CPAD 2025.1 - Computação para Análise de Dados

Discente: Lucas Edson Silva de Araújo
E-mail: lucas.edson@ufrpe.br

Docente Orientador: Prof. Dr. Ermeson Andrade
E-mail: ermeson.andrade@ufrpe.br

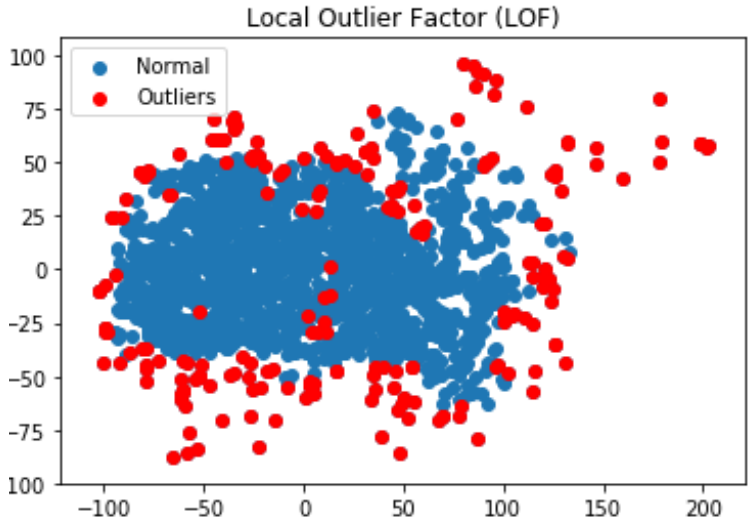
Detecção Inteligente de Anomalias em Ambientes de Borda - LOF (Local Outilier Factor)

Temática central:

- Projeto voltado à detecção de anomalias em motores industriais, com foco na aplicação de IA leve e eficiente para ambientes de borda, onde há limitações de poder computacional.

LOF (Local Outilier Factor):

- Algoritmo de detecção de anomalias que identifica pontos fora do padrão com base na densidade local. Comparada a densidade de um ponto com a de seus vizinhos — quanto mais isolado, maior o fator de anomalia.



Entendendo o LOF – Local Outlier Factor

📘 O que é o LOF?

O LOF é um algoritmo de detecção de anomalias baseado em densidade. Ele avalia o quão isolado um ponto está em relação aos seus vizinhos mais próximos.

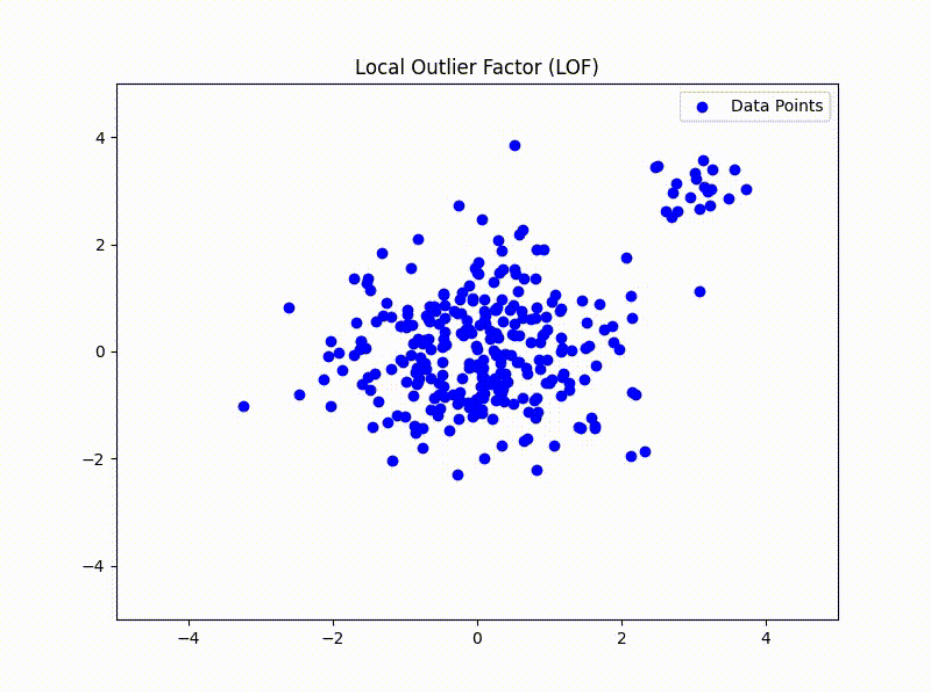
⚙️ Como funciona?

1. **Distância entre Vizinhos:**
   Calcula a distância entre um ponto e seus *k* vizinhos mais próximos.

2. **Densidade Local:**
   Mede a densidade de um ponto com base na média das distâncias para esses vizinhos.

3. **Fator de Anomalia (LOF Score):**
   Compara a densidade local do ponto com a dos vizinhos.

   - Se for similar → ponto normal.
   - Se for significativamente menor → ponto é uma anomalia.



Entendendo o LOF – Local Outlier Factor

📈 **Interpretação do LOF Score:**

- LOF ≈ 1 → Ponto normal
- LOF > 1 → Possível anomalia
- Quanto maior o LOF, maior a chance de ser um outlier

✅ **Vantagens:**

- Detecta anomalias locais, mesmo em regiões com diferentes densidades
- Não assume distribuição dos dados

⚠️ **Limitações:**

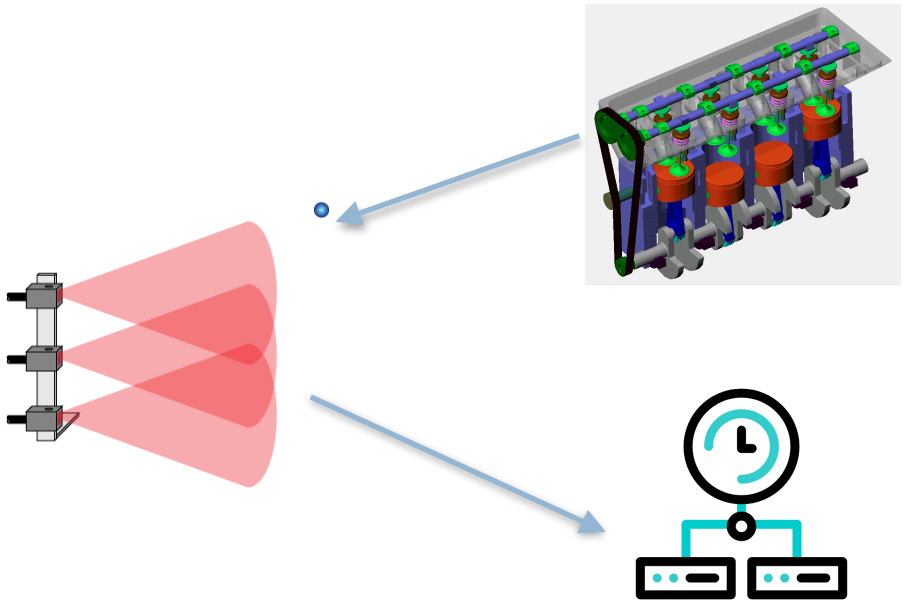
- Sensível à escolha de k
- Pode ter alto custo computacional em grandes volumes de dados

Estrutura dos Conjuntos de Dados

UFRPE
Universidade Federal Rural de Pernambuco

Composição Geral

- **Total de amostras**: 6.000
- **Coleta**: Realizada em **intervalos de 30 segundos**
- **Tipos de dados**:
  - **3 motores industriais**
  - **3750 amostras normais**
  - **2250 amostras anômalas**

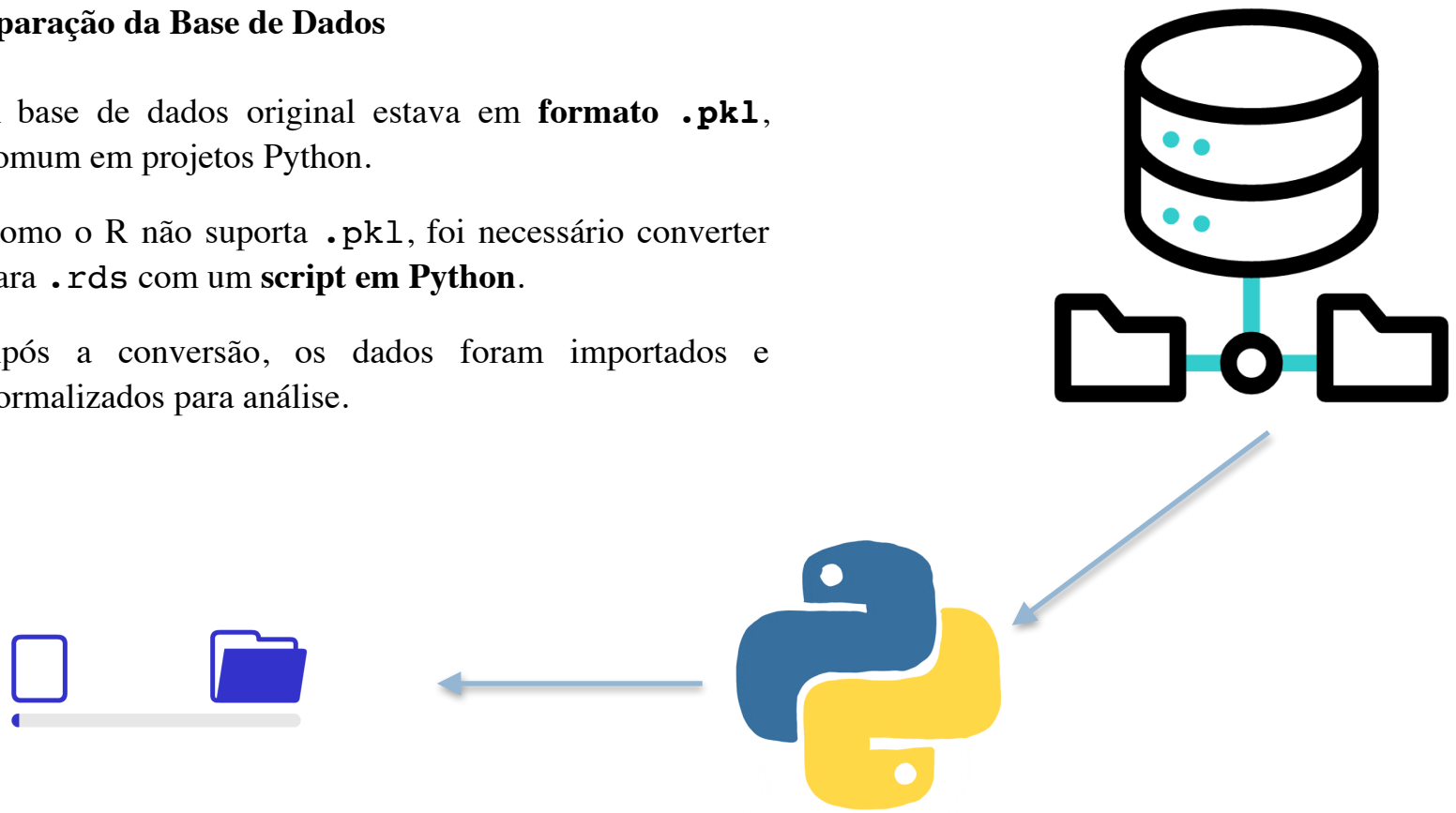


Subconjunto	Amostras Normais	Amostras Anômalas	Total de Amostras
Treinamento	1500 (500 por motor)	0	1500
Teste	0	6000	4000
Total	**1500**	**6000**	**7500**

Processo de implementação no R

🔁 **Preparação da Base de Dados**

- A base de dados original estava em **formato `.pkl`**, comum em projetos Python.
- Como o R não suporta `.pkl`, foi necessário converter para `.rds` com um **script em Python**.
- Após a conversão, os dados foram importados e normalizados para análise.



Processo de implementação no

🔧 **Etapas da Implementação**

1. Carregamento de Bibliotecas

As bibliotecas utilizadas nesta implementação são:

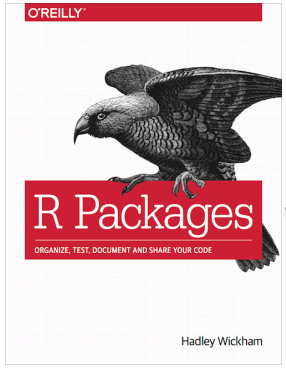
- `dbscan`: fornece a função `lof()` para cálculo do Local Outlier Factor.
- `ggplot2`: usada para visualizações (gráficos de densidade e histogramas).
- `caret`: empregada para métricas de avaliação como matriz de confusão e acurácia.
- `reticulate`: incluída para compatibilidade com possíveis integrações Python (não usada diretamente no código atual).

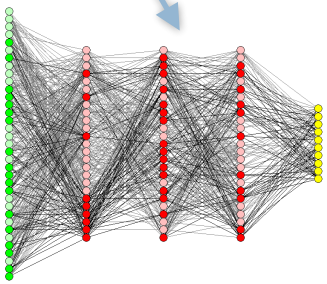
2. Pré-processamento dos Dados

- **Carregamento dos Dados**: arquivos `.rds` são lidos e classificados em normais (treino) e anômalos (teste) com base no nome dos arquivos.
- **Normalização**: todos os dados são convertidos para escala padrão (z-score), garantindo comparabilidade entre variáveis.
- **Remoção de Colunas Irrelevantes**: a coluna `date_time` é removida e valores `NA` são eliminados.
- **(Opcional)**: Caso os dados contenham PCA (`obj$pca_model$x`), a projeção principal é usada diretamente.

3. Aplicação do LOF (Local Outlier Factor)

- **Agregação**: os dados normalizados de treino (normais) e teste (anômalos) são combinados em um único conjunto para cálculo do LOF.
- **Parâmetro k**: o número de vizinhos (`minPts`) é definido como 20, o que influencia a sensibilidade à densidade local.
- **Cálculo dos Scores**: o vetor de LOF é separado novamente em treino e teste após o cálculo conjunto.





Processo de implementação no R

4. Visualização dos Resultados

- Histogramas e gráficos de densidade são gerados para comparar a distribuição dos scores LOF dos dados normais e anômalos.
- Um limiar de corte (threshold) é definido como o percentil 85% dos scores normais (treino), representado graficamente.

5. Detecção de Anomalias

- Os dados de teste são classificados como anômalos se o LOF estiver acima do threshold definido.
- O número total de anomalias detectadas é exibido.

6. Avaliação do Modelo

- Como todos os dados de teste são anômalos, os rótulos reais são `1`.
- A matriz de confusão é montada com os casos verdadeiros positivos (TP) e falsos negativos (FN).
- São calculadas as métricas de:
  - **Acurácia**: proporção de anomalias corretamente detectadas.
  - **Recall (Sensibilidade)**: proporção de anomalias reais que foram detectadas corretamente.

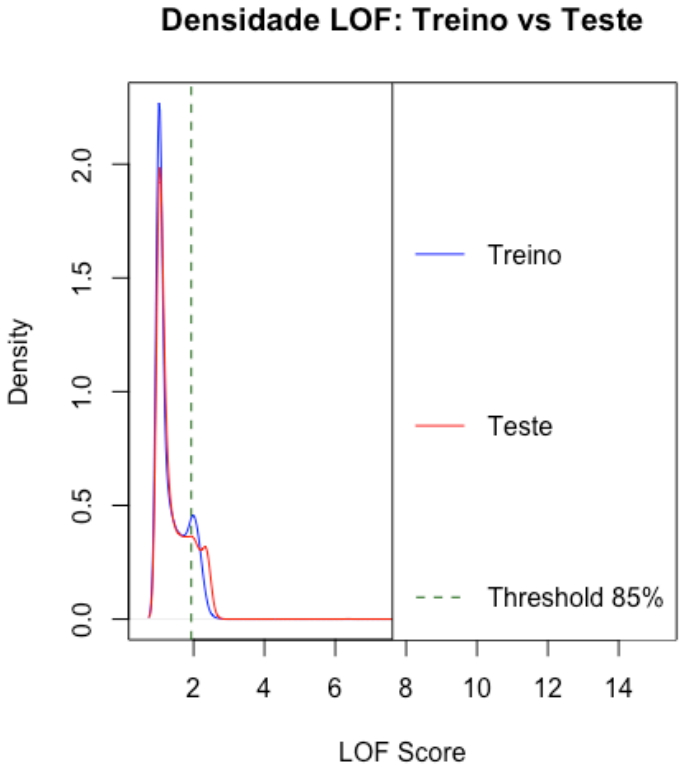
7. Armazenamento dos Resultados

- Os scores LOF dos dados de teste são salvos em um arquivo `.rds` para análise futura ou reuso.

Processo de implementação no R

Avaliação dos Resultados

- Visualização gráfica para destacar padrões e anomalias (cores indicam pontos suspeitos).
- Quando disponível, os rótulos foram usados para calcular métricas como:
  - Acurácia
  - Precisão
  - Sensibilidade
- Benchmark de performance realizado com **medição do tempo de execução**.



RESULTADOS

UFRPE
Universidade Federal Rural de Pernambuco

- Amostra de teste, 6000 ruidos anómalos.

	Real: Anomalia (1)
Previsto: Anomalia (1)	1112
Previsto: Normal (0)	5138

Métrica	Valor
Acurácia (Accuracy)	≈21,64%
Recall (Sensibilidade)	≈21,64%

Exercícios práticos — LOF no R

UFRPE
Universidade Federal Rural de Pernambuco

📦 1. Preparação do Ambiente

Instale e carregue os pacotes necessários:
`DMwR2`, `ggplot2`, `caret`, e `datasets` (caso use conjuntos embutidos).

🗂️ **2. Importação e Normalização dos Dados**

a) Importe um conjunto de dados `.csv` de sua escolha ou use o `iris`.
b) Normalize as variáveis numéricas.

🕵️ **3. Aplicação do LOF**

a) Aplique o LOF com **k = 5**
b) Visualize os scores gerados
c) Adicione uma coluna "Anomalia" indicando se `LOF > 1.5`

🎛️ **4. Ajuste de Hiperparâmetros**

a) Repita o LOF com **k variando de 3 a 20**
b) Compare os resultados: quais valores geram mais ou menos anomalias?

🧬 **5. Redução de Dimensionalidade (Desafio!)**

Aplique **PCA** e mantenha 2 ou 3 componentes principais
Reaplique o LOF e compare o desempenho (tempo e anomalias)

📈 **6. Visualização e Avaliação**

a) Faça um gráfico com `ggplot2` mostrando os pontos anômalos

Propostas de Exercícios para Aprendizado do LOF no R

UFRPE
Universidade Federal Rural de Pernambuco

🧠 1. Entendendo o LOF

Objetivo: Compreender como o LOF detecta anomalias e sua relação com a densidade dos dados.
Exercício:
a) Explique, com suas palavras, o que é o LOF e como ele identifica outliers.
b) Crie um gráfico para comparar dados "normais" e "anômalos" com LOF aplicado.

🔍 2. Trabalhando com Dados Reais

Objetivo: Aplicar o LOF em um conjunto de dados real e explorar suas funcionalidades.
Exercício:
a) Encontre um conjunto de dados que tenha **anomalias claras** (por exemplo, `iris`, `mtcars`, ou um dataset de sua escolha).
b) Aplique o LOF e identifique as anomalias.
c) Visualize as anomalias no gráfico, destacando as observações "suspeitas".

🎯 3. Ajuste de Parâmetros

Objetivo: Aprender como o parâmetro **k** impacta os resultados do LOF.
Exercício:
a) Teste o LOF para diferentes valores de **k** (ex: 3, 5, 10, 20) e veja como isso altera os scores.
b) Discuta como o número de vizinhos pode influenciar a detecção de anomalias (mais vizinhos → menos anomalias detectadas?).

🧩 4. Análise de Tempo e Eficiência

Objetivo: Avaliar o desempenho do LOF em termos de **tempo de execução** e **uso de memória**.
Exercício:
a) Aplique o LOF em uma base de dados grande (por exemplo, com centenas ou milhares de linhas).
b) Meça o tempo de execução utilizando `system.time()`.
c) Compare o tempo de execução antes e depois de realizar **redução de dimensionalidade** com PCA.

💡 5. Análise de Resultados e Métricas

Objetivo: Avaliar a precisão do modelo utilizando métricas de desempenho.
Exercício:
a) Se o seu conjunto de dados tiver rótulos (labels), calcule a **matriz de confusão**.
b) Avalie as métricas como **precisão**, **recall**, e **F1-score** usando a função `confusionMatrix` do pacote **caret**.

🌐 6. LOF em Aplicações do Mundo Real

Objetivo: Explorar o uso do LOF em diferentes áreas de aplicação.
Exercício:
a) Pesquise 3 exemplos de como o LOF é usado no mercado (por exemplo, para detecção de fraudes financeiras, manutenção preditiva, etc.).
b) Discuta como o LOF pode ser aplicado de forma prática nesses cenários.

🎨 7. Visualização Criativa

Objetivo: Criar visualizações que ajudem a entender a distribuição e os outliers dos dados.
Exercício:
a) Crie **diferentes tipos de gráficos** (como `scatter plots`, `boxplots`, ou gráficos de densidade) para visualizar os dados com e sem os outliers.
b) Experimente também usar diferentes representações de **cores** ou **formas** para destacar as anomalias.

Referências

- Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF: Identifying Density-Based Local Outliers. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (pp. 93-104). ACM. https://doi.org/10.1145/342009.335388
- Hodge, V. J., & Austin, J. (2004). A Survey of Outlier Detection Methodologies. Artificial Intelligence Review, 22(2), 85-126. https://doi.org/10.1023/B:AIRE.0000044037.98894.87
- Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly Detection: A Survey. ACM Computing Surveys (CSUR), 41(3), 1-58. https://doi.org/10.1145/1541880.1541882
- Campos, G., Zimek, A., Sander, J., et al. (2016). On the Evaluation of Unsupervised Anomaly Detection: Measures, Datasets, and an Empirical Study. Data Mining and Knowledge Discovery, 30(4), 891-927. https://doi.org/10.1007/s10618-015-0449-8
- López, V., García, S., & Herrera, F. (2012). A Survey on Statistical Approaches for the Analysis of Class Imbalance in Data Mining. Computational Statistics & Data Analysis, 56(5), 1492-1515. https://doi.org/10.1016/j.csda.2010.12.016
- R Core Team. (2024). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. https://www.R-project.org
- López, V., García, S., & Herrera, F. (2013). An Evaluation of the LOF and its Variants in High-Dimensional Data. In Proceedings of the European Conference on Artificial Intelligence (pp. 35-42). IOS Press. https://doi.org/10.3233/978-1-61499-290-9-35