

FACULDADE DE INFORMÁTICA E ADMINISTRAÇÃO PAULISTA
INTELIGÊNCIA ARTIFICIAL

ANA BEATRIZ DUARTE DOMINGUES

CARLOS EMILIO CASTILLO ESTRADA

JUNIOR RODRIGUES DA SILVA

Grupo 18

GLOBAL SOLUTION - 2º SEMESTRE

Previsor de Perdas por Desastres Naturais

GITHUB: https://github.com/ana-ddomingues/gs_segsemestre.git

SÃO PAULO

2025

INTRODUÇÃO

Nos dias atuais, a natureza tem se mostrado cada vez mais imprevisível e intensa. Enchentes, secas prolongadas, ondas de calor e deslizamentos de terra impactam regiões inteiras, exigindo respostas rápidas, inteligentes e tecnológicas. Com base nesse cenário, a FIAP propôs como desafio da Global Solution 2025 o desenvolvimento de uma solução digital baseada em dados reais, capaz de prever, monitorar ou mitigar os impactos desses eventos extremos, aplicando conhecimentos integrados de lógica computacional, programação e estruturação de dados.

Nosso grupo optou por desenvolver um sistema inteligente que une aprendizado de máquina com sensores físicos simulados para a previsão de perdas causadas por desastres naturais.

DESENVOLVIMENTO

BANCO DE DADOS E MACHINE LEARNING:

Objetivo do Projeto:

O principal objetivo foi construir uma Rede Neural Artificial (MLP - Multi-Layer Perceptron) utilizando a linguagem Python, com o propósito de prever a ocorrência de eventos extremos ambientais. Isso permitirá a construção de cenários de impacto estimado, mesmo sem dados em tempo real.

Fonte dos Dados:

Utilizamos como base o conjunto de dados disponibilizado pelo Atlas Digital de Desastres no Brasil, compilado a partir do Sistema Integrado de Informações sobre Desastres (S2iD). Este dataset inclui:

- Mais de 70 variáveis por registro;
- Informações sobre:
 - Tipo e local do desastre;
 - Data de ocorrência;
 - Classificação do evento (climatológico, hidrológico, etc.);
 - Valores de prejuízos públicos e privados estimados (em R\$), por setor (educação, agricultura, transporte, etc.).

data	localidade	tipo_evento	precipitacao_mm	nivel_rio_m	umidade_solo_%	temperatura_C	evento_extremo
2023-05-30	Manaus	incêndio	118	0.67	53	41.8	não
2023-12-24	Manaus	enchente	26	1.94	23	40.1	não
2022-04-18	Porto Alegre	seca	123	8.05	56	29.5	não
2022-07-03	São Paulo	deslizamento	0	6.89	85	38.0	não
2022-01-07	Recife	deslizamento	15	0.84	77	28.2	não
2022-12-03	Recife	enchente	152	7.55	20	28.3	sim
2024-03-07	Porto Alegre	incêndio	68	6.89	97	21.1	não
2022-12-31	Manaus	calor_extremo	27	2.17	73	21.9	não
2024-01-04	São Paulo	seca	257	2.83	11	38.4	não
2023-08-02	Manaus	enchente	234	4.92	61	40.0	não

Show 10 per page

12345678910

Figura 1 - Visualização do Dataset.

Desenvolvimento da Rede Neural:

→ Importação de Bibliotecas

Para a criação e desenvolvimento da rede neural, utilizamos as seguintes bibliotecas:

```
# PASSO 2 - Importar bibliotecas
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import classification_report, confusion_matrix

# PASSO 3 - Carregar o dataset
nome_arquivo = list(uploaded.keys())[0] # Pega o nome do arquivo enviado
df = pd.read_csv(nome_arquivo)

# Verificar colunas e primeiras linhas
print("Colunas disponíveis:", df.columns.tolist())
print(df.head())
```

Figura 2 - Importação de bibliotecas e carregamento do dataset.

→ Geração de Variáveis e Divisão do Modelo

Aqui, geramos as variáveis necessárias e, em seguida, dividimos o modelo em conjuntos de treino e teste. Isso é crucial para avaliar o desempenho da rede neural em dados que ela "nunca viu" antes.

```
[26] # PASSO 6 - Separar variáveis preditoras e alvo

# Remover colunas irrelevantes ou não numéricas (exemplo: 'data')
colunas_para_remover = ["evento_extremo", "data"]
X = df.drop(columns=[col for col in colunas_para_remover if col in df.columns], errors='ignore')
y = df["evento_extremo"]

# Verificar se ainda há NaN em y
print("Quantidade de NaN em y (alvo):", y.isna().sum())

# PASSO 7 - Dividir dataset em treino e teste
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42)

# Confirmar que não existem NaN em y_train
print("NaN em y_train:", y_train.isna().sum())

# PASSO 8 - Treinar modelo de Rede Neural MLP

modelo = MLPClassifier(hidden_layer_sizes=(10, 5), max_iter=1000, random_state=42)
modelo.fit(X_train, y_train)
```

Quantidade de NaN em y (alvo): 0
NaN em y_train: 0

MLPClassifier

MLPClassifier(hidden_layer_sizes=(10, 5), max_iter=1000, random_state=42)

Figura 3 - Geração de variáveis e divisão do modelo.

→ Matriz de Confusão

A Matriz de Confusão é uma ferramenta de avaliação que mostra o desempenho do seu modelo de classificação. Ela detalha as previsões corretas e incorretas, categorizando-as em:

- Verdadeiros Positivos (VP): Predito como positivo e é realmente positivo.
- Verdadeiros Negativos (VN): Predito como negativo e é realmente negativo.
- Falsos Positivos (FP): Predito como positivo, mas é realmente negativo (erro Tipo I).
- Falsos Negativos (FN): Predito como negativo, mas é realmente positivo (erro Tipo II).

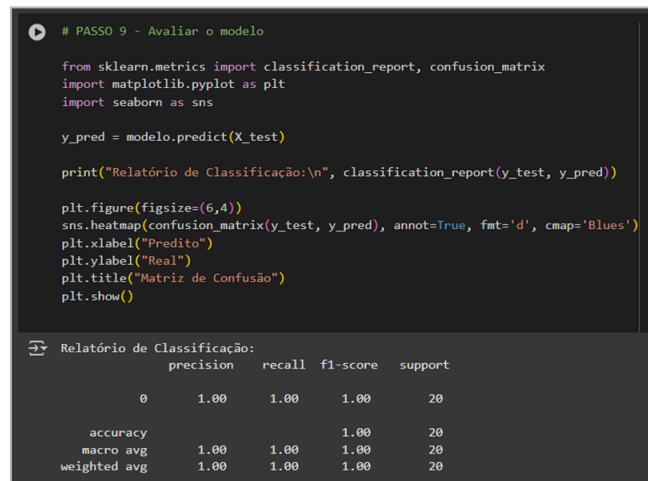


Figura 4 - Relatório de Classificação.

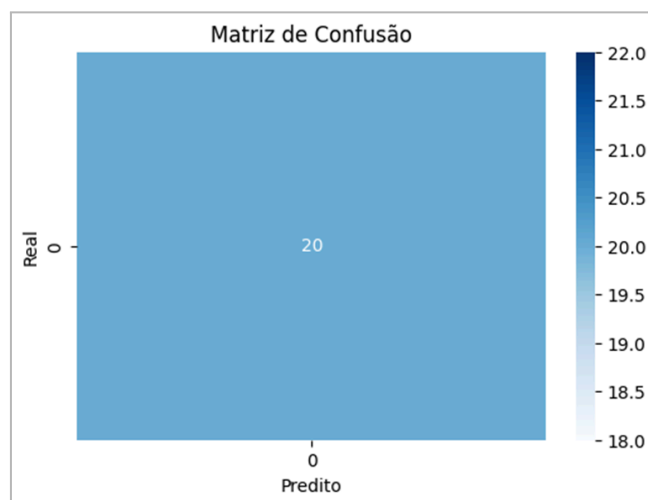


Figura 5 - Matriz de Confusão.

→ Mapa de Correlação

O Mapa de Correlação (ou Heatmap de Correlação) é uma visualização que mostra como cada variável do seu conjunto de dados se relaciona com as outras. Os valores de correlação variam de:

- +1: Correlação positiva perfeita (as variáveis aumentam ou diminuem juntas).
- 0: Sem correlação (as variáveis não têm relação linear clara).
- -1: Correlação negativa perfeita (quando uma variável aumenta, a outra diminui, e vice-versa).

As cores no mapa ajudam a visualizar a força e a direção da correlação:

- Vermelho: Indica uma forte correlação positiva.
- Azul Escuro: Indica uma forte correlação negativa.
- Branco: Indica ausência de correlação.

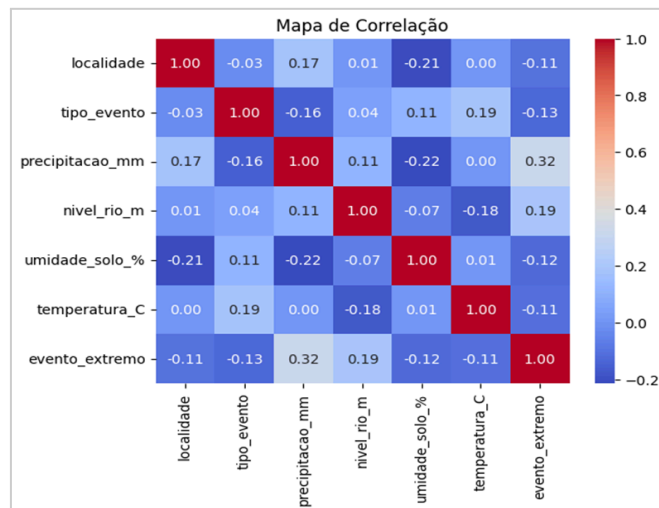


Figura 6 - Mapa de Correlação.

Variável	Correlação com evento_extremo	Interpretação
precipitacao_mm	0.32 (positiva moderada)	Aumentos de chuva estão associados com maior chance de evento extremo.

nivel_rio_m	0.19	O nível do rio também pode indicar risco de eventos, como enchentes.
tipo_evento	-0.13	Baixa correlação, mas pode ajudar se codificado corretamente.
umidade_solo_%	-0.12	Pode ter influência em eventos como seca ou deslizamento.
temperatura_C	-0.11	Baixa correlação direta, mas pode combinar com outras variáveis.
localidade	-0.11	Muito fraca, talvez mais útil como agrupamento/região.

Principais insights:

- precipitacao_mm é a variável mais correlacionada com evento_extremo, o que faz sentido (muita chuva → enchente ou deslizamento).
- nivel_rio_m também é relevante, indicando que o modelo pode usar isso como sinal de risco.
- Variáveis como temperatura_C e umidade_solo_% têm correlação baixa, mas ainda podem ser úteis em combinação com outras (não devem ser descartadas sem testes).
- localidade tem correlação negativa fraca, pode ser melhor tratada como agrupamento ou dummy variable (variáveis categóricas).

Valor Prático:

- Ferramenta útil para tomada de decisão preventiva e dimensionamento de ajuda humanitária.
 - Pode ser adaptada por prefeituras, Defesa Civil e pesquisadores.
 - Apoia visualmente o entendimento de risco a partir de variáveis simples.
 - Permite reforçar os aprendizados de programação, lógica e modelagem estatística.
-

SIMULAÇÃO DE AMBIENTE COM SENSORES DO WOWKI

Além da modelagem de dados, desenvolvemos um protótipo funcional no simulador Wokwi, usando o microcontrolador ESP32 para simular uma estação climática inteligente que monitora, detecta e avisa sobre condições de risco de desastres naturais, como enchentes e incêndios florestais, com base em sensores de baixo custo e lógica embarcada.

Sensores e Componentes Utilizados:

- DHT22: mede temperatura e umidade.
- HC-SR04: simula sensor ultrassônico para medir o nível da água (possíveis enchentes).
- LDR (fotoresistor): mede luminosidade, associada à radiação solar e risco de seca/incêndio.
- Botão: simula chuva intensa repentina.
- LEDs (vermelho, amarelo e verde): alertas visuais sobre o estado do ambiente.

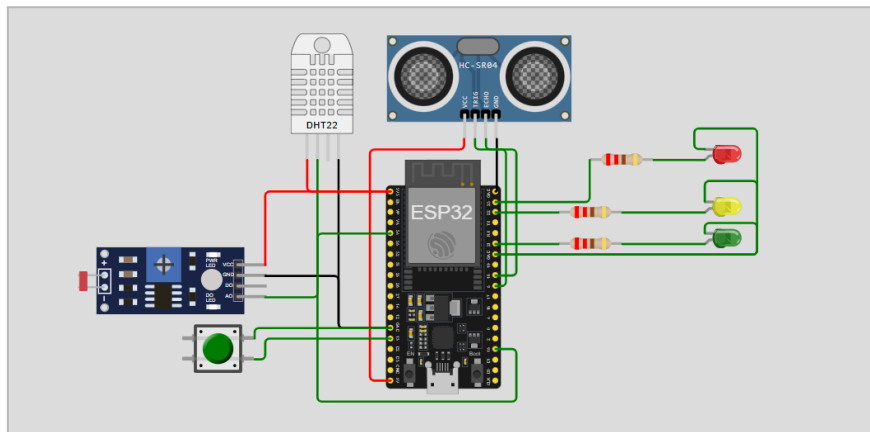


Figura 7 – Montagem dos sensores no simulador Wokwi.

Justificativa dos LEDs

Utilizamos três LEDs simples:

- Verde: ambiente em condições normais

- Amarelo: atenção (nível elevado de água ou baixa luminosidade)
- Vermelho: risco crítico (alta temperatura, seca extrema ou botão de emergência ativado)

Temp: 36.10 C	Hum: 39.50 %	Dist: 298.89 cm	LDR: 2991	Emergência: NÃO
Temp: 36.10 C	Hum: 39.50 %	Dist: 298.96 cm	LDR: 2991	Emergência: NÃO
Temp: 36.10 C	Hum: 39.50 %	Dist: 298.89 cm	LDR: 2991	Emergência: NÃO
Temp: 4.70 C	Hum: 75.50 %	Dist: 298.93 cm	LDR: 2991	Emergência: NÃO
Temp: 4.70 C	Hum: 75.50 %	Dist: 298.89 cm	LDR: 2991	Emergência: NÃO
Temp: 4.70 C	Hum: 75.50 %	Dist: 298.93 cm	LDR: 2991	Emergência: SIM
Temp: 4.70 C	Hum: 75.50 %	Dist: 298.89 cm	LDR: 2991	Emergência: NÃO
Temp: 40.30 C	Hum: 21.50 %	Dist: 298.91 cm	LDR: 2991	Emergência: NÃO
Temp: 40.30 C	Hum: 21.50 %	Dist: 298.93 cm	LDR: 2991	Emergência: NÃO
Temp: 40.30 C	Hum: 21.50 %	Dist: 160.99 cm	LDR: 1262	Emergência: NÃO
Temp: 40.30 C	Hum: 21.50 %	Dist: 160.97 cm	LDR: 1262	Emergência: SIM

Figura 8 - Exemplo das leituras no Monitor Serial do Wokwi. Os dados exibidos incluem temperatura, umidade, distância detectada (nível de água), luminosidade e o status do botão de emergência.

A simulação demonstrou que o sistema é capaz de monitorar múltiplos fatores ambientais de forma simultânea e fornecer alertas visuais simples e eficientes. Isso valida a proposta de um protótipo funcional e educativo, com potencial para aplicação real com sensores físicos e integração com bancos de dados ou plataformas de visualização.

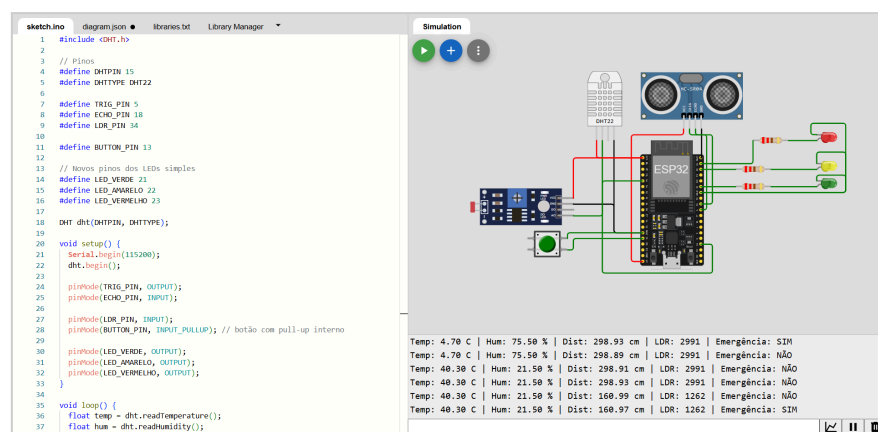


Figura 9 - Simulação final no Wokwi, com os sensores conectados, o Monitor Serial ativo e parte do código visível.

RESULTADOS ESPERADOS

- Um sistema de monitoramento ambiental que simula condições de desastre em tempo real usando sensores físicos;
- Um modelo preditivo funcional capaz de estimar o prejuízo total com base em dados históricos;
- Integração das saídas dos sensores com alertas visuais fáceis de interpretar;
- Protótipo útil para testes futuros com sensores reais em ambientes escolares ou simuladores;

CONCLUSÃO

A solução proposta conecta dados históricos com sensores físicos simulados, proporcionando uma abordagem prática e didática para compreender os impactos dos desastres naturais. O sistema é flexível, pode ser adaptado a diferentes regiões e cenários, e cumpre com o objetivo da Global Solution 2025 de aliar tecnologia, lógica computacional e impacto social.

Futuramente, a integração com plataformas em nuvem, dashboards interativos e APIs de clima em tempo real pode tornar essa ferramenta ainda mais poderosa.

Para assistir o vídeo da execução do projeto, acesse este link:

https://youtu.be/RpS_HxwNmxk