



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO E SISTEMAS
CURSO DE ENGENHARIA DE PRODUÇÃO CIVIL

Tiago Lopes de Souza

**Previsão de Desastres Associados a Eventos Climáticos Extremos no Estado
de Santa Catarina: Um Estudo de Caso Baseado em Redes Neurais e Formação
de Clusters Georreferenciados**

Florianópolis

2025

Tiago Lopes de Souza

Previsão de Desastres Associados a Eventos Climáticos Extremos no Estado de Santa Catarina: Um Estudo de Caso Baseado em Redes Neurais e Formação de Clusters Georreferenciados

Trabalho de Conclusão de Curso submetido ao curso de Engenharia de Produção Civil do Centro Tecnológico da Universidade Federal de Santa Catarina como requisito parcial para a obtenção do título de Bacharel em Engenharia Civil com habilitação em Engenharia de Produção

Orientador(a): Prof. Dr. Ricardo Villarroel Dávalos

Florianópolis

2025

Ficha de identificação da obra

Souza, Tiago Lopes

Previsão de Desastres Associados a Eventos Climáticos
Extremos no Estado de Santa Catarina: Um Estudo de Caso
Baseado em Redes Neurais e Formação de Clusters
Georreferenciados / Tiago Lopes Souza ; orientador,
Ricardo Villarroel Davalos, 2025.

72 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro
Tecnológico, Graduação em Engenharia de Produção Civil,
Florianópolis, 2025.

Inclui referências.

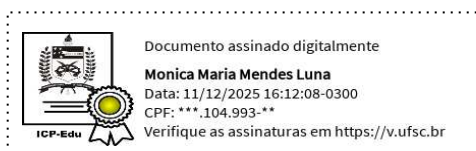
1. Engenharia de Produção Civil. 2. Previsão de
Desastres. 3. Redes Neurais. 4. Logística Humanitária. 5.
Clusterização. I. Davalos, Ricardo Villarroel. II.
Universidade Federal de Santa Catarina. Graduação em
Engenharia de Produção Civil. III. Título.

Tiago Lopes de Souza

Previsão de Desastres Associados a Eventos Climáticos Extremos no Estado de Santa Catarina: Um Estudo de Caso Baseado em Redes Neurais e Formação de Clusters Georreferenciados

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do título de Bacharel em Engenharia Civil com habilitação em Engenharia de Produção e aprovado em sua forma final pelo Curso Engenharia de Produção Civil

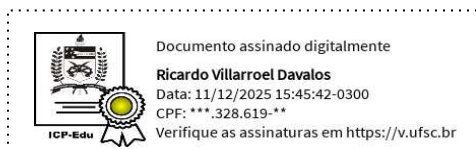
Florianópolis, 04 de dezembro de 2025



Prof.^a Mônica Maria Mendes Luna, Dr.^a.

Coordenação do Curso

Banca examinadora



Prof. Ricardo Villarroel Dávalos, Dr.

Orientador

Prof. Sérgio Fernando Mayerle, Dr.
Universidade Federal de Santa Catarina

Prof.^a Fabiana Santos Lima, Dr.^a.
Universidade Federal de Santa Catarina

Florianópolis, 2025.

AGRADECIMENTOS

Agradeço aos meus familiares que me apoiaram e me deram suporte em todos os momentos deste trabalho e da graduação.

Aos meus colegas que me acompanharam nesta jornada ao longo dos anos deste curso e sempre me incentivaram a seguir em frente.

E um agradecimento em especial ao professor Ricardo Villarroel Dávalos por me apoiar, motivar e ajudar em todos os momentos de necessidade.

RESUMO

A ocorrência recorrente de desastres naturais no estado de Santa Catarina tem provocado impactos significativos sobre a população, a infraestrutura e o meio ambiente, evidenciando a necessidade de soluções tecnológicas que permitam prever e mitigar tais eventos com maior eficiência. Diante desse cenário, este trabalho propõe a aplicação de redes neurais recorrentes do tipo Long Short-Term Memory (LSTM) para a previsão de eventos climáticos extremos, utilizando dados históricos meteorológicos e socioambientais como base de aprendizado. Além da etapa de previsão, são empregadas técnicas de clusterização, em especial o algoritmo K-Means, com o objetivo de agrupar os municípios de acordo com a similaridade de ocorrência e intensidade dos eventos previstos, possibilitando a otimização da logística e da alocação de recursos emergenciais. Os resultados esperados incluem o aprimoramento da capacidade de resposta das autoridades competentes, a redução de perdas humanas e materiais e o fortalecimento das políticas públicas de gestão de risco e adaptação climática. Assim, a abordagem proposta busca contribuir para o desenvolvimento de ferramentas inteligentes para apoiar o monitoramento e planejamento preventivo,

Palavras-chave: Previsão de Desastres, Redes Neurais, LSTM, Clusterização, Logística Humanitária, K-Means, Gestão De Desastres, Santa Catarina.

ABSTRACT

The recurrent occurrence of natural disasters in the state of Santa Catarina has caused significant impacts on the population, infrastructure, and the environment, highlighting the need for technological solutions capable of predicting and mitigating such events more efficiently. In this context, this study proposes the application of Long Short-Term Memory (LSTM) recurrent neural networks for forecasting extreme weather events, using historical meteorological and socio-environmental data as the learning basis. In addition to the forecasting stage, clustering techniques, particularly the K-Means algorithm, are employed to group municipalities according to the similarity in occurrence and intensity of the predicted events, enabling the optimization of logistics and the allocation of emergency resources. The expected outcomes include improved response capacity of public authorities, reduced human and material losses, and strengthened public policies for risk management and climate adaptation. Thus, the proposed approach aims to contribute to the development of intelligent tools to support preventive planning and monitoring.

Keywords: Disaster Forecasting, Neural Networks, LSTM, Clustering, Humanitarian Logistics, K-Means, Disaster Management, Santa Catarina.

LISTA DE FIGURAS

Figura 1 – Ciclo da logística humanitária	23
Figura 2 - Modelo não Linear	24
Figura 3 - Representação da RNN	26
Figura 4 - Etapas metodológicas	35
Figura 5 - Fluxo do projeto.....	37
Figura 6 - Mapeamento e limpeza dos dados.....	38
Figura 7 - Mapeamento dos eventos	38
Figura 8 - Eventos mensais	39
Figura 9 - Frequência de eventos	40
Figura 10 - Quantidade de eventos por município	41
Figura 11 – Features	42
Figura 12 - Normalização e escala	42
Figura 13 - Janelamento temporal	42
Figura 14 - Split temporal.....	43
Figura 15 - Modelo LSTM	44
Figura 16 - Escolha do k ótimo	45
Figura 17 - Argumentos para previsão	45
Figura 18 - Teste de previsão	46
Figura 19 - Histórico de treino.....	47
Figura 20 - Clusterização dos eventos	48
Figura 21 - Georreferenciamento dos clusters por município	51

LISTA DE ABREVIATURAS E SIGLAS

LSTM	Long Short Term Memory
MAE	Erro Médio Absoluto
RMSE	Raiz do Erro Quadrático Médio
RNA	Redes Neurais Artificiais
CNN	Redes Neurais convolucionais
RNN	Redes Neurais Recorrentes
SEDEC	Secretaria Nacional de Proteção e Defesa Civil

SUMÁRIO

1	INTRODUÇÃO.....	15
1.1	PROBLEMA DE PESQUISA.....	16
1.2	OBJETIVOS.....	17
1.2.1	Objetivo Geral.....	17
1.2.2	Objetivo Específico.....	18
1.3	JUSTIFICATIVA.....	18
1.4	ESTRUTURA DO TRABALHO.....	19
2	FUNDAMENTAÇÃO TEÓRICA.....	20
2.1	DESASTRE NATURAIS.....	20
2.1.1	PREVISÃO DE DESASTRE NATURAIS.....	20
2.2	LOGÍSTICA HUMANITÁRIA	21
2.3	ANÁLISE DE DADOS	23
2.4	REDES NEURAIS	25
2.4.1	Redes Neurais Recorrentes (RNN) e LSTMs.....	26
2.5	CLUSTERING E OTIMIZAÇÃO LOGÍSTICA	28
2.5.1	K-MEANS: Um método de Clustering para Logística Humanitária	29
2.6	MÉTRICAS DE AVALIAÇÃO	30
2.6.1	RMSE – Raiz do Erro Quadrático Médio	31
2.6.2	MAE – Erro Médio Absoluto	31
2.7	CONSIDERAÇÕES FINAIS DO CAPÍTULO.....	32
3	METODOLOGIA.....	34
3.1	MATERIAL E MÉTODOS.....	34
3.1.1	Abordagem Quantitativa	34
3.1.2	Estudo de Caso	35
3.2	PROCEDIMENTOS METODOLOGICOS	35
3.3	DELIMITAÇÕES	36
4	ESTUDO DE CASO.....	37
4.1	COLETA E TRATAMENTO DE DADOS	37
4.2	ANÁLISE EXPLORATÓRIA	39
4.3	APLICAÇÃO DE REDES NEURAIS	41
4.3.1	Features e Normalização	41

4.3.2	Janelamento Temporal	42
4.3.3	Split Temporal	43
4.3.4	Modelo LSTM.....	43
4.4	CRIAÇÃO DOS CLUSTERS	44
4.5	RESULTADOS E DISCUSSÕES	45
5	CONCLUSÕES E TRABALHOS FUTUROS.....	52
5.1	CONCLUSÕES	52
5.2	TRABALHOS FUTUROS	54
	REFERÊNCIAS	56
	APENDICE A – CÓDIGO REDES NEURAIS LSTM	61
	APENDICE B – CÓDIGO CLUSTER K-MEANS	68

1 INTRODUÇÃO

A previsão de desastres naturais é um desafio global que demanda soluções inovadoras e tecnologicamente avançadas, considerando o aumento da frequência e intensidade desses eventos devido às mudanças climáticas. Tais desastres como enchentes, secas, deslizamentos de terra e tempestades severas causam impactos significativos em vidas humanas, infraestrutura urbana e na economia local e nacional (Smith, 2013). Segundo dados do Painel Intergovernamental sobre Mudanças Climáticas (IPCC, 2021), as alterações nos padrões climáticos têm intensificado a ocorrência de eventos extremos, ampliando a necessidade de sistemas eficazes de monitoramento e resposta. No Brasil, o estado de Santa Catarina é especialmente vulnerável a fenômenos climáticos extremos, como enchentes, tempestades e deslizamentos de terra, que frequentemente resultam em situações de emergência e calamidade pública (Joner; Ávila; Mattedi, 2021). Esses eventos tornam evidente a urgência de estratégias mais eficientes para mitigação e preparação, baseadas em tecnologias que permitam não apenas prever catástrofes com antecedência, mas também organizar de forma mais eficaz a resposta emergencial.

Com os avanços na tecnologia e na ciência de dados, a inteligência artificial tem se mostrado uma ferramenta importante na previsão de fenômenos naturais. Entre as abordagens mais eficazes estão as redes neurais artificiais, com destaque para as redes Long Short-Term Memory (LSTM), que demonstram grande capacidade de identificar padrões complexos em séries temporais meteorológicas e prever a ocorrência de eventos extremos (Zhang *et al.*, 2019). Esses modelos têm sido amplamente utilizados em diversas aplicações, desde a previsão de chuvas intensas até a identificação de riscos de deslizamentos de terra. No entanto, prever um desastre é apenas uma parte da solução; é igualmente crucial que a resposta a esses eventos seja rápida e eficaz.

A distribuição de itens humanitários em situações de crise é um desafio logístico significativo. O planejamento inadequado pode levar a atrasos na entrega de suprimentos essenciais, como água, alimentos e medicamentos, aumentando o sofrimento das populações afetadas. Diversos fatores como a quantidade de atores envolvidos na ajuda humanitária e a escassez de recursos suficientes dificultam na coordenação de socorro (Balcik *et al.*, 2003).

Segundo Sheu (2010), um sistema logístico eficiente para distribuição de recursos humanitários deve considerar fatores como acessibilidade, tempo de resposta e disponibilidade de suprimentos, garantindo que as regiões mais afetadas recebam a assistência necessária no menor tempo possível. Isso se torna especialmente relevante em contextos onde a infraestrutura local pode estar comprometida devido ao impacto do desastre, dificultando o transporte e a entrega de recursos essenciais.

Outros estudos indicam que a integração de modelos de previsão climática com estratégias logísticas pode aumentar significativamente a eficiência das operações de resposta (Chen *et al.*, 2020). Isso ocorre porque, ao antecipar eventos extremos, as autoridades podem planejar previamente a localização e a quantidade de suprimentos necessários, reduzindo os impactos e melhorando a resiliência das comunidades afetadas.

Diante desse contexto, este trabalho tem como objetivo integrar modelo de previsão baseado em redes neurais e estratégias de otimização logística de *clusters* para aprimorar a resposta a desastres naturais em Santa Catarina. Acredita-se que essa abordagem pode contribuir significativamente para a tomada de decisões das autoridades, auxiliando na redução dos impactos dos eventos climáticos extremos e fortalecendo a resiliência das comunidades afetadas.

1.1 PROBLEMA DE PESQUISA

Santa Catarina enfrenta desafios recorrentes relacionados a desastres naturais, devido à sua localização geográfica e condições climáticas. Enchentes, deslizamentos de terra e tempestades são frequentes, causando danos severos à infraestrutura, à economia e à qualidade de vida da população. Segundo a Defesa Civil de Santa Catarina (2021), o estado registrou dezenas de eventos climáticos extremos nos últimos anos, exigindo constantes mobilizações emergenciais.

A necessidade de uma resposta rápida e eficiente a esses eventos é evidente. A previsão meteorológica tradicional, embora avançada, ainda apresenta limitações na previsão de eventos extremos com precisão espacial e temporal adequadas. Além disso, a logística de distribuição de suprimentos para as áreas afetadas é um desafio crítico, exigindo estratégias baseadas em análise de dados e otimização de recursos.

A história recente de Santa Catarina é marcada por eventos catastróficos, como as enchentes de 2008, que afetaram mais de 1,5 milhão de pessoas e resultaram em prejuízos superiores a R\$ 4 bilhões (Silva *et al.*, 2015). Além dos impactos econômicos, esses eventos também comprometem a infraestrutura, com estradas interditadas, comunidades isoladas e interrupção de serviços essenciais.

A necessidade de uma resposta rápida e eficiente a esses eventos é evidente. A previsão meteorológica tradicional, embora avançada, ainda apresenta limitações na previsão de eventos extremos com precisão espacial e temporal adequadas. Segundo estudos recentes, a integração de redes neurais na previsão de desastres pode aumentar a precisão na antecipação de eventos climáticos severos (Garcia *et al.*, 2020). No entanto, mesmo com previsão eficiente, a gestão de desastres depende de uma logística bem estruturada para distribuição de recursos humanitários.

Atualmente, mais de 90% da assistência humanitária internacional é destinada à resposta emergencial, enquanto uma parcela menor é voltada para medidas preventivas. Dessa forma, investir em ações preparatórias, como o planejamento estratégico para emergências, a implementação de centros operacionais de crise e o armazenamento antecipado de suprimentos essenciais, pode resultar em uma distribuição mais eficiente da ajuda e em um atendimento mais eficaz às populações afetadas (Maharjan e Hanaoka, 2017). A utilização de métodos computacionais para otimizar a alocação desses centros é essencial para minimizar os impactos dos desastres (Rocha & Almeida, 2021).

1.2 OBJETIVOS

Neste tópico são apresentados os objetivos gerais e específicos deste trabalho.

1.2.1 Objetivo Geral

O presente trabalho tem como objetivo desenvolver um modelo de previsão de desastres naturais para o estado de Santa Catarina utilizando redes neurais do tipo Long Short-Term Memory (LSTM), além de aplicar técnicas de clusterização para aprimorar a capacidade de antecipação de eventos extremos e aumentar a eficiência logística na resposta a desastres, reduzindo impactos sociais e econômicos.

1.2.2 Objetivo Específico

Para atingir o objetivo geral, este estudo propõe os seguintes objetivos específicos:

- a) Analisar o histórico de desastres naturais em Santa Catarina, identificando padrões e fatores climáticos que contribuem para sua ocorrência;
- b) Coletar e preparar dados históricos, garantindo sua adequação para o treinamento do modelo preditivo baseado em redes neurais LSTM;
- c) Desenvolver e avaliar um modelo de redes neurais LSTM, verificando sua capacidade de prever eventos climáticos;
- d) Aplicar a técnica K-Means de clusterização para definir otimização logística;

1.3 JUSTIFICATIVA

Os desastres naturais representam um dos maiores desafios enfrentados pelo estado de Santa Catarina, afetando diretamente a população, a economia e a infraestrutura local. A recorrência de eventos como enchentes e deslizamentos de terra evidencia a necessidade de soluções mais eficazes para melhorar a capacidade de previsão e resposta a essas ocorrências. Tecnologias baseadas em inteligência artificial, como as redes neurais do tipo Long Short-Term Memory (LSTM), têm se destacado por sua capacidade de identificar padrões complexos em séries temporais meteorológicas, contribuindo significativamente para a previsão de eventos extremos.

Por exemplo, Sarwade, Shinde e Patil (2024) demonstraram que modelos LSTM podem prever com precisão eventos de precipitação extrema, utilizando variáveis meteorológicas como temperatura, umidade e precipitação.

Além da previsão precisa, a logística humanitária desempenha um papel essencial na mitigação dos impactos dos desastres. A alocação estratégica de centros de distribuição de suprimentos pode reduzir o tempo de resposta e melhorar a eficiência na entrega de recursos às comunidades afetadas (Balci & Beyhan, 2021). Métodos como o K-Means permitem agrupar áreas afetadas com base em variáveis como densidade populacional, acessibilidade e extensão dos danos, facilitando a

identificação de zonas prioritárias de intervenção (Balci; Beyhan, 2021). Essa estratégia contribui para a redução do tempo de resposta e melhora a eficiência na distribuição de suprimentos essenciais, como alimentos, água potável e medicamentos.

Segundo Zhang et al. (2018), a aplicação de clusterização em operações logísticas pode reduzir significativamente os custos operacionais e aumentar a cobertura das áreas afetadas, ao permitir o planejamento mais estratégico dos centros de distribuição. Além disso, Oliveira *et al.* (2023) destacam que a integração de algoritmos de agrupamento com heurísticas de otimização pode potencializar os resultados em cenários complexos e dinâmicos, como os de desastres naturais.

1.4 ESTRUTURA DO TRABALHO

No primeiro capítulo do presente trabalho é introduzido brevemente a contextualização do objetivo do estudo, assim como os problemas identificados e justificativa.

No segundo capítulo é abordado a fundamentação teórica, onde são introduzidos os conceitos necessários para o desenvolvimento do trabalho, através de referenciais teóricos que tratam as técnicas utilizadas.

O terceiro capítulo apresenta a etapa metodológica, e as principais ferramentas utilizadas para a elaboração dos objetivos.

O quarto capítulo é destinado à apresentação dos resultados encontrados através da implementação das ferramentas apresentadas

No quinto e capítulo é apresentado as conclusões referentes ao trabalho desenvolvido.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 DESASTRE NATURAIS

Desastres naturais são fenômenos extremos resultantes de processos naturais intensificados, como terremotos, erupções vulcânicas, deslizamentos, enchentes, furacões, tornados, tsunamis, incêndios e tempestades. Esses eventos podem causar prejuízos significativos tanto em termos materiais quanto em vidas humanas. Devido a esses impactos, qualquer iniciativa voltada para sua prevenção ou gerenciamento é de extrema relevância, pois contribui para a redução dos danos e para uma resposta mais eficiente diante dessas ocorrências (Braga et al. 2011).

Segundo Smith (2013), esses eventos podem ser classificados em hidrológicos, meteorológicos, geofísicos e climatológicos, dependendo da origem do fenômeno. No contexto brasileiro, e especialmente em Santa Catarina, as enchentes, deslizamentos e tempestades são os desastres mais recorrentes devido às características climáticas e geográficas do estado (Almeida et al., 2018).

A crescente urbanização e a ocupação de áreas de risco aumentam a vulnerabilidade da população aos desastres naturais (Santos et al., 2020). E segundo Castro (2003) o fator predominante na intensificação do desastre é o grau de vulnerabilidade do local afetado. Isso reforça a necessidade de estratégias eficazes de monitoramento, previsão e mitigação.

2.1.1 PREVISÃO DE DESASTRE NATURAIS

Desastres naturais são eventos extremos que resultam de fenômenos naturais, como enchentes, terremotos, deslizamentos de terra, furacões, entre outros, e podem causar danos significativos à população e infraestrutura de uma região. No contexto de Santa Catarina, enchentes, deslizamentos e tempestades são recorrentes devido às condições climáticas e geográficas do estado. A previsão precisa desses eventos é fundamental para reduzir os danos causados, permitindo uma resposta rápida e eficiente.

Tradicionalmente, a previsão de desastres naturais é realizada através de modelos meteorológicos que analisam parâmetros como volume de chuvas,

temperatura, umidade e pressão atmosférica. No entanto, a previsão de desastres naturais com alto grau de precisão espacial e temporal continua sendo um desafio, pois os métodos tradicionais, muitas vezes, falham em capturar a complexidade dos fenômenos climáticos extremos (Silva *et al.*, 2015).

A introdução de técnicas mais avançadas, como as redes neurais, oferece uma abordagem mais robusta para prever desastres, identificando padrões complexos em grandes volumes de dados históricos e proporcionando previsões mais precisas. Estudos demonstram que a implementação dessas técnicas melhora significativamente a precisão das previsões e reduz os impactos dos desastres naturais (Ferreira *et al.*, 2020). Além disso, a utilização de dados de satélites combinados com algoritmos de aprendizado profundo pode aprimorar ainda mais a eficácia dessas previsões (Li *et al.*, 2021).

2.2 LOGÍSTICA HUMANITÁRIA

A logística humanitária refere-se à gestão eficiente de recursos, serviços e informações destinados a mitigar os efeitos de desastres naturais, assegurando que a assistência chegue de maneira eficaz às populações afetadas. Essa logística abrange atividades como transporte, armazenamento e distribuição de suprimentos essenciais, tais como alimentos, água potável, medicamentos e equipamentos de emergência. No contexto brasileiro, a logística humanitária tem ganhado relevância devido à frequência de eventos climáticos extremos e à necessidade de respostas rápidas e coordenadas (Silva, 2021).

A gestão logística em situações de crise apresenta desafios significativos, especialmente em ambientes de alta complexidade e incerteza, onde infraestruturas podem estar comprometidas e os sistemas de comunicação e transporte, limitados. Essas dificuldades são exacerbadas pela carência de estudos relacionados ao preparo e resposta de operações, evidenciando a necessidade de aprimoramento na gestão logística durante desastres (Silva; Santos; Oliveira, 2022).

A logística humanitária está intrinsecamente ligada ao conceito de cadeia de suprimentos, que envolve o planejamento e a execução de uma rede de distribuição eficiente para otimizar os tempos de resposta. Estudos demonstram que a utilização de modelos matemáticos, como a programação linear e algoritmos de otimização, pode melhorar significativamente a eficiência do transporte e a alocação dos centros

de distribuição de ajuda, resultando em economia de tempo e custos (Balci; Beyhan, 2021).

Além disso, novas abordagens tecnológicas, como o uso de ferramentas de geolocalização, sensoriamento remoto e inteligência artificial, têm transformado a logística humanitária. A integração de sistemas de previsão de desastres com as operações logísticas tem o potencial de reduzir o tempo de resposta, identificar os pontos mais críticos e garantir a entrega de recursos de forma mais precisa (PARK et al., 2020).

A logística humanitária é um campo multifacetado que exige não apenas eficiência operacional, mas também uma grande capacidade de adaptação, inovação e colaboração entre diversos atores, como governos, organizações não governamentais (ONGs) e empresas privadas. Em muitos cenários de desastre, a colaboração interinstitucional e a coordenação entre diferentes atores podem ser decisivas para o sucesso das operações de ajuda (Neumann, 2017).

A logística humanitária é composta por um ciclo que abrange todas as fases de um desastre: preparação, resposta e reconstrução. Cada etapa exige estratégias logísticas específicas, voltadas para garantir o abastecimento de recursos essenciais de forma eficiente e eficaz. Segundo Lima *et al.* (2016), compreender esse ciclo é fundamental para estruturar operações que minimizem os impactos dos desastres e maximizem a capacidade de resposta das instituições envolvidas.

A fase de preparação envolve o planejamento e a organização dos recursos antes da ocorrência do desastre. Essa etapa inclui a definição de centros de distribuição, rotas logísticas, parcerias com fornecedores e capacitação das equipes envolvidas. Como apontado por Silva e Silva (2019), a antecipação e o mapeamento de riscos são cruciais para garantir maior agilidade no momento da resposta. Nessa fase, também se busca a criação de estoques estratégicos e a elaboração de planos de contingência para diferentes cenários.

Durante a fase de resposta, o foco se volta para o atendimento imediato das necessidades das populações afetadas. Trata-se da etapa mais crítica do ciclo, uma vez que a eficácia da distribuição de suprimentos pode ser determinante para a sobrevivência das vítimas. De acordo com Gomes e Baptista (2017), os principais desafios logísticos nesse momento incluem a escassez de tempo, a infraestrutura danificada e a dificuldade de acesso às áreas afetadas. Nessa fase, decisões rápidas

e precisas sobre alocação de recursos, priorização de demandas e coordenação entre diferentes atores são essenciais.

Já a fase de reconstrução visa restaurar a normalidade nas regiões atingidas, reestruturando os sistemas logísticos, sociais e econômicos. Ela envolve o reabastecimento dos estoques utilizados na resposta, a avaliação da eficiência das ações realizadas e o aprimoramento dos planos para eventos futuros. Para Oliveira e Sousa (2021), a reconstrução deve ser integrada com políticas de prevenção e resiliência, promovendo ações sustentáveis que reduzam a vulnerabilidade das comunidades em longo prazo.

Figura 1 – Ciclo da logística humanitária



Fonte: Autor

Além dessas três fases principais, alguns autores consideram a mitigação como uma etapa complementar, voltada para reduzir os riscos antes mesmo da preparação (Lima *et al.*, 2016). Essa abordagem preventiva visa ações estruturais, como obras de contenção, bem como ações educativas e de conscientização.

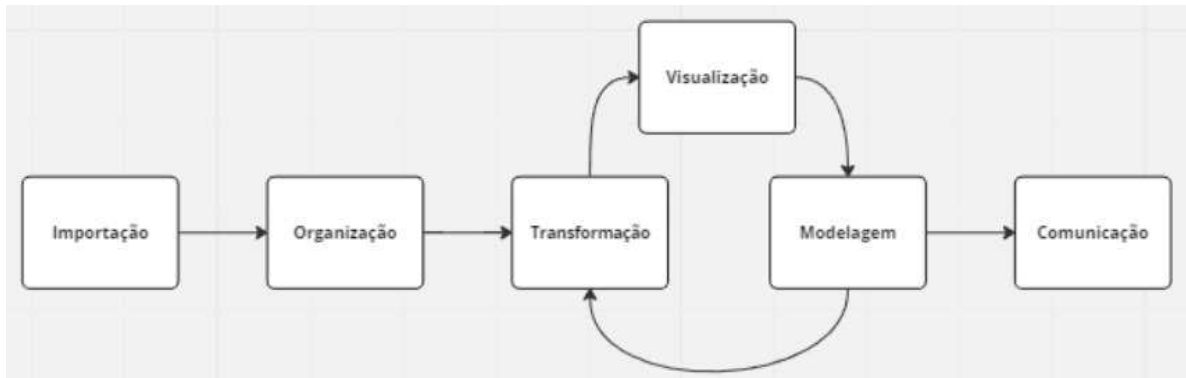
2.3 ANÁLISE DE DADOS

A definição de Silvestre (2007) sobre a análise de dados destaca a importância de personalizar os métodos de acordo com o objetivo de cada análise. No caso de análise de desastre, a personalização é essencial, pois cada conjunto de

dados coletado representam um tipo de desastre de acordo com o evento, como enchentes, terremotos, deslizamentos de terra, furacões.

Além disso, o processo de análise de dados aplicado a eventos catastróficos segue um fluxo cíclico, conforme descrito por Wickham (2023), que pode ser dividido nas seguintes etapas, como descrito na figura 2:

Figura 2 - Modelo não Linear



Fonte: Zaghi (2023).

Onde este processo pode ser dividido em seis etapas:

- i. Importação: Etapa de coleta de dados nos diversos banco de dados;
- ii. Organização: Os dados coletados são organizados e categorizados de acordo com as variáveis relevantes;
- iii. Transformação: Processo de manipulação dos dados, onde esta é uma etapa cíclica que visa aperfeiçoar a análise de acordo com implementação da etapa de modelagem;
- iv. Visualização: Ilustração gráfica e de tabelas das informações contidas no banco de dados, etapa que facilita o entendimento e interpretação dos dados e resultados;
- v. Modelagem: Etapa para aplicação de modelos preditivos, identificação de relação entre as variáveis. Esta etapa em conjunto com as etapas III e IV são cíclicas, de acordo com a necessidade de adaptação do modelo e interesse.
- vi. Comunicação: Etapa final com intuito de demonstrar os resultados obtidos através de gráfico ou relatórios que ilustram a avaliação e testes de hipóteses.

Muitas vezes, as fontes de dados não fornecem uma estrutura pronta para a realização de estudos e análises. Isso demanda um processo essencial de

organização e transformação da base de dados. Segundo Wickham (2023), esse processo, que abrange a organização e transformação de dados, é uma série de ações que envolvem a modificação dos dados. Inicialmente, busca-se interpretar as variáveis existentes, organizando-as em classes e formatos adequados. Posteriormente, se necessário, são criadas variáveis e colunas que descrevem outros comportamentos presentes em cada uma das observações do conjunto.

Em suma, a análise de dados aplicada não apenas transforma dados em informações úteis, mas também desempenha um papel crucial na predição de eventos. Com o uso de técnicas como redes neurais, é possível identificar padrões e tendências ocultas nos dados, permitindo uma gestão mais eficiente e estratégica.

2.4 REDES NEURAIIS

As redes neurais artificiais (RNA) são modelos computacionais inspirados no funcionamento do cérebro humano, compostos por camadas de unidades chamadas neurônios artificiais, que processam informações e aprendem com os dados. Elas são extremamente eficazes para resolver problemas complexos, como a previsão de séries temporais, um desafio recorrente em meteorologia e em estudos sobre desastres naturais (Zhang *et al.*, 2019).

Uma rede neural é composta por camadas de neurônios artificiais organizadas em uma arquitetura específica: uma camada de entrada, uma ou mais camadas ocultas, e uma camada de saída. Cada neurônio em uma camada está conectado a outros neurônios da camada seguinte, formando um sistema interligado que processa informações de maneira hierárquica. O processo de aprendizado das redes neurais envolve a otimização de pesos associados a essas conexões, ajustando-os de modo que a rede seja capaz de aprender a partir dos dados de treinamento e realizar previsões a partir de novos dados (Goodfellow; Bengio; Courville, 2016).

As redes neurais convolucionais (CNNs) e redes neurais recorrentes (RNNs) têm sido amplamente aplicadas na análise de padrões climáticos e previsão de eventos extremos. Trabalhos recentes apontam que a combinação dessas arquiteturas pode melhorar a robustez dos modelos e permitir previsões mais confiáveis (Guo *et al.*, 2022). Além disso, o uso de técnicas como o aprendizado por reforço vem sendo explorado para melhorar a adaptação dos modelos a cenários em constante mudança (Ribeiro *et al.*, 2023).

2.4.1 Redes Neurais Recorrentes (RNN) e LSTMs

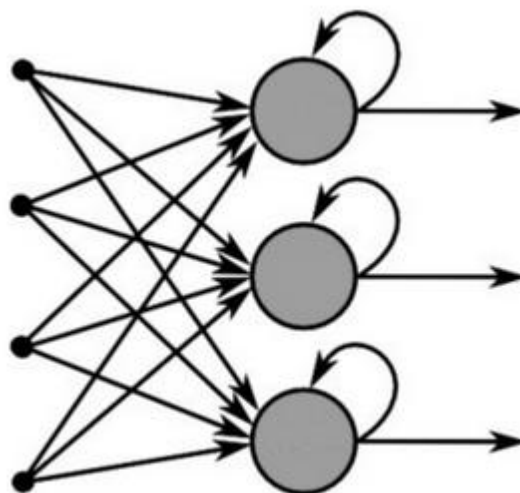
As Redes Neurais Recorrentes (RNN) são projetadas especificamente para trabalhar com dados sequenciais e temporais. As RNNs conseguem armazenar informações de estados anteriores para influenciar decisões futuras.

Contudo, as RNNs convencionais sofrem com o problema de desvanecimento de gradiente, o que pode comprometer seu desempenho ao lidar com sequências longas. Para mitigar essa limitação, foram desenvolvidas variantes, como as Long Short-Term Memory (LSTM), que são mais adequadas para capturar dependências de longo prazo nos dados.

O LSTM tem se mostrado altamente eficaz na previsão de fenômenos que envolvem dados temporais, como o clima, que pode capturar padrões temporais complexos, o que o torna ideal para a previsão de desastres naturais, como tempestades e enchentes (Zhang et al., 2019). Sua capacidade de aprender com sequências temporais e prever eventos futuros o torna uma ferramenta promissora para a previsão de desastres naturais com alta precisão.

Este modelo além de receber como entradas os dados de exemplo para aprendizado é capaz também de se retroalimentar com dados passados para predição, como representado na Figura 3:

Figura 3 - Representação da RNN



Fonte: <https://www.deeplearningbook.com.br/redes-neurais-recorrentes>. Acesso em 03 de outubro de

Cada célula LSTM pode ser representada por três portas principais (*gates*), porta de entrada, porta de esquecimento e porta de saída. A lógica destas portas é ativada por funções que determinam as proporções de informação que se deve ser adicionada, mantida ou esquecida a cada interação (Goodfellow; Bengio; Courville, 2016).

O funcionamento das portas pode ser descrito pelas seguintes equações:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

A equação 1 representa a porta do esquecimento, essa porta decide quais informações da célula anterior c_{t-1} devem ser descartadas. Valores f_t próximos de 0 representam esquecimentos, enquanto valores próximos a 1 representam retenção de informação.

A porta de entrada representa quais informações serão armazenadas na célula:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$\bar{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (3)$$

Onde o vetor $\bar{C}_t$ representa os novos candidatos de memórias de longo prazo.

Durante a atualização do estado da célula, a equação 4 combina as informações que serão adicionadas com as que serão esquecidas, resultando no novo estado da célula, responsável por armazenar informação ao longo do tempo.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \bar{C}_t \quad (4)$$

Por fim, a porta de saída define quais informações do estado da célula serão transmitidas como saída para o próximo instante temporal.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (6)$$

O vetor h_t representa o estado oculto da rede no tempo t .

Essa arquitetura em múltiplas portas confere às LSTMs a capacidade de armazenar dependências de longo prazo, filtrar ruídos e manter informações relevantes mesmo em séries temporais extensas (Chollet, 2021).

2.5 CLUSTERING E OTIMIZAÇÃO LOGÍSTICA

O clustering (ou agrupamento) é uma técnica de aprendizado não supervisionado que visa agrupar dados semelhantes em conjuntos, ou clusters, com base em suas características. Métodos como K-Means e DBSCAN são amplamente utilizados para realizar essa tarefa (Jain, 2010). No contexto da distribuição de ajuda humanitária, o clustering pode ser empregado para otimizar a localização dos centros de distribuição de itens como alimentos, água e medicamentos, a fim de garantir uma resposta rápida e eficiente.

A otimização logística busca minimizar custos e maximizar a eficiência no processo de distribuição. A aplicação de técnicas de clustering para alocar centros de distribuição permite a identificação de locais estratégicos que atendem a uma maior quantidade de pessoas afetadas, ao mesmo tempo que minimizam o tempo e os custos envolvidos na entrega de recursos (Balci & Beyhan, 2021). Estudos indicam que o uso de modelos de clustering dinâmico, combinados com algoritmos genéticos, pode otimizar a distribuição de recursos de forma ainda mais eficaz (Oliveira et al., 2023).

A otimização logística e o uso de clustering desempenham um papel crucial na melhoria da eficiência das operações de resposta humanitária em desastres naturais. Ambos os conceitos visam aumentar a eficácia no gerenciamento de recursos e reduzir os custos associados à distribuição de suprimentos. O clustering é uma técnica de agrupamento de dados que organiza elementos semelhantes em grupos ou clusters, enquanto a otimização logística envolve a aplicação de métodos matemáticos para maximizar a eficiência no transporte, alocação e distribuição de recursos. A combinação dessas abordagens pode melhorar substancialmente a eficácia das operações de ajuda em áreas afetadas por desastres naturais (Balci & Beyhan, 2021).

Em um cenário de desastre, as áreas de maior necessidade podem ser determinadas com base em dados como precipitação, intensidade do vento, e número de vítimas. Ao realizar o agrupamento dessas regiões, as equipes de resgate podem determinar as zonas prioritárias de intervenção e otimizar as rotas de distribuição de recursos, como alimentos, água potável, medicamentos e materiais de resgate. O uso de K-Means, por exemplo, pode agrupar áreas afetadas de acordo com características como proximidade de vias de acesso, quantidade de danos em infraestrutura e

densidade populacional, ajudando a alocar centros de distribuição de maneira mais estratégica (Jain, 2010).

Além disso, o DBSCAN tem sido cada vez mais utilizado em situações de emergência, pois é capaz de identificar áreas de alta densidade (como zonas críticas) e áreas de baixa densidade (como regiões menos afetadas), mesmo quando os dados contêm ruído (Zhu et al., 2019). A identificação de zonas de alta densidade é particularmente útil para otimizar a distribuição de recursos limitados, direcionando-os para onde são mais necessários.

Estudos demonstram que a integração de clustering e otimização logística pode reduzir significativamente os tempos de resposta e aumentar a cobertura das áreas afetadas (Balci & Beyhan, 2021). Uma aplicação prática desse modelo seria a criação de uma rede de centros de distribuição que são estrategicamente localizados nas áreas identificadas pelo clustering, com rotas de transporte otimizadas para minimizar custos e tempo de entrega.

O uso de algoritmos de aprendizado de máquina para prever áreas de risco, identificar padrões de demanda e otimizar a alocação de recursos em tempo real está se tornando uma área promissora. Modelos de aprendizado profundo, como as redes neurais convolucionais (CNN) e redes neurais recorrentes (RNN), podem ser empregados para processar grandes volumes de dados espaciais e temporais e, assim, fornecer previsões mais precisas e respostas logísticas mais rápidas (LI et al., 2021).

2.5.1 K-MEANS: Um método de Clustering para Logística Humanitária

O K-Means é um dos algoritmos de clustering mais populares e amplamente utilizados em diversas áreas, incluindo a logística humanitária. Este método visa particionar um conjunto de dados em K grupos, ou clusters, de forma que os dados dentro de cada grupo sejam mais semelhantes entre si do que com os dados de outros clusters. Ele é particularmente útil em situações de desastres naturais, onde a necessidade de identificar zonas de alta demanda e otimizar a distribuição de recursos é urgente (Zhang *et al.*, 2018).

Jain (2010) cita que o algoritmo K-Means segue uma abordagem iterativa para minimizar a soma das distâncias quadradas entre os pontos de dados e o centro de seus respectivos clusters. O processo é composto por três etapas principais:

- I. Escolha dos centroides iniciais: Inicialmente, o número de clusters (K) é definido e K pontos aleatórios são escolhidos como os centroides dos clusters. Os centroides representam as médias dos pontos dentro de um cluster;
- II. Atribuição dos pontos aos clusters: Cada ponto de dados é então atribuído ao cluster cujo centroide está mais próximo. Isso é feito com base na distância euclidiana entre os pontos e os centroides;
- III. Atualização dos centroides: Após a atribuição dos pontos, os centroides dos clusters são recalculados, tomando a média de todos os pontos atribuídos a cada cluster. O algoritmo então repete os passos de atribuição e atualização até que os centroides se estabilizem e não mudem significativamente;

O algoritmo da iteração realizada pelo método *k-means* pode ser representada de acordo com a equação 7:

$$J = \sum_{j=1}^n \sum_{k=1}^K \mu_{kj} D^2(z_k, x_j) \quad (7)$$

onde $D^2(z_k, x_j)$ representa a distância euclidiana de x_j para z_k e $z_1, z_2, z_3, \dots, z_k$ são os centroides formados pelos clusters.

Uma das principais vantagens do *k-means* é sua simplicidade e rapidez, o que o torna adequado para análise em tempo real em cenários de desastre. Além disso, o K-Means é capaz de lidar com grandes volumes de dados e é relativamente fácil de implementar, o que o torna uma escolha popular para a logística humanitária (Davidson *et al.*, 2017).

2.6 MÉTRICAS DE AVALIAÇÃO

A classificação é uma empreitada complexa e multifacetada, conforme destacado por Géron (2017), sendo frequentemente mais desafiadora do que avaliar um regresso. Diversas ferramentas estão disponíveis para testar a qualidade de um modelo de classificação, com métricas específicas que oferecem insights sobre seu

desempenho. Algumas dessas métricas incluem R-Quadrado, RMSE (Raiz do Erro Quadrático Médio), MAE (Erro Absoluto Médio) e MAPE (Erro Percentual Absoluto Médio).

2.6.1 RMSE – Raiz do Erro Quadrático Médio

A raiz do erro quadrático médio (RMSE), ou *Root Mean Square Error* é o erro quadrático entre as observações reais e a previsão no conjunto dos dados.

Conforme mencionado por Chai e Draxler (2014), a escolha dessa métrica é mais apropriada quando a distribuição dos erros do modelo se assemelha a uma distribuição gaussiana. Isso é particularmente útil para evitar que tendências de erro, ocasionadas por *outliers*, influenciem excessivamente a soma total dos erros

O RMSE é calculado da seguinte forma:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (8)$$

onde y_i representa o valor observado, $\hat{y}_i$ o valor previsto e n o número total de observações.

Valores de RMSE próximos de zero indicam maior precisão e melhor ajuste do modelo aos dados reais, enquanto valores mais altos sugerem maior dispersão dos erros e menor capacidade de generalização (Hyndman; Koehler, 2006). De acordo com Brownlee (2020), o RMSE é particularmente útil quando se deseja identificar a sensibilidade do modelo a grandes desvios

2.6.2 MAE – Erro Médio Absoluto

O Erro Médio Absoluto (MAE), ou Mean Absolute Error em inglês, é uma métrica que calcula a média da diferença absoluta entre os valores reais e os valores previstos. Essa métrica considera o valor absoluto das diferenças, levando em conta tanto valores positivos quanto negativos. Conforme explicado por Hastie *et al.* (2009), o MAE é menos sensível a outliers nos dados em comparação com o Erro Quadrático Médio (MSE) e pode ser mais apropriado para problemas nos quais erros maiores não são tão críticos quanto erros menores.

A sensibilidade reduzida a outliers do MAE é uma característica valiosa em situações em que observações extremas podem distorcer significativamente as métricas de desempenho. Essa métrica proporciona uma visão mais equilibrada da precisão do modelo, considerando a magnitude absoluta dos erros, o que é particularmente útil quando erros maiores não têm um impacto desproporcionalmente maior do que erros menores.

Além disso, ao adotar o MAE, os resultados podem ser interpretados de maneira mais intuitiva, pois representa a média das diferenças absolutas entre as previsões e os valores reais. Isso pode facilitar a compreensão do impacto dos erros do modelo e fornecer uma medida mais realista da qualidade de suas previsões, especialmente em situações em que é fundamental minimizar desvios absolutos.

A equação 9 apresenta o cálculo do MAE:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (9)$$

em que y_i é o valor real, $\hat{y}_i$ é o valor previsto e n o número total de observações.

Segundo Gerón (2023), valores baixos de MAE indicam previsões mais próximas dos valores reais, sendo possível interpretar o resultado em termos percentuais quando os dados estão normalizados.

2.7 CONSIDERAÇÕES FINAIS DO CAPÍTULO.

Neste capítulo, foram discutidos os principais conceitos relacionados à previsão de desastres naturais e à logística humanitária, com destaque para o uso de redes neurais, como LSTM, e técnicas de clustering. A previsão precisa de eventos climáticos extremos, por meio de redes neurais, possibilita uma antecipação mais eficaz das necessidades de intervenção e otimização das operações de resposta. Além disso, o uso de clustering, como o método K-Means, auxilia no agrupamento de áreas afetadas por características geográficas e infraestruturais, facilitando a identificação de zonas prioritárias e a alocação estratégica de recursos essenciais.

A otimização logística, combinada com o clustering, oferece uma solução eficiente para a gestão de centros de distribuição e rotas de transporte, reduzindo o tempo de resposta e os custos envolvidos. A integração dessas abordagens tecnológicas pode transformar a logística humanitária em desastres naturais,

garantindo uma resposta mais ágil, eficiente e precisa, o que resulta em uma ajuda mais eficaz para as populações impactadas. Assim, o uso dessas ferramentas contribui significativamente para a melhoria da capacidade de resposta em situações de emergência.

3 METODOLOGIA

A relevância da metodologia em um trabalho científico está associada à necessidade de um suporte teórico consistente, o qual envolve a escolha da abordagem de pesquisa mais apropriada para responder aos objetivos propostos, assim como a definição dos métodos e técnicas mais adequados para orientar seu desenvolvimento e execução (Miguel, 2007).

Este trabalho adota uma abordagem quantitativa, estruturada como um estudo de caso aplicado ao estado de Santa Catarina, com ênfase em modelagem preditiva (LSTM) e análise de clusterização para suporte à logística humanitária.

3.1 MATERIAL E MÉTODOS

O estudo terá como fonte de informações o banco de dados da Secretaria Nacional de Defesa Civil de Santa Catarina (SEDEC/SC), de onde foi obtido informações dos eventos entre o período de 2013 até 2023.

Os dados obtidos foram solicitados através do canal de comunicação disponível no portal e recebidos em formatos de planilha contendo as seguintes informações: Município, Microrregião, evento ocorrido, item de assistência humanitária disponibilizado assim como a quantidade e a data de ocorrência.

Os dados disponibilizados correspondem a 447 eventos relatados entre junho de 2013 até março de 2023.

Para manipulação, análise e modelagem dos dados obtidos através das fontes citadas anteriormente será utilizado a linguagem de programação PYTHON, comumente utilizada para o desenvolvimento de trabalhos computacionais e gráficos.

3.1.1 Abordagem Quantitativa

Esta pesquisa adota uma abordagem quantitativa, na mensuração objetiva de variáveis, na formulação e no teste dedutivo de hipóteses e no uso de técnicas estatísticas para garantir validade, confiabilidade e replicabilidade dos resultados (Lakatos; Marconi, 2010).

3.1.2 Estudo de Caso

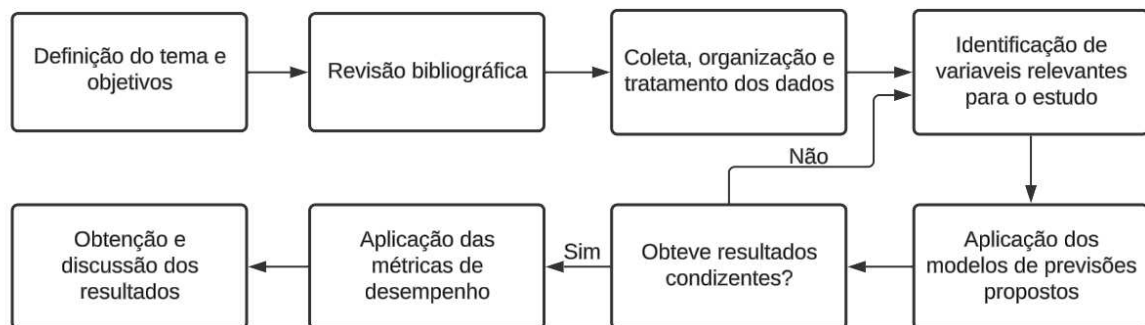
O estudo de caso é uma estratégia de pesquisa qualitativa que permite uma análise profunda e contextualizada de um fenômeno em particular (Yin, 2015). No presente trabalho, o estudo de caso envolve a análise de desastres de Santa Catarina.

Segundo Gil (2019), o estudo de caso é particularmente útil quando se pretende realizar uma investigação detalhada de um fenômeno que não pode ser totalmente compreendido fora de seu contexto.

3.2 PROCEDIMENTOS METODOLOGICOS

As etapas metodológicas para elaboração deste trabalho formam desenvolvidas de acordo com o fluxograma da Figura 4.

Figura 4 - Etapas metodológicas



Fonte: Autor

O trabalho inicia-se com a definição clara do tema e objetivos, estabelecendo o escopo da pesquisa. É importante definir claramente o escopo do trabalho, identificando qual é o problema a ser investigado e estabelecendo os objetivos do estudo.

A revisão de literatura consiste em pesquisar e reunir informações relevantes sobre o tema, como métodos semelhantes já aplicados, estudos que tratam de previsões utilizando redes neurais e análises de desastres. Essa fase fornece a base teórica para o desenvolvimento da pesquisa.

A etapa seguinte envolve a coleta, revisão e tratamento de dados, onde são adquiridos os dados relevantes para o estudo, sendo avaliada sua qualidade e consistência. Nesse processo, são identificadas as variáveis relevantes para a pesquisa.

A aplicação dos modelos de previsões é então realizada, utilizando técnicas adequadas ao objetivo do estudo e observado os resultados obtidos e analisado se tais resultados condizem com o esperado.

Após a aplicação dos modelos, verifica-se se os resultados obtidos estão dentro das expectativas e são condizentes com a realidade observada. Caso não sejam, é necessário retornar às etapas anteriores para ajustar o modelo ou revisar os dados.

Posteriormente, as métricas de avaliação são aplicadas para mensurar a eficácia dos modelos em relação aos dados observados.

Os resultados obtidos são, então, discutidos em profundidade, contextualizando-os com a revisão bibliográfica e as variáveis identificadas. Essa metodologia abrangente visa garantir uma abordagem sistemática, desde a formulação dos objetivos até a interpretação e discussão dos resultados, contribuindo para a validade e robustez da pesquisa.

3.3 DELIMITAÇÕES

O trabalho aborda todos os dados fornecidos do ano de 2013 até 2023. Contemplando apenas as informações quantitativas em relação as catástrofes ocorridas.

O escopo deste trabalho abrange somente uma análise dos desastres ocorridos no estado de Santa Catarina.

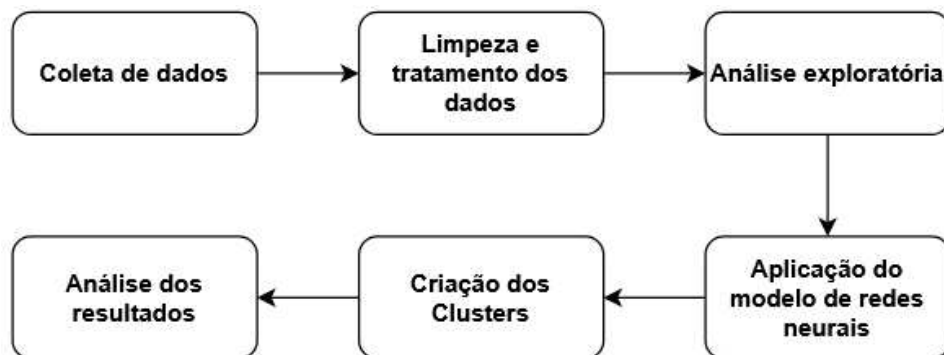
O presente trabalho consiste em aplicar o modelo de rede neurais do tipo Long Short-Term Memory (LSTM) sobre os indicadores de desastres, realizar avaliações através de métricas, entretanto não tem como objetivo aferir modificações e tomadas de decisões baseadas nos resultados obtidos.

Esta pesquisa depende da qualidade e disponibilidade dos dados meteorológicos e de desastre. Limitações como registros incompletos, baixa resolução temporal ou inconsistências em fontes públicas podem influenciar os resultados.

4 ESTUDO DE CASO

Este capítulo especifica as etapas para o desenvolvimento deste estudo, descrevendo os passos para análise e aplicação do modelo de previsão proposto em conjunto com o método de clusterização. As etapas abordadas são representadas na figura a seguir:

Figura 5 - Fluxo do projeto



Fonte: Autor

4.1 COLETA E TRATAMENTO DE DADOS

Para a etapa de coleta de dados foram acessados os acervos da Secretaria Nacional de Proteção e Defesa Civil (SEDEC), de onde foram obtidas as informações referentes ao período entre 2013 e 2023 dos eventos para treino do modelo de previsão, do estado de Santa Catarina.

Os dados obtidos são fornecidos com informações referentes ao município, microrregião e tipo de desastre ocorrido, além da data de ocorrência. Outros dados também são disponibilizados como material de apoio para os afetados, bem como quantidade e custo dos materiais.

Para o treinamento do modelo de redes neurais foram filtrados apenas as colunas necessárias para treinamento, sendo elas os municípios, os eventos ocorridos e a data de ocorrência, em seguida efetuado uma verificação dos dados obtidos.

O pré-processamento de dados é o processo de avaliar, filtrar, transformar e codificar dados para que um algoritmo de aprendizado de máquina possa compreendê-los. O principal objetivo do pré-processamento de dados é eliminar problemas como valores ausentes e melhorar a qualidade dos dados (Lakefs, 2023).

A determinação adequada dos dados tem alto impacto na performance de modelos de aprendizagem de máquina (Witten; Frank; Hall, 2016)

Para utilização destes dados, inicialmente foi realizado uma limpeza removendo linhas com valores nulos para não prejudicar o treinamento, em sequência adequação das unidades e mapeamento dos eventos para unidades quantitativas, através do script ilustrado na figura 6:

Figura 6 - Mapeamento e limpeza dos dados

```
DATE_COL = "Data"
TARGET_COL = "Evento"
FEATURE_COLS = ["municipio"]

df[DATE_COL] = pd.to_datetime(df[DATE_COL], errors="coerce")
df = df.dropna(subset=[DATE_COL]).sort_values(DATE_COL)

df[TARGET_COL] = df[TARGET_COL].astype(str).str.strip()
labels = list(pd.unique(df[TARGET_COL]))
label_to_id = {lbl: i for i, lbl in enumerate(labels)}
id_to_label = {i: lbl for lbl, i in label_to_id.items()}
df["Evento_id"] = df[TARGET_COL].map(label_to_id).astype("int32")

print("Mapeamento (id -> label):")
for i in range(len(labels)):
    print(f" {i} -> {id_to_label[i]}")

df.head(10)
```

Fonte: Autor

A figura 7 ilustra o mapeamento de eventos:

Figura 7 - Mapeamento dos eventos

	município	Evento	Data	Evento_id
0	Três Barras	Enxurrada	2013-06-26	0
1	Mafra	Enxurrada	2013-06-26	0
2	Porto União	Enxurrada	2013-06-26	0
3	Três Barras	Enxurrada	2013-07-01	0
4	Canoinhas	Inundação	2013-07-01	1
5	Irineópolis	Inundação	2013-07-01	1
6	Porto União	Enxurrada	2013-07-01	0
7	São Joaquim	Inundação	2013-07-24	1
8	Ararangua	Inundação	2013-08-26	1
9	Caçador	Alagamento	2013-09-22	2

Fonte: Autor

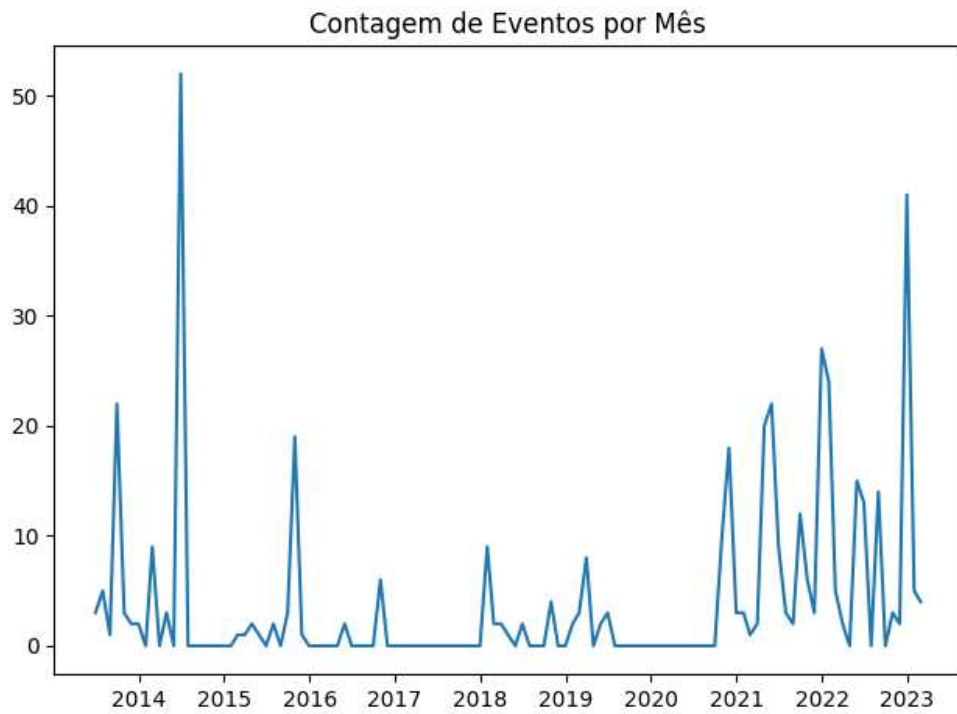
4.2 ANÁLISE EXPLORATÓRIA

Através de uma análise exploratória é possível visualizar e identificar as características dos dados utilizados e examinar as variáveis relevantes (McKinney, 2017).

Gráficos de séries temporais são utilizados para identificar padrões sazonais, tendências e possíveis anomalias que ocorrem nos dados ao longo do tempo. A aplicação conjunta dessas técnicas possibilita uma compreensão mais aprofundada da dinâmica temporal dos fenômenos analisados, contribuindo para a extração de insights relevantes sobre o comportamento dos dados (Chatfield, 2004).

A Figura 8 apresenta a quantidade eventos mensais de 2013 a 2023:

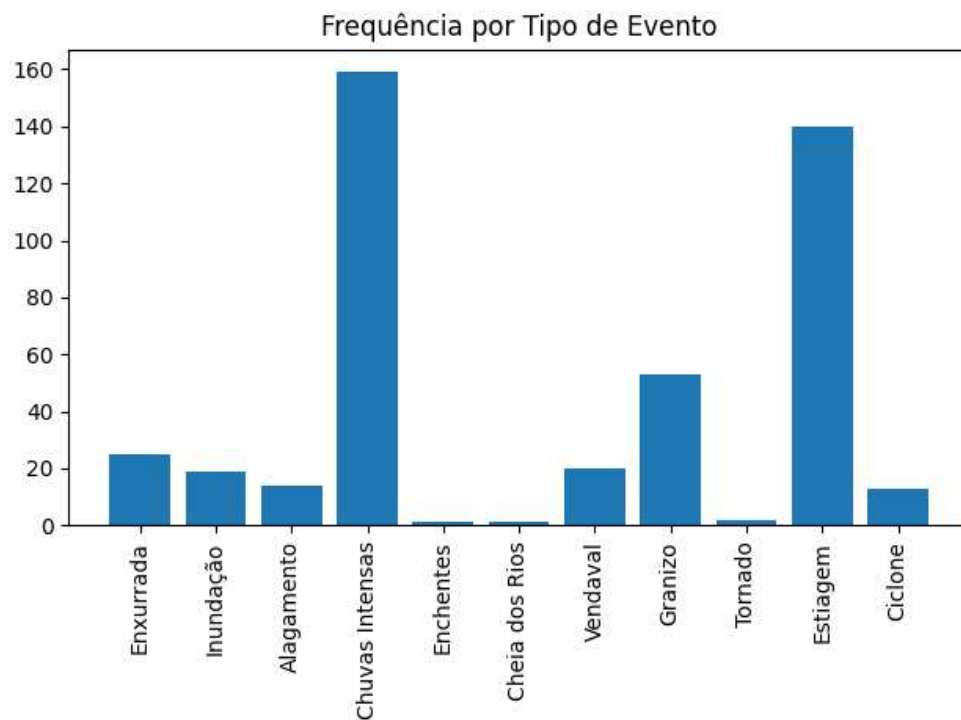
Figura 8 - Eventos mensais



Fonte: Autor

Chuvas intensas foram os eventos mais expressivos no período analisado neste estudo, como apresentado pela figura 9:

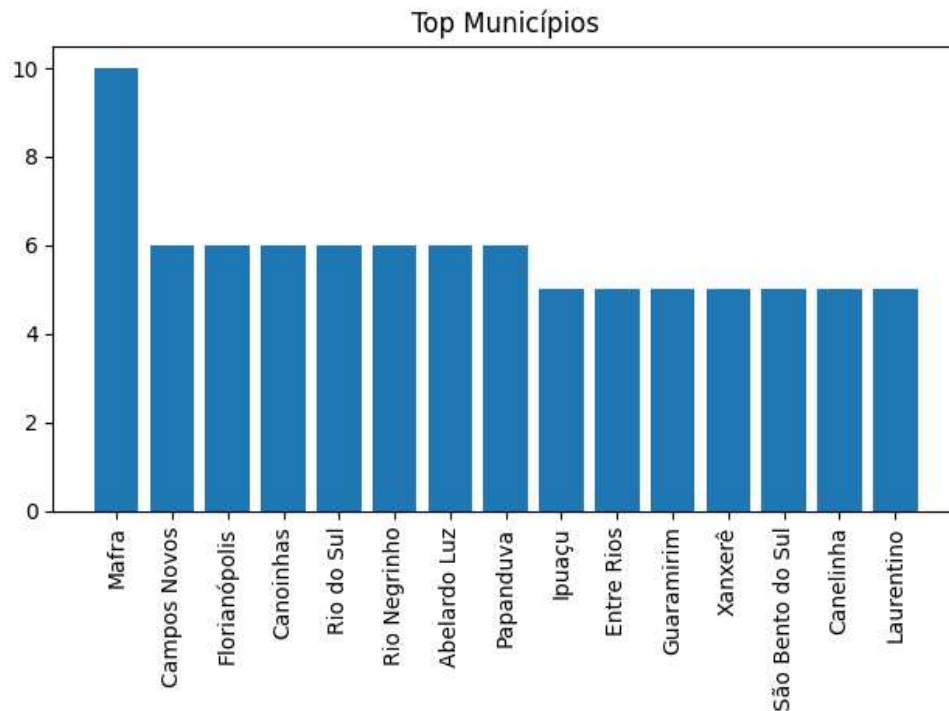
Figura 9 - Frequência de eventos



Fonte: Autor

Segundo a análise da figura 10, representando os municípios com mais eventos extremos, a cidade de Mafra apresenta destaque:

Figura 10 - Quantidade de eventos por município



Fonte: Autor

4.3 APLICAÇÃO DE REDES NEURAIS

4.3.1 Features e Normalização

Segundo Kamber e Pei (2022) a escolha adequada das variáveis influencia na capacidade preditiva da rede neural, pois cada variável contribui informações específicas sobre o comportamento do fenômeno analisado.

Para este modelo foram escolhidos os eventos e municípios como features, ilustrados na figura a seguir.

Figura 11 – Features

```
base_cols = feature_cols + [args.target_col]
data = df[base_cols].copy()

non_numeric_feats = [c for c in feature_cols if not is_numeric_dtype(data[c])]
if non_numeric_feats:
    print("[INFO] Aplicando one-hot em features não numéricas:", non_numeric_feats)
    data = pd.get_dummies(data, columns=non_numeric_feats, drop_first=False)
```

Fonte: Autor

Após a escolha das variáveis, foi aplicado o processo de normalização dos dados. A normalização ajusta os valores para uma mesma escala, evitando que variáveis com magnitudes diferentes exerçam influência desproporcional sobre o treinamento (Géron, 2023). Nesse modelo, utilizou-se a técnica de Min-Max Scaling, que transforma os dados para o intervalo entre 0 e 1. A figura 12 apresenta sua normalização e escala.

Figura 12 - Normalização e escala

```
# Escalonamento (MinMax 0-1) de todas as colunas
scalers = {}
for c in data.columns:
    s = MinMaxScaler()
    data[c] = s.fit_transform(data[[c]])
    scalers[c] = s
```

Fonte: Autor

4.3.2 Janelamento Temporal

O janelamento temporal é uma técnica utilizada para transformar séries temporais em conjuntos de dados supervisionados. Essa abordagem cria janelas deslizantes que permitem ao modelo aprender padrões de dependência temporal entre observações passadas e valores futuros (Brownlee, 2020).

O parâmetro *Lookback* são as observações históricas do modelo e *horizon* é o evento subsequente previsto, descrito na figura a seguir.

Figura 13 - Janelamento temporal

```
# Construção das janelas
features_used = [c for c in data.columns if c != args.target_col]
X_all = data[features_used].values if features_used else data[[args.target_col]].values
y_all = data[args.target_col].values

X_seq, y_seq = make_sequences(X_all, y_all, args.lookback, args.horizon)
```

Fonte: Autor

4.3.3 Split Temporal

A divisão dos dados em conjuntos de treinamento e teste é um método padrão na modelagem preditiva. No entanto, em problemas temporais, essa divisão deve respeitar a ordem cronológica das observações, evitando que o modelo seja treinado com informações futuras, esse método também evita o problema de *data leakage*, situação em que o modelo tem acesso a informações do futuro, o que poderia inflar artificialmente sua acurácia (Géron, 2023).

Nesta etapa foi alocado 70% dos dados para treinamento do modelo, 15% para teste e 15% para validação.

Figura 14 - Split temporal

```
p.add_argument("--val-size", type=float, default=0.15, help="Proporção para validação (final da série)")
p.add_argument("--test-size", type=float, default=0.15, help="Proporção para teste (final da série)")

# Split temporal
n = len(X_seq)
n_test = int(n * args.test_size)
n_val = int(n * args.val_size)
n_train = n - n_val - n_test
if n_train <= 0 or n_val <= 0 or n_test <= 0:
    print("[ERRO] Partição inválida. Ajuste val-size/test-size.", file=sys.stderr)
    sys.exit(1)

X_train, y_train = X_seq[:n_train], y_seq[:n_train]
X_val, y_val = X_seq[n_train:n_train+n_val], y_seq[n_train:n_train+n_val]
X_test, y_test = X_seq[n_train+n_val:], y_seq[n_train+n_val:]
```

Fonte: Autor

4.3.4 Modelo LSTM

As LSTMs utilizam unidades de memória compostas por portas de entrada, esquecimento e saída, que controlam o fluxo de informações relevantes e descartam padrões temporais irrelevantes. Essa estrutura permite à rede armazenar e reter

informações importantes ao longo de longos períodos, tornando-a adequada para análise de séries temporais e previsão de eventos complexos (Goodfellow; Bengio; Courville, 2016).

Para este modelo a rede foi composta por camadas LSTM empilhadas, seguidas por camadas densas para a saída preditiva, ilustrado na figura 15.

Figura 15 - Modelo LSTM

```
# ----- Modelo -----
model = models.Sequential()
# Entrada explícita baseada no shape do treino
model.add(layers.Input(shape=X_train.shape[1:]))
for i in range(args.lstm_layers):
    return_sequences = (i < args.lstm_layers - 1)
    model.add(layers.LSTM(args.lstm_units, return_sequences=return_sequences))
    model.add(layers.Dropout(args.dropout))
model.add(layers.Dense(1))

opt = tf.keras.optimizers.Adam(learning_rate=args.lr)
model.compile(optimizer=opt, loss="mse", metrics=[tf.keras.metrics.MeanAbsoluteError(name="mae")])

es = callbacks.EarlyStopping(monitor="val_loss", patience=15, restore_best_weights=True)
rlrop = callbacks.ReduceLROnPlateau(monitor="val_loss", factor=0.5, patience=5)

hist = model.fit(
    X_train, y_train,
    validation_data=(X_val, y_val),
    epochs=args.epochs,
    batch_size=args.batch_size,
    callbacks=[es, rlrop],
    verbose=1
)
```

Fonte: Autor

O código completo para elaboração da modelagem obtenção dos gráficos acima encontra-se no apêndice A.

4.4 CRIAÇÃO DOS CLUSTERS

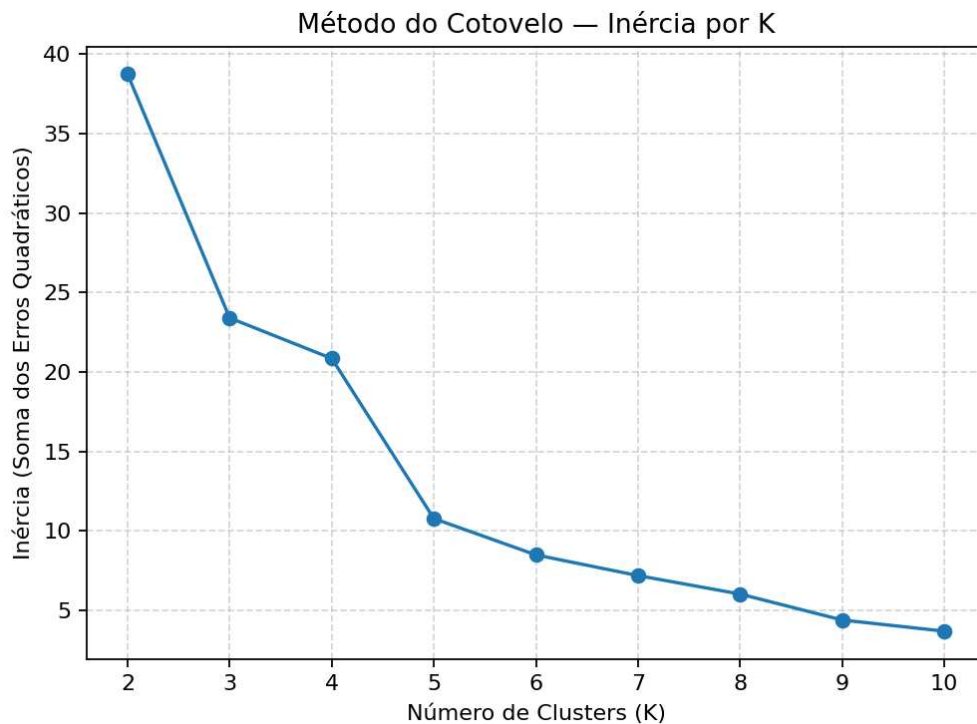
Para a criação dos clusters foi utilizado o método *k-means* através dos dados obtidos pelo modelo de previsão de redes neurais, devido a sua robustez, eficiência, e ampla utilização em estudos similares na literatura para modelos geográficos (Lima *et al*, 2014).

A determinação do número ótimo de cluster é importante pois influencia diretamente na interpretação dos resultados obtidos (Han; Kamber; Pei, 2022). Para

a escolha do valor ideal de K , utilizou-se o método do cotovelo, uma técnica amplamente empregada em análises de clusterização (Kodinarias; Gad, 2013).

Segundo observado pela figura 16, o ponto de inflexão, ou valor ótimo de k foi de valor 4:

Figura 16 - Escolha do k ótimo



Fonte: Autor

O código para obtenção dos clusters e determinação do valor ótimo de k está presente no apêndice B.

4.5 RESULTADOS E DISCUSSÕES

Para execução do modelo foram utilizados os seguintes argumentos padrões, de acordo com a figura 17:

Figura 17 - Argumentos para previsão

```

p.add_argument("--file", required=True, help="Caminho para .xlsx ou .csv")
p.add_argument("--sheet", type=int, default=0, help="Índice da planilha (se Excel)")
p.add_argument("--date-col", required=True, help="Nome da coluna de data")
p.add_argument("--target-col", required=True, help="Nome da coluna alvo (ex.: Evento, ocorrências)")
p.add_argument("--feature-cols", default="", help="Colunas de features separadas por vírgula (opcional)")
p.add_argument("--freq", default=None, help="Reamostrar para frequência (ex.: D, W, M) – opcional")
p.add_argument("--fill", default="ffill", choices=["ffill", "bfill", "zero", "mean"], help="Preenchimento de faltantes")
p.add_argument("--lookback", type=int, default=30, help="Passos históricos")
p.add_argument("--horizon", type=int, default=1, help="Passos à frente para prever")
p.add_argument("--val-size", type=float, default=0.15, help="Proporção para validação (final da série)")
p.add_argument("--test-size", type=float, default=0.15, help="Proporção para teste (final da série)")
p.add_argument("--epochs", type=int, default=50)
p.add_argument("--batch-size", type=int, default=32)
p.add_argument("--lstm-units", type=int, default=64)
p.add_argument("--lstm-layers", type=int, default=1)
p.add_argument("--dropout", type=float, default=0.2)
p.add_argument("--lr", type=float, default=1e-3)
p.add_argument("--model-out", default="modelo_lstm.keras")
p.add_argument("--plot-out", default="treino_performance.png")
args = p.parse_args()

```

Fonte: Autor

Os resultados obtidos pelo modelo de redes neurais através do método LSTM foram elaborados alterando os valores padrões de lookback, horizon e epoch e verificando os resultados da predição, desta maneira os seguintes parâmetros para execução do código forma os que obtiveram melhores resultados:

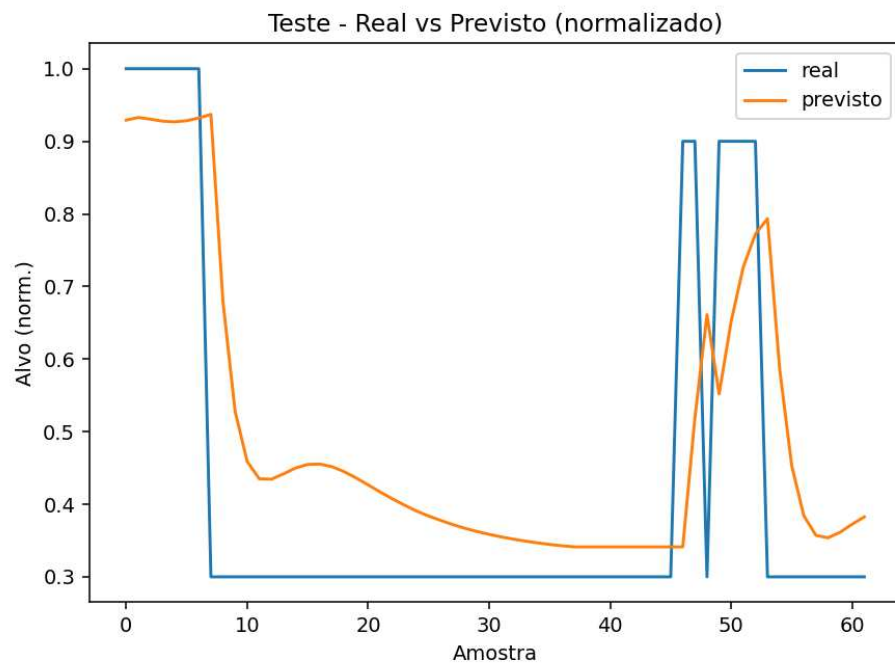
Tabela 1 - Parâmetros para execução

Parâmetros	Valores
date-col	Data
target-col	Evento
feature-cols	Município
Lookback	30
Horizon	1
Epoch	50

Fonte: Autor

Dada os parâmetros inseridos, o modelo apresentou os resultados reais em comparação aos previstos, de acordo com a figura 18.

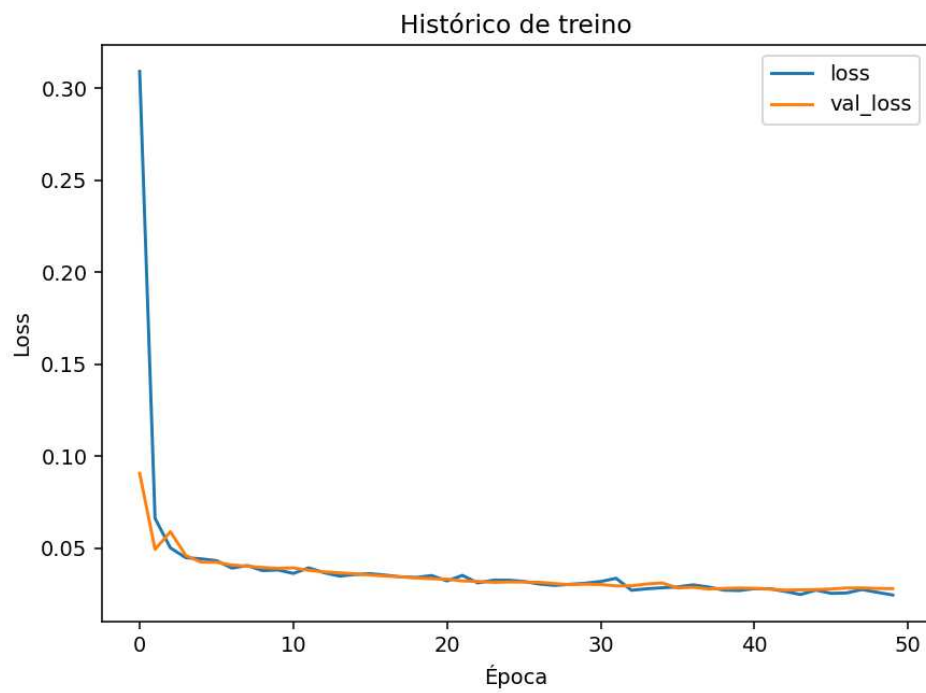
Figura 18 - Teste de previsão



Fonte: Autor

E o histórico de treino ilustrado na figura 19.

Figura 19 - Histórico de treino



Fonte: Autor

Analisando os resultados encontrados é possível observar que o gráfico de teste de previsões em comparação com os valores reais se aproximou de maneira similar.

Da mesma maneira, no gráfico de histórico de treino, o valor da perda é reduzido tomando como valor 50 para o parâmetro Época.

Outro resultado que corrobora o resultado do modelo são as métricas de avaliação da Tabela 2:

Tabela 2 - Métricas de avaliação

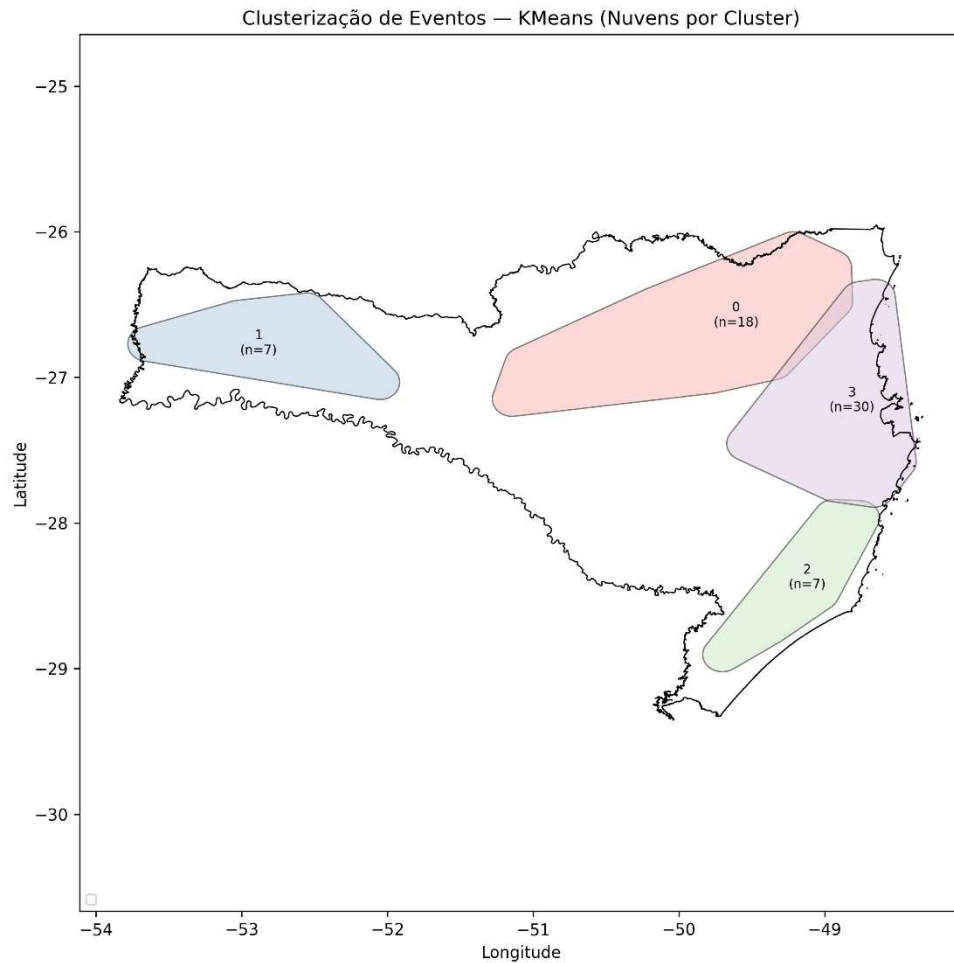
Métricas	Valor
MAE (normalizado)	0.1349
RMSE (normalizado)	0.1868
MAE (original)	1.3495
RMSE (original)	1.8686

Fonte: Autor

Onde o MAE normalizado apresentou erro absoluto médio de 0,13 ou 13%, considerado precisão alta, assim como o resultado da métrica RMSE normalizado, com valores de 0,19 ou 19% considerado moderado.

O resultado obtido através do modelo de clusterização pelo método *k-means* apresenta a formação de 4 clusters, como visto na figura 20, de acordo com a escolha embasada pelo método do cotovelo.

Figura 20 - Clusterização dos eventos



Fonte: Autor

A distribuição dos eventos para o cluster 0 a apresentou maiores incidências de chuvas intensas e estiagem, como visto na tabela 3.

Tabela 3 - Distribuição cluster 0

Cluster 0	
Evento	Média
Chuvas Intensas	0,4444444
Estiagem	0,4444444
Granizo	0,1111111

Fonte: Autor

No cluster 1, o evento de chuvas intensas obteve a maior proporção, de acordo com a tabela 4.

Tabela 4 - Distribuição cluster 1

Cluster 1	
Evento	Média
Chuvas Intensas	0,7142857
Estiagem	0,2857143

Fonte: Autor

Na distribuição do cluster 2, chuvas intensas obteve maior proporção, como visto na tabela 5.

Tabela 5 - Distribuição cluster 2

Cluster 2	
Evento	Média
Chuvas Intensas	0,4285714
Estiagem	0,2857143
Ganizo	0,2857143

Fonte: Autor

Na tabela 6, o cluster 3, chuvas intensas apresentou maior proporção, dentre os eventos como estiagem e granizo.

Tabela 6 - Distribuição cluster 3

Cluster 3	
Evento	Média
Chuvas Intensas	0,5
Estiagem	0,3666667
Granizo	0,1333333

Fonte: Autor

Observa-se que os modelos resultaram em clusters onde o evento chuvas intensas predominam, com exceção do cluster 0, onde chuvas intensas e estiagem são predominantes em sua maioria. É possível observar também que no cluster 1 houve uma aglomeração maior de eventos.

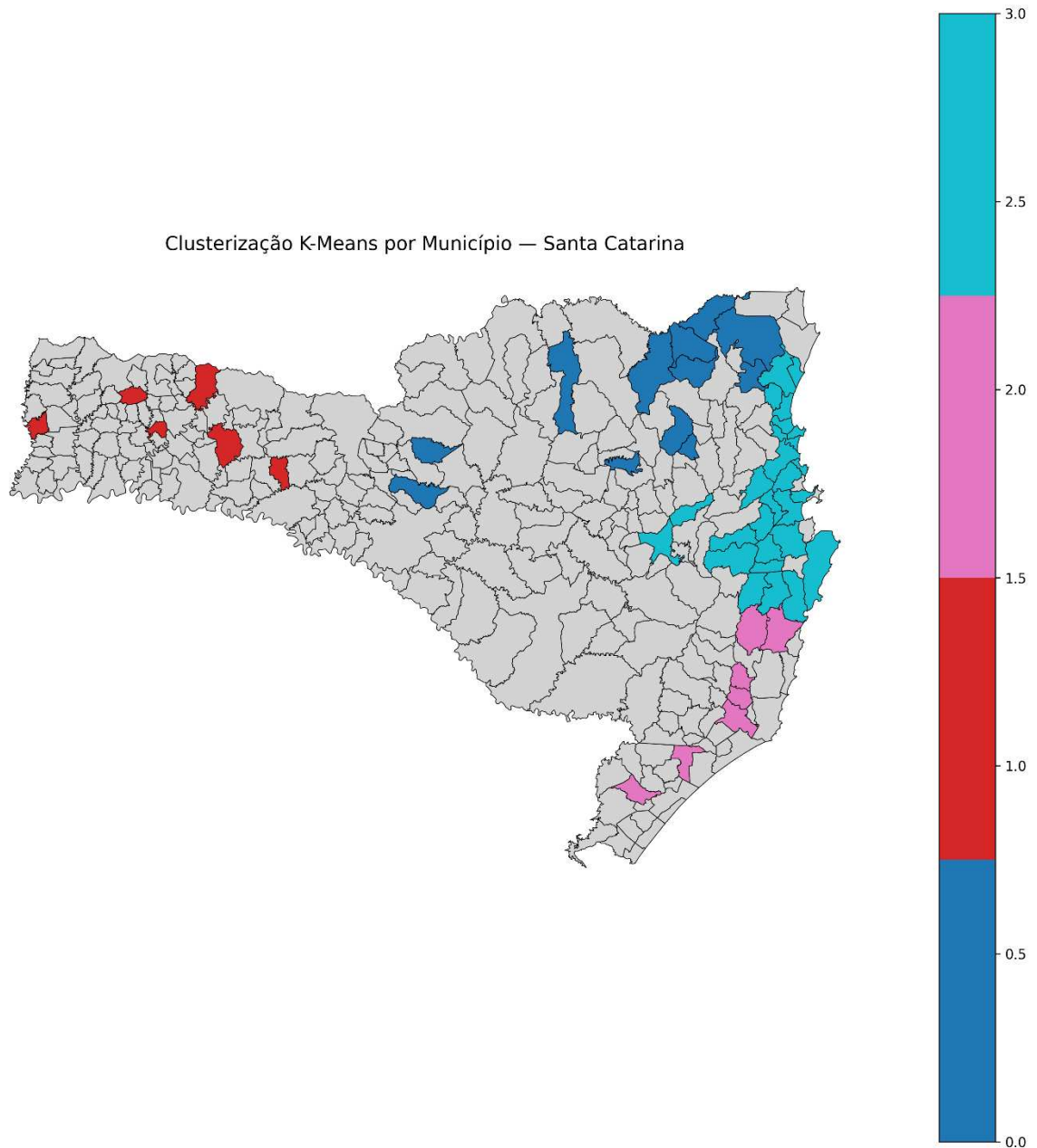
A aplicação do algoritmo K-Means aos municípios de Santa Catarina, utilizando como base os resultados gerados pelo modelo de previsão, possibilitou a formação dos quatro clusters distintos.

Para visualizar essa segmentação territorial, foi realizado o georreferenciamento dos municípios, a partir da obtenção das coordenadas geográficas (latitude e longitude) de cada localidade. Esse procedimento permitiu incorporar a distribuição espacial na análise e identificar quais municípios pertencem a cada cluster. Dessa forma, o modelo agrupou regiões com comportamentos semelhantes e também permitiu visualizar espacialmente esses padrões, favorecendo a interpretação regional.

A figura 21 representa a georreferenciação dos municípios presentes na formação de cada cluster diferenciado por coloração.

O cluster 0 representa os municípios do norte e planalto norte do estado, em azul. Já em vermelho, representando a região oeste está o cluster 1. Em sequência, o cluster 2, engloba os municípios do sul do estado e encosta da serra, evidenciado na coloração rosa. Por último, o cluster 3 que reúne municípios localizados na faixa litorânea centro-norte, em parte do Médio Vale do Itajaí e na região da Grande Florianópolis.

Figura 21 - Georreferenciamento dos clusters por município



5 CONCLUSÕES E TRABALHOS FUTUROS

5.1 CONCLUSÕES

O presente trabalho teve como propósito desenvolver um modelo de previsão de desastres naturais para o estado de Santa Catarina, utilizando redes neurais do tipo Long Short-Term Memory (LSTM) e técnicas de clusterização voltadas à otimização logística na resposta a eventos extremos. Ao longo do estudo, foi possível

demonstrar o potencial das abordagens computacionais aplicadas à gestão de riscos climáticos.

A análise do histórico de desastres naturais em Santa Catarina permitiu compreender a distribuição temporal e espacial dos eventos, identificando padrões relevantes e evidenciando a influência de fatores climáticos.

Em seguida, procedeu-se à coleta, organização e preparação dos dados históricos, garantindo sua adequação para o uso em redes neurais. As etapas de seleção de variáveis, normalização, tratamento de valores ausentes e construção de janelas temporais permitiram estruturar os dados para utilização na arquitetura LSTM.

O desenvolvimento e a avaliação do modelo LSTM demonstraram sua capacidade de prever eventos climáticos. As métricas obtidas indicaram que o modelo tem potencial de antecipar variações relevantes no comportamento climático, reforçando sua aplicabilidade como ferramenta de apoio à tomada de decisão em contextos de gestão de risco.

Por fim, a aplicação do algoritmo K-Means possibilitou agrupar os municípios catarinenses com base nas características espaciais dos eventos previstos. O georreferenciamento permitiu integrar o componente territorial ao processo de análise, facilitando a identificação de regiões com perfis de risco semelhantes. Esses clusters oferecem suporte à otimização logística, auxiliando na distribuição estratégica de recursos emergenciais e no planejamento de ações preventivas de forma regionalizada.

Dessa forma, conclui-se que a integração entre modelos de previsão baseados em aprendizado profundo e técnicas de clusterização representa uma abordagem promissora para aprimorar a capacidade de antecipação e resposta a desastres naturais em Santa Catarina. Embora ainda existam desafios relacionados à disponibilidade, precisão e granularidade dos dados, os resultados obtidos reforçam o potencial dessas metodologias para reduzir impactos sociais, econômicos e ambientais, contribuindo para políticas públicas mais eficientes e para o fortalecimento da resiliência das comunidades vulneráveis.

Em conclusão, foi observado que o método de redes neurais emerge como uma ferramenta relevante para aprimorar a capacidade de previsão em diversos setores. Sua capacidade de analisar grandes volumes de dados, identificar padrões complexos e adaptar-se dinamicamente a mudanças nas condições oferece uma abordagem ágil e eficiente para antecipar cenários futuros.

Da mesma forma, a utilização de modelos como a clusterização desempenham um papel crucial no contexto de otimização de recursos e agilidade nas demandas, como é o caso abordado neste estudo, de maneira que estar geolocalizado estrategicamente significa um atendimento mais eficaz as pessoas atingidas por eventos extremos e aos danos ocasionados.

A revisão bibliográfica desempenhou um papel importante ao contextualizar o estudo dentro do conhecimento existente e forneceu uma base teórica necessária.

É possível observar que apesar dos resultados obtidos pelo modelo de predição apresentar valores expressivos em relação as métricas de análise (MAE 13% e RMSE 19%) a parametrização dos clusters, com o k ótimo apresentou uma grande área não ocupada pelo modelo.

5.2 TRABALHOS FUTUROS

Embora o modelo desenvolvido neste trabalho tenha apresentado resultados satisfatórios na previsão de desastres naturais e na segmentação territorial via clusterização, diversas oportunidades de aprimoramento e aprofundamento se apresentam para pesquisas futuras.

Em primeiro lugar, recomenda-se a ampliação da base de dados utilizada, incorporando variáveis meteorológicas e ambientais de maior granularidade, como índices de instabilidade atmosférica, pressão atmosférica, velocidade do vento, umidade relativa, dados de radar meteorológico e imagens de satélite. A inclusão dessas variáveis pode aumentar a capacidade preditiva do modelo LSTM e permitir a identificação de padrões mais complexos relacionados à formação de eventos extremos.

Além disso, estudos futuros podem explorar arquiteturas mais avançadas de aprendizado profundo, ou ainda modelos híbridos que combinem redes neurais com algoritmos estatísticos de séries temporais. Essas abordagens podem melhorar o desempenho, reduzir o tempo de treinamento e lidar de forma mais eficiente com variações espaciais e temporais dos dados.

Outra possibilidade relevante consiste na realização de validações temporais mais robustas, como testes sazonais, validação cruzada para séries temporais e análises de sensibilidade que permitam avaliar o comportamento do modelo frente a diferentes cenários climáticos. Da mesma forma, integrar dados em tempo real por

meio de APIs meteorológicas poderia transformar o modelo em uma ferramenta operacional de monitoramento contínuo.

Por fim, sugere-se integrar os resultados obtidos em plataformas interativas de geovisualização, como dashboards em Power BI, ArcGIS, QGIS, permitindo que gestores públicos e equipes técnicas possam consultar previsões, identificar zonas de risco e priorizar ações de forma mais intuitiva e eficiente.

REFERÊNCIAS

- ALMEIDA, D. F.; et al. Vulnerabilidade a desastres naturais em Santa Catarina. **Revista Brasileira de Geografia**, v. 72, n. 3, p. 45-59, 2018.
- ALMEIDA, F. et al. A análise de risco e previsão de desastres naturais: **tendências e desafios**. São Paulo: Editora Acadêmica, 2018.
- BALCI, S.; BEYHAN, G. Application of clustering methods in humanitarian logistics: **A review**. **Sustainability**, v. 13, n. 3, p. 1358, 2021.
- BALCI, E.; BEYHAN, B. Optimization of Humanitarian Logistics Networks for Disaster Response. **International Journal of Disaster Risk Reduction**, v. 55, p. 102-114, 2021.
- BALCIK, B. et al. Coordination in humanitarian relief chains: practices, challenges and opportunities. **International Journal Production Economics**. Science Direct, v.126, n. 1, p. 22-34, 2010.
- BROWNLEE, J. Deep Learning for Time Series Forecasting: **Predict the Future with MLPs, CNNs and LSTMs in Python**. 1. ed. Melbourne: Machine Learning Mastery, 2020.
- CHATFIELD, C. The Analysis of Time Series: **An Introduction**, v. 6. 2024.
- CHAI, T.; DRAXLER, R. R.: Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature, **Geosci. Model Dev.**, 7, 1247–1250, <https://doi.org/10.5194/gmd-7-1247-2014>, 2014.
- CHEN, W.; WANG, L. Applications of linear programming in disaster response logistics. **Journal of Humanitarian Logistics**, v. 12, p. 230-245, 2017.
- CHEN, Y.; CHEN, C.; CHEN, W. A data-driven approach to disaster relief logistics planning. **International Journal of Disaster Risk Reduction**, v. 45, 2020.
- CHICCO, D.; WARRENS, M. J.; JURMAN, G. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. **Peerj Computer Science**, v. 7. 2021.
- CHOLLET, F. **Deep Learning with Python**. 2. ed. New York: Manning, 2021.
- DEFESA CIVIL DE SANTA CATARINA. **Relatório anual de desastres**. Florianópolis, 2021.
- FANG, H.; et al. Hybrid models for natural disaster prediction and early warning systems. **Geoscience and Remote Sensing Letters**, v. 19, n. 3, p. 152-160, 2022.

FERREIRA, A. L.; et al. Aplicação de redes neurais na previsão de desastres naturais. **Journal of Environmental Science and Technology**, v. 13, n. 5, p. 429-437, 2020.

GARCIA, L. et al. Deep learning for weather forecasting: A review. **Journal of Artificial Intelligence Research**, v. 69, p. 879-918, 2020.

GÉRON, A. **Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. O'Reilly Media, 2017, 576p.

GÉRON, A. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow**. 3. ed. Sebastopol: O'Reilly Media, 2023.

GIL, A. C. **Métodos e técnicas de pesquisa social**. 7. ed. São Paulo: Atlas, 2019.

GOMES, A. L.; BAPTISTA, J. S. Logística humanitária: desafios e possibilidades no contexto dos desastres. **Revista Gestão & Tecnologia**, v. 17, n. 1, p. 99–117, 2017.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. MIT Press, 2016.

GUO, Y.; et al. Deep learning for disaster prediction and forecasting: A comprehensive review. **IEEE Transactions on Industrial Informatics**, v. 18, n. 1, p. 100-110, 2022.

HAIR JR., J. F.; BLACK, W. C.; BABIN, B. J.; ANDERSON, R. E. **Multivariate Data Analysis**. Pearson, 2014.

HAN, J.; KAMBER, M.; PEI, J. Data Mining: **Concepts and Techniques**. 4. ed. Amsterdam: Elsevier, 2022.

HASSANI, H.; et al. Machine learning: Applications and methods. **Springer Series in Statistics**, v. 15, n. 2, p. 75-90, 2017.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The elements of statistical learning: data mining, inference, and prediction (2ª ed.)**. Springer, 2009.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural Computation**, v. 9, n. 8, p. 1735-1780, 1997.

HYNDMAN, R. J.; KOEHLER, A. B. *Another Look at Measures of Forecast Accuracy*. **International Journal of Forecasting**, v. 22, n. 4, p. 679–688, 2006.

INMET - Instituto Nacional de Meteorologia. **Boletim Climático Anual**. Brasília, 2022.

IPCC – Intergovernmental Panel on Climate Change. **Sixth Assessment Report: Impacts, Adaptation and Vulnerability**. Genebra: IPCC, 2021.

JAIN, A. K. Data Clustering: 50 Years Beyond K-Means. In: **Proceedings of the 10th International Conference on Data Mining**, 2010. p. 26-34.

JAIN, A. K. Data Clustering: 50 Years Beyond K-Means. **Pattern Recognition Letters**, v. 31, n. 8, p. 651-666, 2010.

JONER, K.; ÁVILA M. R. R.; MATTEDI M. A. Territorialidade e desastre: a gestão dos desastres no Brasil com base no estudo de caso da Defesa Civil em Santa Catarina. urbe. **Revista Brasileira de Gestão Urbana**, v. 13, e20200061, 2021.

KODINARIAS, I. D.; GAD, M. E. K-means clustering algorithm: Application, evaluation, and comparison with hierarchical clustering. **International Journal of Computer Engineering and Applications**, v. 2, n. 4, p. 158–167, 2013.

LAKEFS. Data Preprocessing in Machine Learning: **Steps and Best Practices**. [S.I.], 2023. Disponível em: <https://lakefs.io/blog/data-preprocessing-in-machine-learning/> Acesso em 25 out. 2025.

LIMA, Fabiana Santos; OLIVEIRA, Daniel de; GONÇALVES, Mirian Buss. Formação de clusters para o gerenciamento da cadeia de suprimentos em operações humanitárias. **Exacta**, São Paulo, v. 12, n. 1, p. 87-98, 2014.

LIMA, M. A. F. et al. O ciclo da logística humanitária: fases, práticas e desafios. **Revista Brasileira de Logística**, v. 8, n. 1, p. 112–128, 2016.

LI, Q. et al. Real-time disaster response optimization using deep learning techniques. **IEEE Transactions on Industrial Informatics**, v. 17, p. 1245-1256, 2021.

MARCONI, M. A.; LAKATOS, E. M. **Fundamentos de metodologia científica**. 7. ed. São Paulo: Atlas, 2010.

MARTINS, M. E. G. Coeficiente de determinação. **Revista de Ciência Elementar**, v. 6. 2018.

MCKINNEY, W. **Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython**. O'Reilly Media, 2017

MIGUEL, P. A. C. Estudo de caso na engenharia de produção: estruturação e recomendações para sua condução. **Production**, v. 17, p. 216-229, 2007.

NEUMANN, R. Logística humanitária: práticas e desafios em tempos de crise. **Revista Organizações em Contexto**, v. 13, n. 26, p. 1–15, 2017.

OLIVEIRA, A. M.; et al. Clustering and genetic algorithms for optimizing disaster relief distribution. **Computers & Operations Research**, v. 57, n. 9, p. 124-134, 2023.

OLIVEIRA, M. S.; SOUSA, C. F. Estratégias logísticas na fase de reconstrução pós-desastre: uma revisão de literatura. **Revista de Administração e Inovação Sustentável**, v. 11, n. 3, p. 1–18, 2021.

PARK, Y. et al. Integrating disaster prediction systems with humanitarian logistics operations. **Journal of Humanitarian Logistics and Supply Chain Management**, v. 10, n. 2, p. 123–139, 2020.

RIBEIRO, G.; et al. Reinforcement learning for adaptive disaster response systems. **IEEE Transactions on Computational Intelligence**, v. 18, n. 3, p. 451-461, 2023.

ROCHA, T.; ALMEIDA, P. Logistic strategies for humanitarian aid distribution in Brazil. **Revista de Logística Humanitária**, v. 12, n. 1, p. 89-104, 2021.

SARWADE, M. P.; SHINDE, S. A.; PATIL, V. S. Predictive Modeling of Extreme Weather Forecasting Events: An LSTM Approach. **Current World Environment**, v. 19, n. 1, p. 1–10, 2024.

SANTOS, E. D.; et al. The role of urbanization in increasing disaster risks. **Journal of Urban Planning**, v. 44, n. 2, p. 214-228, 2020.

SILVA, L. A. da. Gestão da logística humanitária: proposta de referencial teórico. Dissertação (Mestrado em Administração) – **Instituto COPPEAD de Administração, Universidade Federal do Rio de Janeiro**, 2021.

SILVA, R. et al. Impactos das enchentes em Santa Catarina. **Revista Brasileira de Gestão Urbana**, v. 7, p. 45-62, 2015.

SILVA, R.; SANTOS, M.; OLIVEIRA, A. Logística humanitária: análise de cinco casos de desastres naturais. **Revista Brasileira de Gestão Urbana**, v. 14, n. 1, p. 45–62, 2022.

SILVA, J. R.; SILVA, T. P. Logística humanitária e a gestão do ciclo de desastres: uma análise aplicada ao contexto brasileiro. **Revista de Gestão de Emergências**, v. 5, n. 2, p. 77–92, 2019.

SILVESTRE, A. L. **Análise de dados e estatística descritiva**. Escolar Editora, 2007.

SMITH, K. Environmental Hazards: Assessing Risk and Reducing Disaster. **Routledge**, 2013.

SHEU, J. B. Dynamic relief-demand management for emergency logistics operations under large-scale disasters. **Transportation Research Part E: Logistics and Transportation Review**, v. 46, n. 1, p. 1-17, 2010.

YIN, R. K. Estudo de caso: **planejamento e métodos**. 5. ed. Porto Alegre: Bookman, 2015.

WICKHAM, H. **R for Data Science**. Sebastopol, 2023.

WITTEN, I. H.; FRANK, E.; HALL, M. A. **Data Mining: Practical Machine Learning Tools and Techniques**. Morgan Kaufmann, 2016.

ZHU, Y. et al. Density-based spatial clustering for disaster risk management. **Geographic Information Science**, v. 33, p. 56-72, 2019.

ZHANG, Y. et al. Optimizing Disaster Relief Operations with K-Means Clustering. **Operations Research Letters**, v. 46, p. 129-134, 2018.

ZHANG, H. et al. Forecasting extreme weather events using LSTM networks. **Neural Networks**, v. 115, p. 108-117, 2019.

ZHANG, G. et al. Artificial neural networks in highway infrastructure management. **Journal of Infrastructure Systems**, vol. 26, no. 1, 2020.

APENDICE A – CÓDIGO REDES NEURAIAS LSTM

```

import argparse
import sys
from pathlib import Path
import math

import numpy as np
import pandas as pd

import tensorflow as tf
from keras import layers, callbacks, models

from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_absolute_error, mean_squared_error

import matplotlib.pyplot as plt

def make_sequences(X: np.ndarray, y: np.ndarray, lookback: int, horizon: int):
    Xs, ys = [], []
    for i in range(len(X) - lookback - horizon + 1):
        Xs.append(X[i : i + lookback])
        ys.append(y[i + lookback + horizon - 1])
    return np.array(Xs), np.array(ys)

def main():
    p = argparse.ArgumentParser(description="Treinar LSTM para predição (genérico) com pré-processamento robusto.")
    p.add_argument("--file", required=True, help="Caminho para .xlsx ou .csv")
    p.add_argument("--sheet", type=int, default=0, help="Índice da planilha (se Excel)")
    p.add_argument("--date-col", required=True, help="Nome da coluna de data")
    p.add_argument("--target-col", required=True, help="Nome da coluna alvo (ex.: Evento, ocorrências)")
    p.add_argument("--feature-cols", default="", help="Colunas de features separadas por vírgula (opcional)")
    p.add_argument("--freq", default=None, help="Reamostrar para frequência (ex.: D, W, M) – opcional")
    p.add_argument("--fill", default="ffill", choices=["ffill", "bfill", "zero", "mean"], help="Preenchimento de faltantes")
    p.add_argument("--lookback", type=int, default=30, help="Passos históricos")
    p.add_argument("--horizon", type=int, default=1, help="Passos à frente para prever")
    p.add_argument("--val-size", type=float, default=0.15, help="Proporção para validação (final da série)")

```

```

p.add_argument("--test-size", type=float, default=0.15, help="Proporção para
teste (final da série)")
p.add_argument("--epochs", type=int, default=50)
p.add_argument("--batch-size", type=int, default=32)
p.add_argument("--lstm-units", type=int, default=64)
p.add_argument("--lstm-layers", type=int, default=1)
p.add_argument("--dropout", type=float, default=0.2)
p.add_argument("--lr", type=float, default=1e-3)
p.add_argument("--model-out", default="modelo_lstm.keras")
p.add_argument("--plot-out", default="treino_performance.png")
args = p.parse_args()

path = Path(args.file)
if not path.exists():
    print(f"[ERRO] Arquivo não encontrado: {path}", file=sys.stderr)
    sys.exit(1)

# Carregar dados
if path.suffix.lower() in [".xlsx", ".xls"]:
    df = pd.read_excel(path, sheet_name=args.sheet, engine="openpyxl")
else:
    df = pd.read_csv(path)

# Checagens básicas
if args.date_col not in df.columns:
    print(f"[ERRO] Coluna de data '{args.date_col}' não encontrada. Colunas:
{list(df.columns)}", file=sys.stderr)
    sys.exit(1)
if args.target_col not in df.columns:
    print(f"[ERRO] Coluna alvo '{args.target_col}' não encontrada. Colunas:
{list(df.columns)}", file=sys.stderr)
    sys.exit(1)

feature_cols = [c.strip() for c in args.feature_cols.split(",") if c.strip()]
for c in feature_cols:
    if c not in df.columns:
        print(f"[ERRO] Feature '{c}' não encontrada nas colunas.",
file=sys.stderr)
        sys.exit(1)

# Preparar data
df[args.date_col] = pd.to_datetime(df[args.date_col], errors="coerce")
df = df.dropna(subset=[args.date_col]).sort_values(args.date_col)
df = df.set_index(args.date_col)

if args.freq:
    agg = {args.target_col: "sum"}
    for c in feature_cols:
        agg[c] = "mean"

```

```

df = df.resample(args.freq).agg(agg)

if args.fill == "ffill":
    df = df.ffmpeg()
elif args.fill == "bfill":
    df = df.bfill()
elif args.fill == "zero":
    df = df.fillna(0)
else:
    df = df.fillna(df.mean(numeric_only=True))

# ----- Alvo categórico -> numérico -----
from pandas.api.types import is_object_dtype, is_string_dtype,
is_categorical_dtype
target_mapping = None
if is_object_dtype(df[args.target_col]) or
is_string_dtype(df[args.target_col]) or is_categorical_dtype(df[args.target_col]):
    df[args.target_col] = df[args.target_col].astype(str).str.strip()
    unique_vals = list(pd.unique(df[args.target_col]))
    mapping = {val: idx for idx, val in enumerate(unique_vals)}
    target_mapping = mapping.copy()
    print("[INFO] Mapeamento categórico → numérico da coluna alvo:")
    for k, v in mapping.items():
        print(f"    {k} → {v}")
    df[args.target_col] = df[args.target_col].map(mapping).astype("float32")

# ----- Features: one-hot em não numéricas -----
from pandas.api.types import is_numeric_dtype
base_cols = feature_cols + [args.target_col]
data = df[base_cols].copy()

non_numeric_feats = [c for c in feature_cols if not is_numeric_dtype(data[c])]
if non_numeric_feats:
    print("[INFO] Aplicando one-hot em features não numéricas:",
non_numeric_feats)
    data = pd.get_dummies(data, columns=non_numeric_feats, drop_first=False)

non_numeric_any = [c for c in data.columns if not is_numeric_dtype(data[c])]
if non_numeric_any:
    print("[AVISO] Removendo colunas não numéricas ainda presentes:",
non_numeric_any)
    data = data.drop(columns=non_numeric_any)

# Escalonamento (MinMax 0-1) de todas as colunas
scalers = {}
for c in data.columns:
    s = MinMaxScaler()
    data[c] = s.fit_transform(data[[c]])
    scalers[c] = s

```

```

# Construção das janelas
features_used = [c for c in data.columns if c != args.target_col]
X_all = data[features_used].values if features_used else
data[[args.target_col]].values
y_all = data[args.target_col].values

X_seq, y_seq = make_sequences(X_all, y_all, args.lookback, args.horizon)
if len(X_seq) < 10:
    print("[AVISO] Poucos exemplos após janelamento. Considere reduzir
lookback/horizon ou revisar dados.", file=sys.stderr)

# Split temporal
n = len(X_seq)
n_test = int(n * args.test_size)
n_val = int(n * args.val_size)
n_train = n - n_val - n_test
if n_train <= 0 or n_val <= 0 or n_test <= 0:
    print("[ERRO] Partição inválida. Ajuste val-size/test-size.",
file=sys.stderr)
    sys.exit(1)

X_train, y_train = X_seq[:n_train], y_seq[:n_train]
X_val, y_val = X_seq[n_train:n_train+n_val], y_seq[n_train:n_train+n_val]
X_test, y_test = X_seq[n_train+n_val:], y_seq[n_train+n_val:]

# Depuração de formas
print("[DEBUG] n_train, n_val, n_test:", n_train, n_val, n_test)
print("[DEBUG] X_seq.shape:", X_seq.shape)
print("[DEBUG] X_train.shape:", X_train.shape)
print("[DEBUG] X_val.shape:", X_val.shape)
print("[DEBUG] X_test.shape:", X_test.shape)

# ----- Modelo -----
model = models.Sequential()
# Entrada explícita baseada no shape do treino
model.add(layers.Input(shape=X_train.shape[1:]))
for i in range(args.lstm_layers):
    return_sequences = (i < args.lstm_layers - 1)
    model.add(layers.LSTM(args.lstm_units, return_sequences=return_sequences))
    model.add(layers.Dropout(args.dropout))
model.add(layers.Dense(1))

opt = tf.keras.optimizers.Adam(learning_rate=args.lr)
model.compile(optimizer=opt, loss="mse",
metrics=[tf.keras.metrics.MeanAbsoluteError(name="mae")])

es = callbacks.EarlyStopping(monitor="val_loss", patience=15,
restore_best_weights=True)

```

```

    rlrop = callbacks.ReduceLROnPlateau(monitor="val_loss", factor=0.5,
patience=5)

    hist = model.fit(
        X_train, y_train,
        validation_data=(X_val, y_val),
        epochs=args.epochs,
        batch_size=args.batch_size,
        callbacks=[es, rlrop],
        verbose=1
    )

    # Avaliação
    pred_test = model.predict(X_test).ravel()
    mae = mean_absolute_error(y_test, pred_test)
    rmse = math.sqrt(mean_squared_error(y_test, pred_test))
    mae_norm, rmse_norm = mae, rmse
    mae_orig, rmse_orig = None, None

    try:
        inv_true = scalers[args.target_col].inverse_transform(y_test.reshape(-
1,1)).ravel()
        inv_pred = scalers[args.target_col].inverse_transform(pred_test.reshape(-
1,1)).ravel()
        mae_inv = mean_absolute_error(inv_true, inv_pred)
        rmse_inv = math.sqrt(mean_squared_error(inv_true, inv_pred))
        print(f"[MÉTRICAS] Teste (normalizado) -> MAE: {mae:.6f} | RMSE:
{rmse:.6f}")
        print(f"[MÉTRICAS] Teste (escala original) -> MAE: {mae_inv:.3f} | RMSE:
{rmse_inv:.3f}")
        mae_orig, rmse_orig = mae_inv, rmse_inv
    except Exception as _:
        print(f"[MÉTRICAS] Teste (normalizado) -> MAE: {mae:.6f} | RMSE:
{rmse:.6f}")

    # Gráficos
    plt.figure()
    plt.plot(hist.history.get("loss", []), label="loss")
    if "val_loss" in hist.history:
        plt.plot(hist.history["val_loss"], label="val_loss")
    plt.title("Histórico de treino")
    plt.xlabel("Época")
    plt.ylabel("Loss")
    plt.legend()
    plt.tight_layout()
    plt.savefig(args.plot_out, dpi=140)
    plt.close()

    plt.figure()

```

```

plt.plot(y_test, label="real")
plt.plot(pred_test, label="previsto")
plt.title("Teste - Real vs Previsto (normalizado)")
plt.xlabel("Amostra")
plt.ylabel("Alvo (norm.)")
plt.legend()
plt.tight_layout()
plt.savefig("teste_real_vs_previsto.png", dpi=140)
plt.close()

# Salvar modelo
model.save(args.model_out)
print(f"[OK] Modelo salvo em: {args.model_out}")
print(f"[OK] Gráficos salvos em: {args.plot_out} e
teste_real_vs_previsto.png")

# Relatório
try:
    report_path = Path("relatorio_lstm.txt")
    with open(report_path, "w", encoding="utf-8") as f:
        f.write("RELATÓRIO DO TREINAMENTO LSTM\n")
        f.write("-----\n")
        f.write(f"Arquivo de dados: {path}\n")
        f.write(f>Date col: {args.date_col}\n")
        f.write(f>Target col: {args.target_col}\n")
        f.write(f>Features usadas ({len(features_used)}): {features_used}\n")
        f.write(f>Frequência (reamostragem): {args.freq}\n")
        f.write(f>Preenchimento: {args.fill}\n")
        f.write(f>Lookback: {args.lookback} | Horizon: {args.horizon}\n")
        f.write(f>Split -> train/val/test: {1-args.val_size-
args.test_size:.2f}/{args.val_size:.2f}/{args.test_size:.2f}\n")
        f.write("\n--- Shapes ---\n")
        f.write(f>X_seq.shape: {X_seq.shape}\n")
        f.write(f>X_train.shape: {X_train.shape} | X_val.shape: {X_val.shape}
| X_test.shape: {X_test.shape}\n")
        f.write("\n--- Métricas ---\n")
        f.write(f>MAE (normalizado): {mae_norm}\n")
        f.write(f>RMSE (normalizado): {rmse_norm}\n")
        f.write(f>MAE (original): {mae_orig}\n")
        f.write(f>RMSE (original): {rmse_orig}\n")
        if 'target_mapping' in locals() and target_mapping:
            f.write("\n--- Mapeamento do alvo (categórico → numérico) ---\n")
            for k, v in target_mapping.items():
                f.write(f>{k} -> {v}\n")
            f.write("\nArquivos gerados:\n")
            f.write(f" - Modelo: {args.model_out}\n")
            f.write(f" - Gráficos: {args.plot_out}, teste_real_vs_previsto.png\n")
    print(f"[OK] Relatório salvo em: {report_path.resolve()}")
except Exception as e:

```

```

    print(f"[AVISO] Falha ao escrever relatório: {e}")

# tenta recuperar o mapeamento categórico (se existir)
label_map = None
if 'target_mapping' in locals() and target_mapping:
    label_map = {v: k for k, v in target_mapping.items()}

# previsões em escala original (inv_pred) ou normalizada (pred_test)
y_pred = inv_pred if 'inv_pred' in locals() else pred_test

# recupera municípios correspondentes às últimas previsões
n_pred = len(y_pred)
municipios = df.iloc[-n_pred:]["municipio"].values if "municipio" in
df.columns else [None] * n_pred

# mapeia IDs numéricos de volta a rótulos (se aplicável)
if label_map is not None:
    y_pred_labels = [label_map.get(int(round(v)), v) for v in y_pred]
else:
    y_pred_labels = y_pred

df_out = pd.DataFrame({
    "municipio": municipios,
    "evento_previsto": y_pred_labels
})

out_path = Path("previsoes_lstmtest1.csv")
df_out.to_csv(out_path, index=False, encoding="utf-8-sig")
print(f"[OK] Previsões salvas em: {out_path.resolve()}")
print(df_out.head(10))

if __name__ == "__main__":
    main()

```

APENDICE B – CÓDIGO CLUSTER K-MEANS

```

import pandas as pd
import numpy as np
from pathlib import Path
import matplotlib.pyplot as plt

from geobr import read_municipality, read_state
import geopandas as gpd
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from unicode import unicode

def norm(s: str) -> str:
    if pd.isna(s): return ""
    return unicode(str(s)).strip().lower()

def attach_sc_centroids(df, muni_col='município'):
    gdf_sc = read_municipality(code_muni='SC', year=2020,
simplified=True).to_crs(4326)
    gdf_sc['name_norm'] = gdf_sc['name_muni'].apply(norm)
    gdf_sc['lat'] = gdf_sc.geometry.centroid.y
    gdf_sc['lon'] = gdf_sc.geometry.centroid.x
    out = df.copy()
    out['name_norm'] = out[muni_col].apply(norm)
    merged = out.merge(gdf_sc[['name_norm', 'lat', 'lon', 'name_muni', 'code_muni']],
on='name_norm', how='left')
    miss = merged['lat'].isna().sum()
    if miss:
        print(f"[AVISO] {miss} linhas não foram casadas com municípios de SC.")
    return merged

def run_kmeans(df_with_coords, n_clusters=6, random_state=42):
    tmp = df_with_coords.dropna(subset=['lat', 'lon']).copy()
    if tmp.empty:
        raise ValueError("Sem coordenadas válidas para clusterizar.")
    km = KMeans(n_clusters=n_clusters, random_state=random_state, n_init='auto')
    tmp['cluster'] = km.fit_predict(tmp[['lat', 'lon']].values)
    return tmp, km

def plot_static_sc_points(df_clusters, png_out='mapa_clusters_sc_kmeans.png',
dpi=300):
    gdf_state = read_state(code_state='SC', year=2020,
simplified=True).to_crs(4326)
    fig, ax = plt.subplots(figsize=(8.5, 8.5), dpi=dpi)
    gdf_state.boundary.plot(ax=ax, linewidth=0.8, color='black')

    clusters = sorted(df_clusters['cluster'].unique())

```

```

import itertools
palette = plt.cm.tab20.colors if len(clusters) > 10 else plt.cm.tab10.colors
color_cycle = itertools.cycle(palette)
colors = {c: next(color_cycle) for c in clusters}

for c in clusters:
    sub = df_clusters[df_clusters['cluster'] == c]
    ax.scatter(sub['lon'], sub['lat'], s=12, alpha=0.9, label=f'Cluster {c}',
c=[colors[c]])

ax.set_title("Clusterização – KMeans (SC)", fontsize=12)
ax.set_xlabel("Longitude"); ax.set_ylabel("Latitude")
ax.legend(loc='lower left', fontsize=8, frameon=True)
ax.set_aspect('equal', adjustable='datalim')
plt.tight_layout(); plt.savefig(png_out, dpi=dpi, bbox_inches='tight');
plt.close(fig)
print(f"[OK] Mapa (pontos) salvo em: {png_out}")

def plot_sc_municipios_por_cluster(
    df_clusters,
    png_out="mapa_clusters_sc_municipios1.png",
    figsize=(12, 12),
):
    url_mapa_sc = "https://raw.githubusercontent.com/tbrugz/geodata-
br/refs/heads/master/geojson/geojs-42-mun.json"

    try:
        mapa_sc_mun = gpd.read_file(url_mapa_sc)
    except Exception as e:
        print(f"Erro ao carregar o arquivo GeoJSON do mapa: {e}")
        print("O mapa não será plotado, mas a análise de clusters ainda é
válida.")
        return

    possible_name_cols = ["name", "NM_MUNICIP", "nm_municip", "NM_MUN",
"municipio"]
    name_col = None
    for c in possible_name_cols:
        if c in mapa_sc_mun.columns:
            name_col = c
            break

    if name_col is None:
        print("Não foi possível identificar a coluna de nome do município no
GeoJSON.")
        print("Colunas disponíveis:", list(mapa_sc_mun.columns))
        return

    mapa_sc_mun["name_norm"] = mapa_sc_mun[name_col].apply(norm)

```

```

def build_cluster_clouds(df_clusters, buffer_m=15000, min_pts=5):
    df_pts = df_clusters.dropna(subset=["lon", "lat"]).copy()
    if df_pts.empty:
        raise ValueError("Sem coordenadas válidas para nuvens.")
    gdf = gpd.GeoDataFrame(df_pts, geometry=gpd.points_from_xy(df_pts["lon"],
df_pts["lat"]), crs="EPSG:4326")
    gdf_m = gdf.to_crs(3857) # metros
    polys = []
    for c, grp in gdf_m.groupby("cluster"):
        if len(grp) < min_pts: continue
        hull = grp.unary_union.convex_hull.buffer(buffer_m)
        polys.append({"cluster": int(c), "geometry": hull})
    if not polys:
        raise ValueError("Nenhum cluster atingiu 'min_pts' para gerar nuvem.")
    clouds_m = gpd.GeoDataFrame(polys, crs="EPSG:3857")
    return clouds_m.to_crs(4326)

def plot_static_sc_clouds(df_clusters,
png_out="mapa_clusters_sc_kmeans_nuvens.png", dpi=300, buffer_m=15000, min_pts=5):
    gdf_state = read_state(code_state="SC", year=2020,
simplified=True).to_crs(4326)
    clouds = build_cluster_clouds(df_clusters, buffer_m=buffer_m, min_pts=min_pts)

    fig, ax = plt.subplots(figsize=(8.5, 8.5), dpi=dpi)
    gdf_state.boundary.plot(ax=ax, linewidth=0.8, color="black")

    import itertools
    palette = plt.cm.Pastel1.colors if len(clouds) <= 9 else plt.cm.Pastel2.colors
    colors = itertools.cycle(palette)

    for _, row in clouds.iterrows():
        color = next(colors)
        gpd.GeoSeries([row.geometry], crs="EPSG:4326").plot(
            ax=ax, facecolor=color, edgecolor="k", alpha=0.5, linewidth=0.8,
label=f"Cluster {row.cluster}"
        )

    for c, grp in df_clusters.groupby("cluster"):
        grp_xy = grp[["lon", "lat"]].dropna()
        if grp_xy.empty: continue
        ax.text(grp_xy["lon"].mean(), grp_xy["lat"].mean(),
f"{c}\n(n={len(grp)})",
            fontsize=8, ha="center", va="center")

    ax.set_title("Clusterização – KMeans (Nuvens, SC)", fontsize=12)
    ax.set_xlabel("Longitude"); ax.set_ylabel("Latitude")

```

```

    ax.set_aspect("equal", adjustable="datalim")
    ax.legend(loc="lower left", fontsize=8, frameon=True)
    plt.tight_layout(); plt.savefig(png_out, dpi=dpi, bbox_inches="tight");
plt.close(fig)
    print(f"[OK] Mapa (nuvens) salvo em: {png_out}")

# Preferência por previsões exportadas
DATA_PATH = Path("previsoes_lstm.csv")
if not DATA_PATH.exists():
    DATA_PATH = Path("dados-modelo.xlsx"); SHEET_INDEX = 0
    assert DATA_PATH.exists(), f"Arquivo não encontrado: {DATA_PATH.resolve()}"
    df = pd.read_excel(DATA_PATH, sheet_name=SHEET_INDEX, engine="openpyxl")
    if 'y_pred' not in df.columns and 'Evento' in df.columns:
        df = df.rename(columns={'Evento': 'y_pred'})
else:
    df = pd.read_csv(DATA_PATH)

if 'municipio' not in df.columns:
    raise ValueError("Coluna 'municipio' não encontrada.")
if 'y_pred' not in df.columns:
    df['y_pred'] = 'evento'

df_geo = attach_sc_centroids(df,
muni_col='municipio').dropna(subset=['lat', 'lon']).copy()

from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

inertias = []
ks = list(range(K_MIN, K_MAX + 1))

for k in ks:
    km = KMeans(n_clusters=k, random_state=42, n_init="auto").fit(X)
    inertias.append(km.inertia_)

plt.figure()
plt.plot(ks, inertias, marker="o", color="tab:blue")
plt.title("Método do Cotovelo – Inércia por K")
plt.xlabel("Número de Clusters (K)")
plt.ylabel("Inércia (Soma dos Erros Quadráticos)")
plt.grid(True, linestyle="--", alpha=0.5)
plt.tight_layout()
plt.savefig("elbow_method.png", dpi=160)
plt.show()
print("[OK] Gráfico do método do cotovelo salvo")

df_clu, km = run_kmeans(df_geo, n_clusters=final_k, random_state=42)
df_clu.to_csv("clusters_sc_kmeans.csv", index=False)

```

```

print("[OK] clusters_sc_kmeans.csv salvo.")

# Mapas
plot_static_sc_points(df_clu, png_out="mapa_clusters_sc_kmeans.png", dpi=300)
plot_static_sc_clouds(df_clu, png_out="mapa_clusters_sc_kmeans_nuvens.png",
                      dpi=300, buffer_m=15000, min_pts=5)

# Estatísticas por cluster
tab = (df_clu.groupby(['cluster', 'y_pred']).size()
       .groupby(level=0).apply(lambda s:
100*s/s.sum()).rename('perc').reset_index())
pivot = tab.pivot(index='cluster', columns='y_pred',
                  values='perc').fillna(0).sort_index()
print("Distribuição de eventos por cluster (%)")
display(pivot.round(2))
pivot.to_csv("cluster_eventos_percentuais.csv")

# Médias numéricas por cluster (se houver)
num_cols = df_clu.select_dtypes(include=[np.number]).columns.tolist()
for c in ['lat', 'lon', 'cluster', 'code_muni']:
    if c in num_cols:
        num_cols.remove(c)
if num_cols:
    means = df_clu.groupby('cluster')[num_cols].mean().sort_index()
    print("Médias por cluster (variáveis numéricas)")
    display(means.round(3))
    means.to_csv("cluster_medias_numericas.csv")
else:
    print("[INFO] Não há colunas numéricas extras para calcular médias.")

```