

✓ Projeto Alura - Análise de Risco de Chuva

Introdução

Este notebook Jupyter demonstra a utilização do Google Earth Engine (GEE) e do Google Generative AI para realizar a análise de risco de chuva com base no volume precipitado em uma área específica. O código interage com um modelo de linguagem para obter instruções e fornecer resultados de forma estruturada.

Importante

- Este é um projeto didático construído em uma imersão de tecnologia, com integração das plataformas de inteligência artificial e de dados espaciais do Google, e **em hipótese alguma deve ser utilizado para fins de informação ou operacionais**.
- Os resultados da análise não foram validados e podem não representar a realidade, variando acordo com a qualidade dos dados disponíveis.
- É de suma importância consultar os órgãos oficiais e especialistas na área, que prestam esse tipo de serviço, como o CEMADEN e o INMET.

Funcionamento

O usuário fornece informações:

- Cidade a ser pesquisada
- Estado a ser pesquisado
- Data da pesquisa

O informações são repassadas ao modelo:

- A precipitação média anual dos estados do Brasil
- Explica a escala de risco de chuvas utilizada

O modelo, com base no informado, fornece as seguintes informações:

- A chuva média mensal para o estado informado
- A longitude do local informado
- A latitude do local informado

O modelo obtém dados do GEE:

- Carrega o dataset de bacias hidrográficas WWF/HydroATLAS/v1/Basins/level07
- Filtra a bacia hidrográfica para o local informado
- Carrega o dataset de precipitação NASA/GPM_L3/IMERG_V06
- Filtra a precipitação para a data informada
- Calcula a precipitação média na bacia hidrográfica

O modelo analisa os dados e gera o resultado:

- Compara o valor da precipitação observada com a média mensal
- Determina o nível de risco de perigo com base na escala
- Indica as ações recomendadas para a população local

O modelo apresenta o resultado ao usuário:

- Exibe o resultado em formato Markdown estruturado
- Inclui informações sobre o local, data, análise e ações recomendadas

Benefícios

- Análise precisa e atualizada: O GEE fornece dados de precipitação em tempo com alta resolução temporal, razoável resolução espacial e com 2 dias de latência.
- Interpretação facilitada: O modelo de linguagem auxilia na interpretação dos dados e na comunicação dos resultados de forma clara e concisa.

Possíveis aplicações

- Órgãos públicos: Auxiliar na gestão de recursos hídricos e na elaboração de planos de contingência para inundações.
- Defesa Civil: Monitorar áreas de risco e emitir alertas à população em caso de risco iminente de inundaçõa.
- Empresas: Avaliar riscos para infraestrutura e operações em áreas suscetíveis a inundações.
- Pesquisadores: Realizar estudos sobre a dinâmica da precipitação e os impactos das inundações.

Instruções de Uso

Siga as instruções do modelo:

- Responda às perguntas do modelo com as informações solicitadas.
- Forneça dados precisos e atualizados.
- Siga as instruções para a interpretação dos resultados.

O modelo fornecerá o resultado:

- A análise do risco de inundação para a área informada.
- Informações sobre o local, data, precipitação e nível de risco.
- Ações recomendadas para a população local.

```
# Importando as bibliotecas que serão utilizadas
from IPython.display import display, Markdown
from google.colab import userdata
import ee
import geemap
import textwrap
```



```
# Autenticando o Google Earth Engine (GEE)
ee.Authenticate()
```



```
# Abrindo a biblioteca de projetos do GEE
ee.Initialize(project=userdata.get('GEE_PROJECT'))
```



```
# Instalando a API do Google Gen AI
!pip install -q -U google-generativeai
```



```
# Importando o SDK do Python para Google Gen AI
import google.generativeai as genai
```

```
genai.configure(api_key=userdata.get('GOOGLE_API_KEY'))
```



```
# Função para deixar o texto do modelo mais apresentável para o notebook
def to_markdown(text):
    text = text.replace(' * ', ' > ')
    return Markdown(textwrap.indent(text, '> ', predicate=lambda x: True))
```

```
# Função para remover os símbolos que saem na resposta do modelo
def remover_simbolos(texto):
    texto = texto.strip('**') # Remove os asteriscos
    texto = texto.strip('")') # Remove as aspas
    return texto
```



```
# Inicializando o modelo
generation_config = {
    'candidate_count': 1,
    'temperature': 0.1,
}
```



```
# Escolhendo o modelo
model = genai.GenerativeModel(model_name='gemini-1.0-pro', generation_config=generation_config)
```



```
# Abrindo a instância do chat do modelo
chat = model.start_chat(history=[])
```



```
# Chuva Média Mensal por Estado do Brasil (mm) (tabela obtida no Gemini)
precipitacao_media_mensal = {
    "Acre": 2120/12,
    "Alagoas": 1100/12,
    "Amazonas": 2820/12,
    "Amapá": 3290/12,
    "Bahia": 1430/12,
    "Ceará": 2470/12,
    "Distrito Federal": 1920/12,
    "Espírito Santo": 1410/12,
    "Goiás": 1900/12,
    "Maranhão": 2470/12,
    "Mato Grosso": 2230/12,
    "Mato Grosso do Sul": 2310/12,
    "Minas Gerais": 1900/12,
    "Pará": 2820/12,
    "Paraíba": 1240/12,
    "Paraná": 1480/12,
    "Pernambuco": 1290/12,
    "Piauí": 1080/12,
    "Rio de Janeiro": 1350/12,
    "Rio Grande do Norte": 1180/12,
    "Rio Grande do Sul": 1530/12,
    "Rondônia": 2200/12,
    "Roraima": 1770/12,
    "Santa Catarina": 1670/12,
    "São Paulo": 1260/12,
    "Sergipe": 1690/12,
    "Tocantins": 1820/12
}
```



```
# Definindo as instruções do modelo
instrucao_1 = 'Você é um assistente para obtenção de dados geográficos e de precipitação para a definição de perigo com relação'
instrucao_2 = 'A seguir, segue a tabela de precipitação média anual dos estados do Brasil, posteriormente será requisitada essa'
instrucao_3 = 'A escala de risco para chuva é a seguinte: Risco amarelo (Perigo Potencial): volume de chuva até 50 mm/dia, Risc'
instrucao_4 = 'Amarelo significa Perigo Potencial, ou seja, os cidadãos devem ter "cuidado na prática de atividades sujeitas a'
instrucao_5 = 'O resultado final deve ser apresentado na seguinte estrutura: **Local**: Cidade, Estado \n**Data**: Data \n**Ano': Ano
```



```
# Enviando as instruções para o modelo
chat.send_message(instrucao_1)
chat.send_message(instrucao_2)
chat.send_message(instrucao_3)
chat.send_message(instrucao_4)
chat.send_message(instrucao_5)
```



```
response:
GenerateContentResponse(
    done=True,
    iterator=None,
    result=glm.GenerateContentResponse({'candidates': [{"content": {"parts": [
        {"text": "***Estrutura do Resultado Final:**\n\n**Local:** Cidade, Estado\n**Data:**\n\n**Análise Inicial:**\n\n* Comparação do valor da precipitação na data com a\nmédia mensal\n\n**Análise do Risco:**\n\n* Nível de risco de acordo com a escala de\nrisco de chuva\n\n* Ações que devem ser tomadas pela população\n\n**Pessoas Afetadas:**\n\n* Número estimado de pessoas possivelmente afetadas, com base na\npopulação do município\n\n**Exemplo:**\n\n* Local:** São Paulo, São Paulo\n\n**Data:**\n\n10 de janeiro de 2023\n\n**Análise Inicial:**\n\n* A precipitação registrada em 10 de\njaneiro de 2023 foi de 80 mm.\n\n* A média mensal de precipitação para janeiro em São\nPaulo é de 250 mm.\n\n**Análise do Risco:**\n\n* Nível de risco: Amarelo (Perigo\nPotencial)\n\n* Ações que devem ser tomadas pela população\n\n* Tenha cuidado ao\n\n'}}]}})
```

✓ Avaliação do perigo relacionado ao volume precipitado no Brasil

```
cidade = input('Informe a cidade que será a pesquisa\n')
print()
estado = input('Informe o estado que será a pesquisa\n')

local = chat.send_message('O local da pesquisa é na cidade de 'f'{cidade}' 'e no estado 'f'{estado}').text
populacao = chat.send_message('Qual é a população estimada para o local: 'f'{local}').text
```

➡ Informe a cidade que será a pesquisa
Santa Maria

Informe o estado que será a pesquisa
RS

```
data = input('Informe o dia da pesquisa no formato DD/MM/AAAA \n')
```

```
# Data no formato do Google Earth Engine
```

```
data_ee = ee.Date(data[-4:] + '-' + data[3:5] + '-' + data[0:2])
```

➡ Informe o dia da pesquisa no formato DD/MM/AAAA
30/04/2024

```
# Obtendo o valor da chuva média mensal para o estado informado
```

```
chuva_media_estado = chat.send_message('Dado a tabela de chuva média mensal por estados, qual é a chuva média mensal para o estado informado?')  
chuva_media_estado = float(remover_simbolos(chuva_media_estado))  
print('Chuva média mensal do \'f'{estado}\': f'{chuva_media_estado}')
```

➡ Chuva média mensal do RS: 150.0

```
# Obtendo a longitude do local informado
```

```
lon = chat.send_message('Com base na cidade de \'f'{cidade}\'', localizada no estado do \'f'{estado}\'', qual é a longitude do local informado?')  
lon = float(remover_simbolos(lon))
```

```
# Obtendo a latitude do local informado
```

```
lat = chat.send_message('Com base na cidade de \'f'{cidade}\'', localizada no estado do \'f'{estado}\'', qual é a latitude do local informado?')  
lat = float(remover_simbolos(lat))
```

```
# Definindo o ponto para obtenção das informações no Google Earth Engine
```

```
ponto = ee.Geometry.Point([lon, lat])
```

➡

```
# Definindo a bacia hidrográfica do local informado
```

```
# Adicionando o título do mapa
```

```
display(Markdown("## Mapa de localização da bacia hidrográfica da área de estudo: \"f'{cidade}, {estado}'"))
```

```
# Adicionando o mapa formato geemap
```

```
Map = geemap.Map(basemap='Esri.WorldImagery')  
Map.add_basemap('CartoDB.PositronOnlyLabels')
```

```
# Adicionando as bacias do banco de dados HydroATLAS - HydroSHEDS
```

```
bacias = ee.FeatureCollection('WWF/HydroATLAS/v1/Basins/level07')
```

```
# Filtrando a bacia para o local informado
```

```
bacia_filtrada = bacias.filterBounds(ponto)
```

```
# Configurando a visualização da bacia filtrada
```

```
style_params = {  
    'fillColor': '00000000', # 100% transparent fill color  
    'color': '#FF00FF', # Red color for the border  
    'width': 5 # Thicker border  
}
```

```
# Definindo a legenda do mapa
```

```
legend_dict = {  
    'Bacia Hidrográfica': '#FF00FF',  
}
```

```
# Adicionando a legenda ao mapa
```

```
Map.add_legend(legend_dict=legend_dict)
```

```
# Adicionando a bacia hidrográfica ao mapa
```

```
Map.addLayer(bacia_filtrada.style(**style_params), {}, 'Bacia hidrográfica')
```

```
# Centralizando o mapa
```

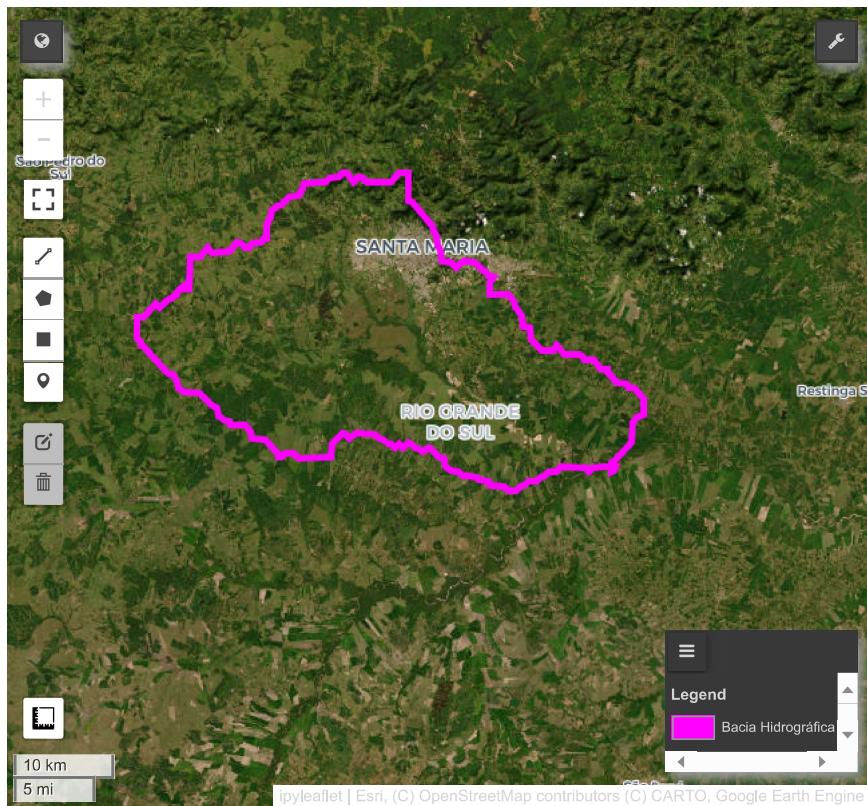
```
Map.centerObject(bacia_filtrada, zoom=10)
```

```
# Mostrando o mapa
```

```
Map
```



Mapa de localização da bacia hidrográfica da área de estudo: Santa Maria, RS



```
# Obtendo informações de precipitação do local para a data informada

# Adicionando o título do mapa
display(Markdown("### Mapa de precipitação acumulada para o dia " f"{data}"))

# Carregando os dados de precipitação do produto Global Precipitation Measurement
gpm = ee.ImageCollection('NASA/GPM_L3/IMERG_V06').select('precipitationCal')

# Ajustando a data
range = ee.Date(data_ee).getRange('day');
gpm = gpm.filter(ee.Filter.date(range));

# Obtendo o valor de precipitação diária - é dividido por 2 pois o valor é registrado a cada 30 minutos (em mm/h)
total_precipitation = gpm.reduce(ee.Reducer.sum()).divide(2)

# Criando uma máscara para aparecer apenas os pixels em que houve chuva registrada no período
mask = total_precipitation.gt(0.5);
precipitation = total_precipitation.updateMask(mask);

# Criando o mapa
Map = geemap.Map(basemap='CartoDB.DarkMatter')

# Definindo a legenda
legend_dict = {
    '0-30': '#000096',
    '30-60': '#0064ff',
    '60-90': '#00b4ff',
    '90-120': '#33db80',
    '120-150': '#9beb4a',
    '150-180': '#ffeb00',
    '180-210': '#ffb300',
    '210-240': '#ff6400',
    '240-270': '#eb1e00',
    '270-300': '#af0000',
}

# Adicione a legenda ao mapa
Map.add_legend(legend_dict=legend_dict)

# Adiciona o total precipitado no mapa
Map.addLayer(precipitation, {'min': 0, 'max': 300, 'palette': ['000096','0064ff', '00b4ff', '33db80', '9beb4a', 'ffeb00', 'ffb30']})

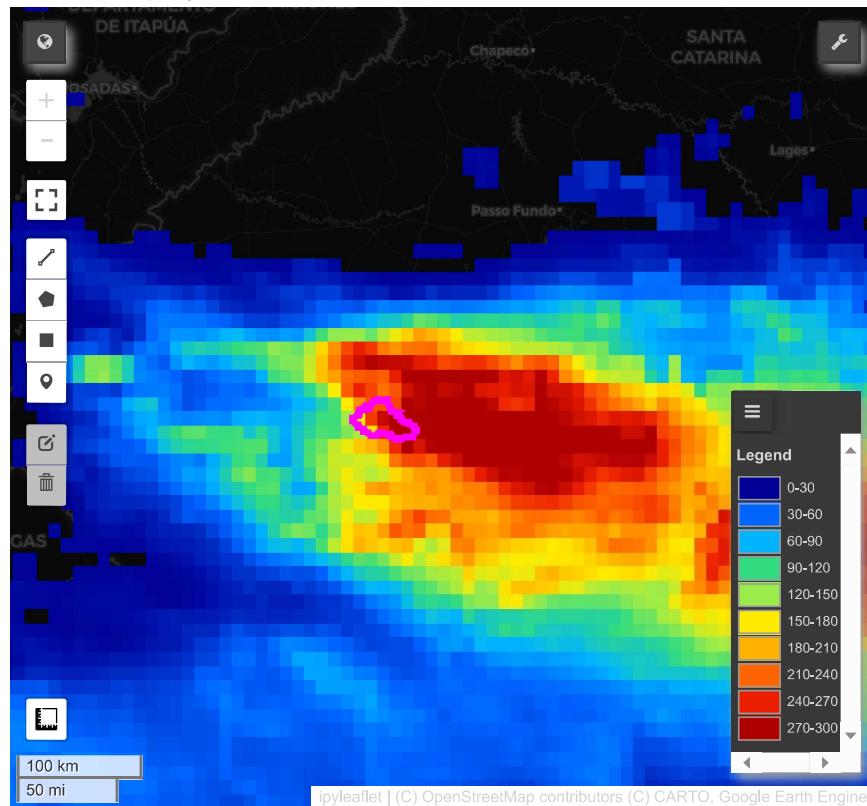
# Adiciona a bacia hidrográfica no mapa
Map.addLayer(bacia_filtrada.style(**style_params), {}, 'Bacia hidrográfica')
```

```
# Centralizando o mapa
Map.centerObject(bacia_filtrada, zoom=7)
```

```
# Mostrando o mapa
Map
```



Mapa de precipitação acumulada para o dia 30/04/2024



```
# Obtendo informações sobre a chuva observada média na bacia hidrográfica
chuva_observada = precipitation.reduceRegion(reducer=ee.Reducer.mean(), geometry=bacia_filtrada, scale=30). getInfo()['precipitation']
```

```
# Verificação do valor precipitado:
```

```
if chuva_observada == None:
    print('Não houve chuva')
```

```
else:
```

```
    print('Precipitação média da bacia hidrográfica no dia 'f'{data}':', round(chuva_observada, 2), 'mm')
```

```
→ Precipitação média da bacia hidrográfica no dia 30/04/2024: 263.77 mm
```

```
# Resultado da análise realizada
```

```
resultado = chat.send_message('Com base no valor precipitado de 'f'{chuva_observada}', em 'f'{cidade}', 'f'{estado}' no dia '
```

```
→
```

```
# Apresentando o resultado
```

```
display(Markdown('### Análise de risco devido a precipitação'))
```

```
display(to_markdown(resultado))
```

```
display(Markdown('### Importante!!'))
```

```
display(Markdown('A presente análise **não deve ser levado em conta por ninguém**.'))
```

```
display(Markdown('Esse código foi criado como parte de um desafio da Imersão de Inteligência Artificial da Alura.'))
```

```
display(Markdown('Portanto, os resultados aqui apresentados não foram validados, servem apenas para fins didáticos.'))
```



Análise de risco devido a precipitação

Análise do Risco de Precipitação

Local: Santa Maria, Rio Grande do Sul Data: 30/04/2024 Valor da Precipitação: 263,77 mm

Análise Inicial:

- A precipitação registrada em Santa Maria em 30/04/2024 foi de 263,77 mm.
- A média mensal de precipitação para abril em Santa Maria é de 150 mm.

Análise do Risco:

- Nível de risco: **Vermelho (Grande Perigo)**
- Ações que devem ser tomadas pela população:

Mantenha-se informado sobre as condições meteorológicas previstas e os possíveis riscos. Siga as instruções e conselhos das autoridades em todas as circunstâncias e prepare-se para medidas de emergência.

Pessoas Afetadas:

- População de Santa Maria: 282.811 habitantes
- Número estimado de pessoas possivelmente afetadas: 282.811 (toda a população da cidade)

Observações:

- O nível de risco e as ações recomendadas podem variar dependendo das condições locais, como topografia, cobertura do solo e infraestrutura.
- É importante seguir as orientações das autoridades locais e estar preparado para tomar medidas de emergência, como evacuar sua casa ou procurar abrigo, se necessário.

Importante!!

A presente análise **não deve ser levado em conta por ninguém**.

Esse código foi criado como parte de um desafio da Imersão de Inteligência Artificial da Alura.

Portanto, os resultados aqui apresentados não foram validados, servem apenas para fins didáticos.