores (SAQ, DEP, SAQCEI, DEPCEI) **não estava de acordo** com o

padrão anterior – ou seja, quase todos os dias houve pelo menos uma variável com erro fora do intervalo previsto, sinalizando uma quebra no comportamento conjunto dos dados.

Em resumo, o método conformal permitiu quantificar e visualizar que o **modelo treinado no período pré-pandemia perdeu validade no período pós-pandemia**, necessitando possivelmente de recalibração ou re-treinamento com dados recentes para voltar a fornecer previsões confiáveis. Os gráficos e métricas acima facilitam identificar **quando e em quais variáveis** ocorreram as maiores discrepâncias, auxiliando na tomada de decisão para manutenção do modelo.
