



Sinalizador Inteligente de Desvios Cadastrais: Reclamação que Corrige Cadastro

Tema: Operação

Autores: Marlon Henrique Cardoso

Co-Autores: Vinicius Franco Taurisano, Natan Marostica Lunardelli e Alexandre Alves De Carvalho

Empresa: EDP São Paulo Distribuição de Energia S.A

Resumo

As inconsistências cadastrais são um problema constante enfrentado por todas as empresas de distribuição de energia elétrica, incluindo a EDP SP. Os desafios surgem devido à complexidade da atualização dos dados cadastrais frente ao dinamismo das alterações realizadas nas redes de distribuição, onde a precisão dos dados cadastrais é crucial para uma rotina de operação eficiente e segura do sistema.

Na rede de distribuição, cada ponto de entrega de energia é vinculado a um número de instalação, que serve como a identidade do cliente. No entanto, divergências entre o cadastro no sistema e a realidade em campo são comuns. Por exemplo, um cliente pode estar cadastrado como sendo atendido por um transformador específico, mas, fisicamente (em campo), estar conectado a rede de outro transformador. Essas inconsistências podem gerar uma série de problemas operacionais, segurança e de atendimento.

Para mitigação e tratativa do problema, o COI da EDP SP desenvolveu uma ferramenta utilizando um modelo em Python que mapeia as instalações com possíveis desvios de cadastro. Esse modelo utiliza dados de reclamações de falta de energia e interrupções no fornecimento para identificar potenciais erros de cadastro. A eficiência do modelo foi validada em campo, demonstrando uma precisão de 90% na identificação correta de instalações com desvios de cadastro, bem como realiza o confronto da informação georeferenciada dos consumidores.

A implementação desse modelo é fundamental para garantir a operação eficiente da rede de distribuição. Ele permite a correção proativa de cadastros incorretos, melhorando a segurança, a apuração de indicadores, e reduzindo deslocamentos improdutivos. Além disso, assegura o pagamento correto de compensações aos clientes, contribuindo para a satisfação e a confiança dos consumidores.

Em resumo, os desvios de cadastro representam um desafio constante para todas as empresas de distribuição de energia elétrica. A EDP SP, ao adotar soluções tecnológicas avançadas, como o modelo em Python, está continuamente evoluindo na resolução desse problema. Essa solução desenvolvida garante a precisão dos cadastros e a qualidade do serviço prestado aos seus clientes, reforçando o compromisso da EDP SP com a excelência e a inovação no setor de distribuição de energia, em busca de eficiência na operação e satisfação dos clientes.

1. Introdução

Na rede de distribuição de uma concessionária, existem diversos tipos de conexões, incluindo a ligação do ramal do cliente. Assim como toda a topologia de rede é cadastrada no sistema de informação geográfica da empresa (mapa), o ponto de entrega de energia do cliente também é cadastrado e vinculado a um número de instalação. A instalação é a identidade do cliente e é utilizada nos canais de atendimento para solicitar serviços ou consultas, como reclamações de falta de energia.

Ao abrir uma reclamação de falta de energia, é gerado um protocolo vinculado ao atendimento do serviço solicitado, e uma ordem de atendimento é aberta para despacho da equipe até o local da instalação do cliente. A ordem gerada é plotada no mapa do Poweron de acordo com as coordenadas geográficas cadastradas para a instalação. Toda instalação está atrelada a um equipamento de proteção, como um transformador, que pode ser a fonte de uma ou várias instalações.

Identificamos uma oportunidade relacionada ao cadastro dessas instalações em relação ao equipamento de proteção no seguinte cenário: o cliente da instalação XYZ está cadastrado no transformador "A" no sistema, mas em campo, o ramal de ligação do cliente está sendo energizado pelo transformador "B".

Ilustração do Cenário

Abaixo, apresentamos uma ilustração do cenário estudado, que mostra a discrepância entre o cadastro no sistema e a realidade em campo (Atentar que o poste é final de linha entre dois secundários).

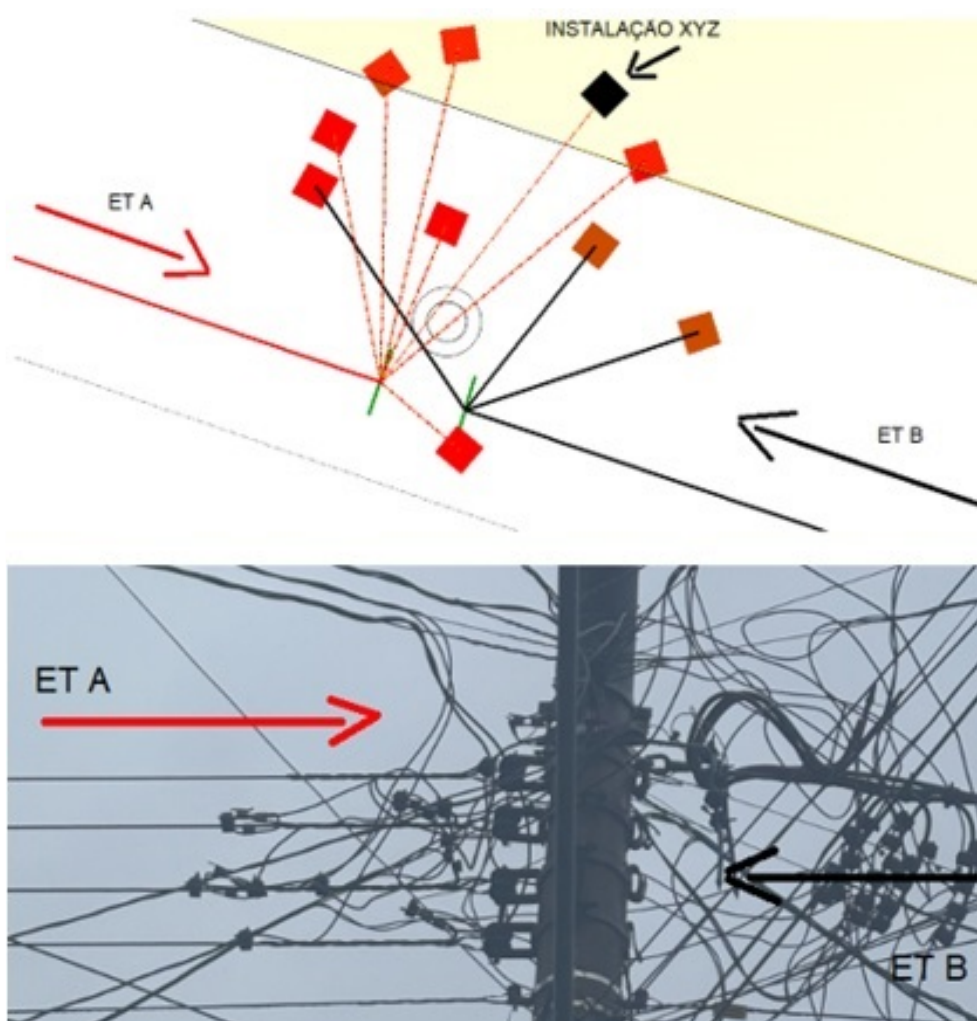


Figura 1: Diagrama e foto do cenário de estudo. A instalação XYZ está cadastrada no transformador "A" (ET A) no sistema, mas em campo, está sendo energizada pelo transformador "B" (ET B).

Impactos do Cadastro Incorreto

Os impactos de um cadastro incorreto vão além do simples atendimento de uma reclamação de falta de energia:

- **Segurança:** Divergência de informações de equipamentos elétricos e coordenadas geográficas.
- **Apuração Incorreta de Indicadores:** Registra interrupção quando não há ou não registra interrupção quando há.
- **Pagamento Incorreto ou Não Pagamento de Compensação:** Multas por extrapolar limites de indicadores.
- **Deslocamento Improdutivo:** O sistema não agrupa a reclamação dentro da ordem do equipamento de proteção, gerando mais de um deslocamento para um mesmo evento.
- **Impacto no Processo de Indenização ao Cliente:** Quando o cliente abre um pedido de indenização por queima de aparelho após perturbação no fornecimento de energia elétrica.

A grande oportunidade identificada foi entender que, quando um cliente está com erro de cadastro e solicita atendimento através de uma reclamação de falta de energia, essa reclamação pode indicar um erro de cadastro para a instalação. Diante do impacto diário desse cenário, criamos um modelo para sinalizar instalações com potenciais desvios de cadastro.

2. Desenvolvimento

Elaboramos um modelo em Python para mapear as instalações com possíveis desvios de cadastro e criamos um aplicativo para registrar os dados coletados em campo, sinalizando essas informações para a área responsável pela correção.

A primeira etapa do código classifica as instalações em dois tipos: **INTERRUPÇÃO** e **RECLAMAÇÃO**, dentro de uma consulta SQL.

- **Tipo INTERRUPÇÃO:** Instalações que registraram interrupções no fornecimento de energia elétrica com duração superior a 3 minutos.
- **Tipo RECLAMAÇÃO:** Ordens canceladas ou classificadas como deslocamento improdutivo, incluindo verificações, defeitos internos, causas não classificadas, prédio fechado, endereço não localizado e causas não determinadas. Contempla reclamações relacionadas a Energia Fraca, Falta de Energia, Luz oscilando ou intermitente e Pique de Energia.

O código cria um DataFrame com os resultados da consulta SQL, separando os dados em dois DataFrames distintos: um para reclamações e outro para interrupções. Iteramos sobre cada cidade presente no DataFrame, processando os dados cidade por cidade. Para cada cidade, filtramos as reclamações e interrupções correspondentes, comparando cada reclamação com cada interrupção para verificar se atendem aos critérios definidos.

Calculamos a distância entre as coordenadas da reclamação e da interrupção utilizando a fórmula de Haversine, determinando a proximidade geográfica entre os eventos. Se a distância for menor que 100 metros, adicionamos as informações à lista `reclamacoes_proximas`.

Definimos os nomes das colunas para o DataFrame do dia, organizando os dados de forma clara e estruturada. Finalmente, criamos o DataFrame do dia utilizando a lista `reclamacoes_proximas`, consolidando os dados em um formato tabular.

	INT	ORD_INT	INICIO_INT	FIM_INT	LAT_INT	LONG_INT	PARENT_INT	REC	ORD_REC	INICIO_REC	FIM_REC	LAT_REC	LONG_REC	PARENTE_REC
1	0000061786	7072430-1	2024-10-11 20:19:00	2024-10-11 21:42:00	-23.524685	-46.200747	130CF00070138	0084402024	7072487-1	2024-10-11 20:40:00	2024-10-13 10:15:00	-23.524373	-46.201299	130ET00008943
2	0000106054	7073846-1	2024-10-12 13:26:00	2024-10-12 15:45:00	-23.799142	-45.401036	185BF00000033	0150974906	7073980-1	2024-10-12 15:13:00	2024-10-12 15:29:00	-23.798925	-45.401240	185ET00539943
3	0000124303	7072097-1	2024-10-11 20:05:00	2024-10-11 21:02:00	-23.480643	-46.362701	165CF00070729	0037864149	7072213-1	2024-10-11 20:17:00	2024-10-12 10:18:00	-23.480888	-46.362595	165ET00002308
6	0000606002	7094364-1	2024-10-29 13:14:00	2024-10-29 15:18:00	-23.445003	-46.413316	045ET00040798	0150434076	7094459-1	2024-10-29 14:29:00	2024-10-29 14:36:00	-23.445092	-46.413566	045ET00587272
7	0000601000	7061893-1	2024-10-03 01:07:00	2024-10-03 04:54:00	-23.470165	-46.528563	045ET00521290	0000021466	7061921-1	2024-10-03 01:57:00	2024-10-03 09:01:00	-23.470051	-46.529354	045CZ00542590
...
21437	0151282499	7072044-1	2024-10-11 18:15:00	2024-10-12 20:19:00	-23.484914	-46.123692	130ET00502109	0151358415	7074242-1	2024-10-12 18:48:00	2024-10-14 09:43:00	-23.484420	-46.123943	130ET00609729
21446	0151283730	7081276-1	2024-10-20 08:30:00	2024-10-20 09:20:00	-23.059341	-45.589781	295ET00604756	0151284600	7081473-1	2024-10-20 08:59:00	2024-10-20 09:16:00	-23.059750	-45.589000	295ET00604758
21738	0151313999	7063523-1	2024-10-04 08:24:00	2024-10-04 11:12:00	-23.552485	-46.257391	130ET00002558	0150787716	7063969-1	2024-10-04 11:11:00	2024-10-04 11:19:00	-23.551813	-46.257129	130ET00009030
21739	0151313999	7063523-1	2024-10-04 08:24:00	2024-10-04 11:12:00	-23.552485	-46.257391	130ET00002558	0150895799	7063779-1	2024-10-04 09:13:00	2024-10-04 09:27:00	-23.551822	-46.257128	130ET00009030
21740	0151313999	7063523-1	2024-10-04 08:24:00	2024-10-04 11:12:00	-23.552485	-46.257391	130ET00002558	0150895799	7063907-1	2024-10-04 10:25:00	2024-10-04 10:27:00	-23.551822	-46.257128	130ET00009030

Figura 4: Tabela final com as informações das instalações com potenciais desvios de cadastros. Esta é a etapa principal do código:

```
# Criar um DataFrame do pandas com os resultados
TBLordem = pd.DataFrame(rows, columns=names)

# Separar os dados em RECLAMACAO e INTERRUPCAO
reclamacoes = TBLordem[TBLordem['TIPO'] == 'RECLAMACAO']
interrupcoes = TBLordem[TBLordem['TIPO'] == 'INTERRUPCAO']

# Inicializar uma lista para armazenar os resultados do dia
reclamacoes_proximas = []

# Iterar sobre cada cidade
for cidade in TBLordem['FAC_CITY'].unique():
    reclamacoes_cidade = reclamacoes[reclamacoes['FAC_CITY'] == cidade]
    interrupcoes_cidade = interrupcoes[interrupcoes['FAC_CITY'] == cidade]

    # Iterar sobre cada reclamacao e interrupcao na mesma cidade
    for _, reclamacao in reclamacoes_cidade.iterrows():
        for _, interrupcao in interrupcoes_cidade.iterrows():
            # Verificar se o início da reclamacao está dentro do período da interrupcao
            if (reclamacao['CLL_CALL_TIME'] >= interrupcao['CLL_CALL_TIME'] and
                reclamacao['CLL_CALL_TIME'] <= interrupcao['CLL_SOLUTION_TIME']):
                # Calcular a distância entre as coordenadas
                coord_reclamacao = (reclamacao['FAC_LATITUDE'], reclamacao['FAC_LONGITUDE'])
                coord_interrupcao = (interrupcao['FAC_LATITUDE'], interrupcao['FAC_LONGITUDE'])
                distancia = haversine(coord_reclamacao, coord_interrupcao)

                # Verificar se a distância é menor que 100 metros
                if distancia < 100:
                    reclamacoes_proximas.append((
                        interrupcao['CLL_CUSTOMER_NUMBER'], interrupcao['ORD_ORDER'], interrupcao['CLL_CALL_TIME'], interrupcao['CLL_SOLUTION_TIME'],
                        interrupcao['FAC_LATITUDE'], interrupcao['FAC_LONGITUDE'], interrupcao['FAC_FACILITY_PARENT'],
                        reclamacao['CLL_CUSTOMER_NUMBER'], reclamacao['ORD_ORDER'], reclamacao['CLL_CALL_TIME'], reclamacao['CLL_SOLUTION_TIME'],
                        reclamacao['FAC_LATITUDE'], reclamacao['FAC_LONGITUDE'], reclamacao['FAC_FACILITY_PARENT']
                    ))

# Definindo os nomes das colunas
colunas = ['INT', 'ORD_INT', 'INICIO_INT', 'FIM_INT', 'LAT_INT', 'LONG_INT', 'PARENT_INT',
            'REC', 'ORD_REC', 'INICIO_REC', 'FIM_REC', 'LAT_REC', 'LONG_REC', 'PARENTE_REC']

# Criando o DataFrame do dia
df_dia = pd.DataFrame(reclamacoes_proximas, columns=colunas)
```

Figura 5: Etapa principal do código para identificar a reclamação próxima de uma interrupção.

Aplicativo para Registro de dados de Campo

Para registrar as validações de campo, criamos um aplicativo desenvolvido pelo Power Apps. O aplicativo possui uma interface amigável e intuitiva, facilitando a inserção e atualização dos dados pelos técnicos em campo. Com ele, os técnicos podem inserir fotos, latitude e longitude do local do cliente, tornando o processo de coleta de dados mais preciso e eficiente. Utilizando dispositivos móveis, os técnicos podem coletar os dados corretos diretamente no local do cliente, conforme identificado pela nossa análise, permitindo que os erros de cadastro sejam corrigidos posteriormente.

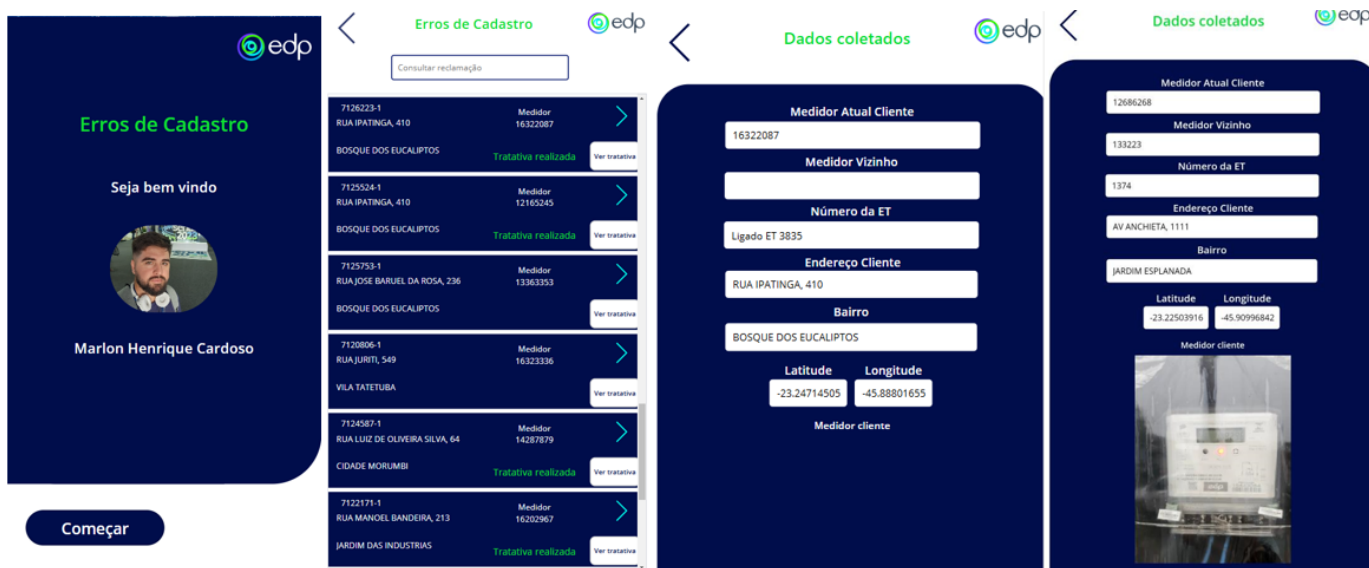


Figura 6: Imagens das telas do app para registro dos dados levantados em campo.

Resultados da análise dos dados

Ao longo do ano, nosso modelo inteligente sinalizou um total acumulado de aproximadamente 5.000 instalações com potenciais erros de cadastro. Observamos algumas tendências e proporções que destacam a eficiência do nosso modelo, especialmente em cenários de *maior demanda* e em *momentos de contingência*.

É importante ressaltar que 70% das reclamações foram canceladas graças à pro atividade do nosso time de call-back, demonstrando a eficácia de nossa abordagem preventiva. No entanto, os outros 30% resultaram em deslocamentos improdutivos, gerando um desperdício significativo. Considerando um custo de 300 reais por atendimento/deslocamento, esses deslocamentos improdutivos representam um desperdício de aproximadamente 450 mil reais. Este dado reforça a utilização constante do modelo.

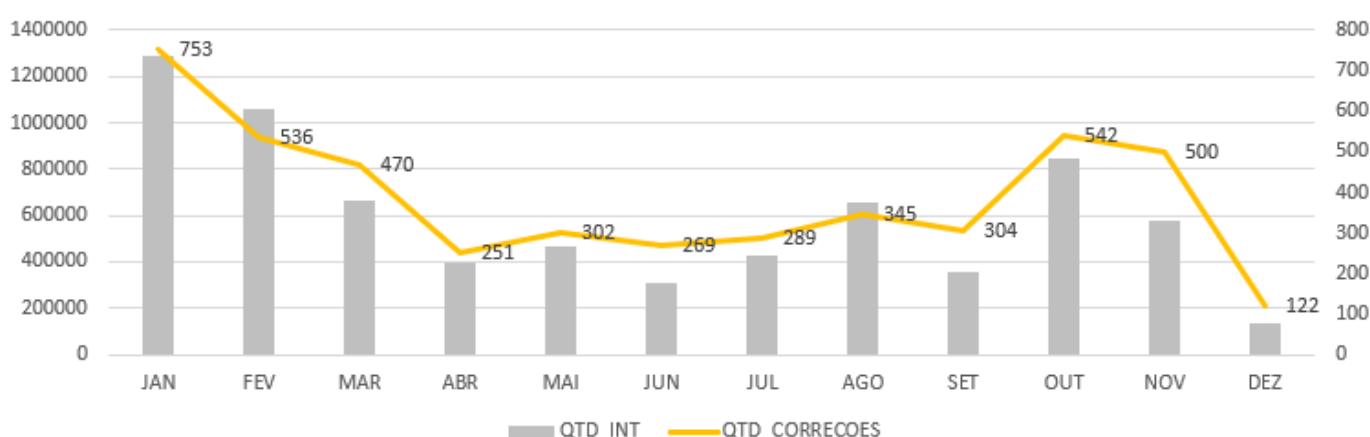


Figura 7: Evolução dos números de interrupções e correções ao longo de 2024.

Analisando especificamente o mês de janeiro de 2024 como exemplo, observamos a proporção de correções em relação ao número de reclamações registradas diariamente. A análise revelou que essa proporção tende a aumentar em dias atípicos, quando o número de interrupções é significativamente maior. Nos dias atípicos, a quantidade de interrupções registradas é elevada, resultando em um aumento no número de reclamações e, conseqüentemente, na proporção de correções.

Abaixo, apresentamos um gráfico que ilustra a evolução dos números de interrupções (QTD_INT) e correções (QTD_CORRECOES) ao longo do mês de janeiro de 2024.

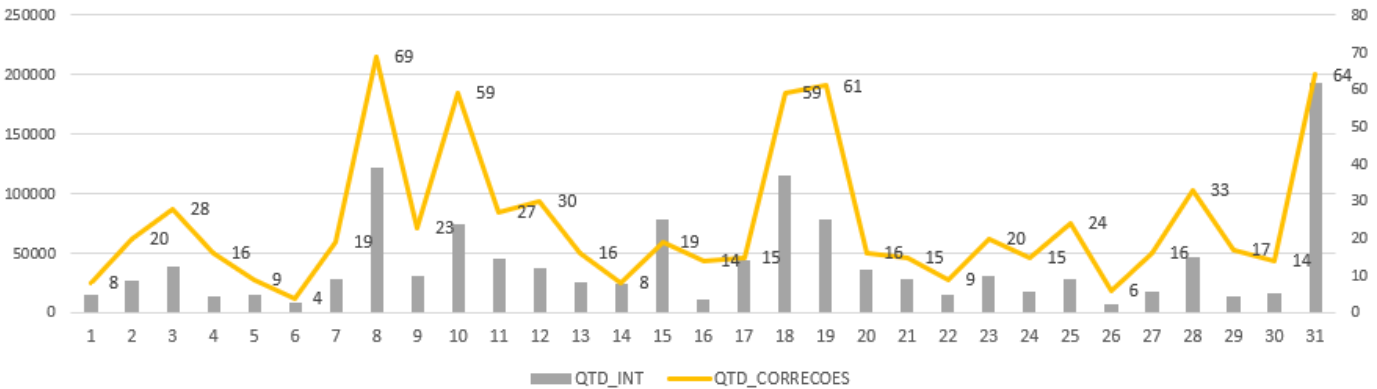


Figura 8: Evolução dos números de interrupções e correções ao longo do mês de janeiro de 2024. Em dias normais, o modelo identifica em média 15 reclamações diárias que podem ser corrigidas. Já em dias atípicos, esse número sobe para uma média de 50 reclamações diárias com potenciais de correção.

Exemplos de Pesquisa:

Pesquisa finalizada para o dia 29/11/2024	demorou 1.82 minutos,	analisando 383 reclamações e 17069 interrupções.	Resultando em 14 reclamações com potenciais de correção
Pesquisa finalizada para o dia 30/11/2024	demorou 1.25 minutos,	analisando 215 reclamações e 19556 interrupções.	Resultando em 12 reclamações com potenciais de correção
Pesquisa finalizada para o dia 01/12/2024	demorou 1.02 minutos,	analisando 175 reclamações e 10571 interrupções.	Resultando em 14 reclamações com potenciais de correção
Pesquisa finalizada para o dia 02/12/2024	demorou 2.90 minutos,	analisando 471 reclamações e 34553 interrupções.	Resultando em 19 reclamações com potenciais de correção
Pesquisa finalizada para o dia 03/12/2024	demorou 4.72 minutos,	analisando 570 reclamações e 48068 interrupções.	Resultando em 27 reclamações com potenciais de correção
Pesquisa finalizada para o dia 04/12/2024	demorou 1.33 minutos,	analisando 421 reclamações e 15628 interrupções.	Resultando em 16 reclamações com potenciais de correção
Pesquisa finalizada para o dia 05/12/2024	demorou 1.06 minutos,	analisando 368 reclamações e 8514 interrupções.	Resultando em 13 reclamações com potenciais de correção
Pesquisa finalizada para o dia 06/12/2024	demorou 2.21 minutos,	analisando 330 reclamações e 20348 interrupções.	Resultando em 30 reclamações com potenciais de correção
Pesquisa finalizada para o dia 07/12/2024	demorou 0.29 minutos,	analisando 33 reclamações e 1167 interrupções.	Resultando em 3 reclamações com potenciais de correção

Figura 9: Exemplo da resposta do modelo com a quantidade de insumos e resultados obtidos. 15/01/2024: O modelo demorou 5.55 minutos com o processamento, resultando na comparação entre 669 reclamações e 78.292 interrupções, identificando 19 reclamações com potenciais de correção. 16/01/2024: O modelo demorou 1.32 minutos com o processamento, resultando na comparação entre 460 reclamações e 10.929 interrupções, identificando 14 reclamações com potenciais de correção.

Resultado do levantamento de campo

Nosso time realizou uma avaliação de campo para medir a eficiência dos resultados obtidos pelo modelo desenvolvido para identificar instalações com potenciais desvios de cadastro. A amostra selecionada incluiu 30 instalações que haviam gerado reclamações de falta de energia, permitindo uma análise detalhada da precisão e eficácia do modelo em situações reais.

Durante a avaliação, nossa equipe visitou as instalações identificadas pelo modelo como tendo potenciais desvios de cadastro. O objetivo era verificar se as reclamações de falta de energia estavam realmente associadas a erros de cadastro, conforme indicado pelo modelo. A metodologia envolveu a comparação das coordenadas geográficas e dos equipamentos de proteção cadastrados no sistema com a realidade observada em campo.

Os resultados da avaliação foram extremamente positivos. O modelo demonstrou uma eficiência de 90% na identificação correta de instalações com desvios de cadastro. Isso significa que, em 90% dos casos analisados, as reclamações de falta de energia estavam de fato relacionadas a erros de cadastro, conforme previsto pelo modelo. Esse alto índice de acerto confirma a robustez e a precisão do modelo em identificar problemas de cadastro que impactam diretamente a qualidade do serviço prestado aos clientes.

A eficiência de 90% alcançada pelo modelo é um indicativo claro de sua capacidade de melhorar significativamente a operação da rede de distribuição. Com essa ferramenta, é possível reduzir deslocamentos improdutivos, melhorar a apuração de indicadores, garantir a segurança das operações e assegurar o pagamento correto de compensações aos clientes. Além disso, a correção proativa dos cadastros incorretos contribui para a satisfação dos clientes e a continuidade do serviço.

3. Conclusão

A análise diária das reclamações e interrupções permite identificar instalações com potenciais desvios de cadastro, o que é crucial para a eficiência operacional e a satisfação do cliente. A implementação do modelo em Python e o uso de um aplicativo para registro de dados em campo têm se mostrado eficazes na identificação e correção de erros de cadastro.

Os resultados demonstram que, em dias atípicos, a quantidade de reclamações com potenciais de correção aumenta significativamente, indicando que o modelo é especialmente útil em situações de maior demanda. A média de reclamações diárias com potenciais de correção varia de 15 em dias normais a 50 em dias atípicos.

A continuidade do uso desse modelo e a análise constante dos dados são essenciais para manter a precisão dos cadastros e garantir a qualidade do serviço prestado aos clientes. A correção de cadastros incorretos não só melhora a segurança e a apuração de indicadores, mas também reduz deslocamentos improdutivos e assegura o pagamento correto de compensações aos clientes.

Próximas etapas:

Treinamento de Equipes: Treinar as equipes de campo e de atendimento para reconhecer sinais de desvios de cadastro e agir rapidamente.

Aprimoramento do Modelo: Continuar aprimorando o modelo em Python para aumentar a precisão e eficiência na identificação de desvios de cadastro.

Integração de Sistemas: Garantir que os sistemas de informação geográfica e de atendimento ao cliente estejam integrados e atualizados com as correções de cadastro realizadas.

Com essas ações, a EDP SP pode continuar a melhorar a qualidade do serviço prestado, aumentar a satisfação dos clientes e otimizar suas operações.

4. Referências bibliográficas

Dados referentes a EDP SP