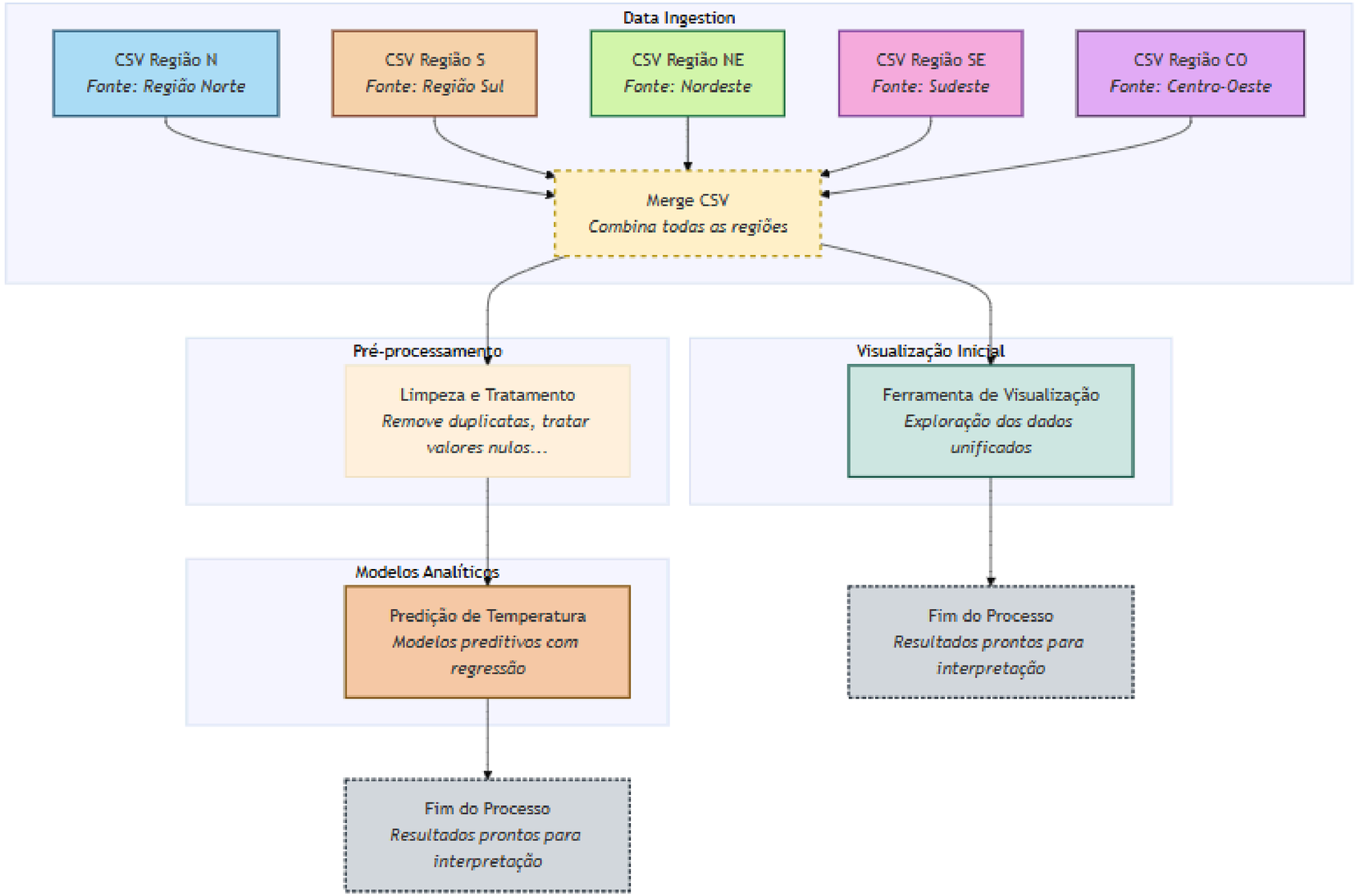
Dask

PROJETO FINAL: ANÁLISE DE DADOS METEREOLÓGICOS BRASILEIROS

Tópicos

- Arquitetura
- Análise Exploratória
- Gráficos e discussão
- Regressão

Arquitetura



Dask

Análise Exploratória dos Dados

Análise da Temperatura Média anual de 2000 a 2021:

- Por região (Norte, Sul, Sudeste, Nordeste e Centro-Oeste)

Análise da Precipitação Média anual de 2000 a 2021:

- Por região (Norte, Sul, Sudeste, Nordeste e Centro-Oeste)

Análise de Temperatura Média e Precipitação Total de todos os anos (2000-2021):

- Por estados do Brasil
- Por estações do ano

Dask

Fluxo Principal da Análise Exploratória

- **Leitura e Concatenação**
- **Limpeza de Dados**
- **Enriquecimento Temporal:**
- **Agregações principais**
- **Geração de Gráficos**

Dask

Agregações principais

```python
def aggregate_annual_precipitation_by_region(df):
    agg = df.groupby(["regiao", "ano"])[COL_PREC].sum().rename
    ("precipitacao_total_anual")
    return agg.compute().reset_index()
```

```python
def aggregate_by_state(df):
    agg = df.groupby(COL_STATE)[
        [COL_TEMP_MAX, COL_TEMP_MIN, COL_PREC]
    ].agg({
        COL_TEMP_MAX: "mean",
        COL_TEMP_MIN: "mean",
        COL_PREC: "sum"
    }).rename(columns={
        COL_TEMP_MAX: "temp_max_media",
        COL_TEMP_MIN: "temp_min_media",
        COL_PREC: "precipitacao_total"
    })
    return agg.compute().reset_index()
```

Dask

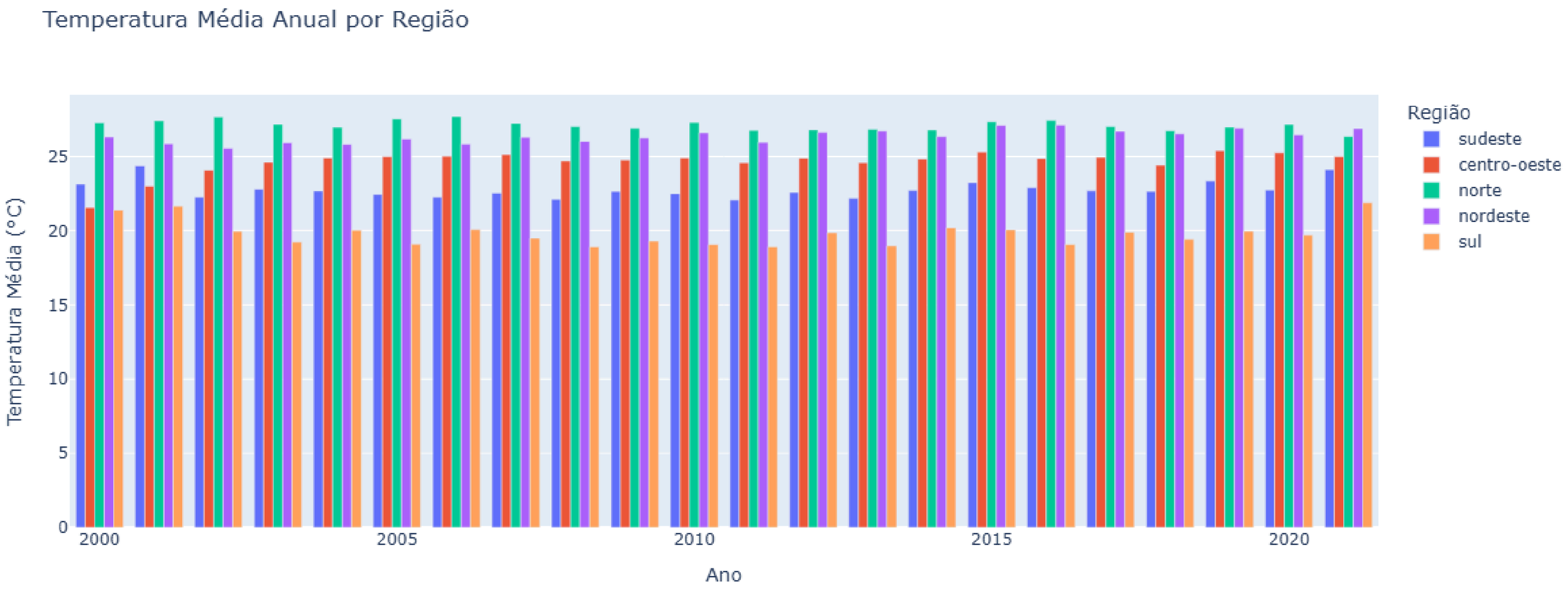
Fluxo Principal da Análise Exploratória

```python
def aggregate_annual_temperature_by_region(df):
    agg = df.groupby(["regiao", "ano"])[[COL_TEMP_MAX]].agg({
        COL_TEMP_MAX: "mean"
    }).rename(columns={
        COL_TEMP_MAX: "temp_media"
    })
    return agg.compute().reset_index()
```

```python
def aggregate_by_season(df):
    agg = df.groupby("estacao")[[COL_TEMP_MAX, COL_TEMP_MIN, COL_PREC]].agg({
        COL_TEMP_MAX: "mean",
        COL_TEMP_MIN: "mean",
        COL_PREC: "sum"
    }).rename(columns={
        COL_TEMP_MAX: "temp_max_media",
        COL_TEMP_MIN: "temp_min_media",
        COL_PREC: "precipitacao_total"
    })
    return agg.compute().reset_index()
```

Temperatura Média anual de 2000 a 2021 por região

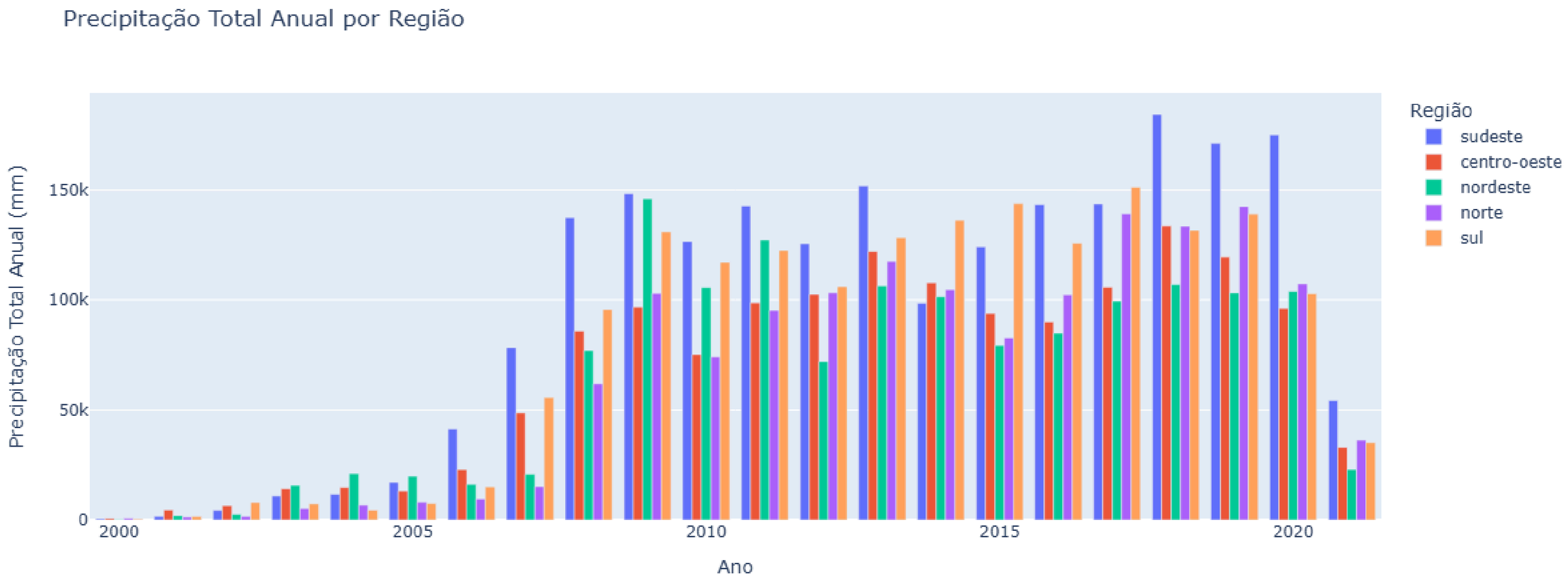
Dask



https://lucasnizio.github.io/grafico-html-bigdata/grafico_temperatura_media_anual_por_regiao.html

Precipitação Total anual de 2000 a 2021 por região

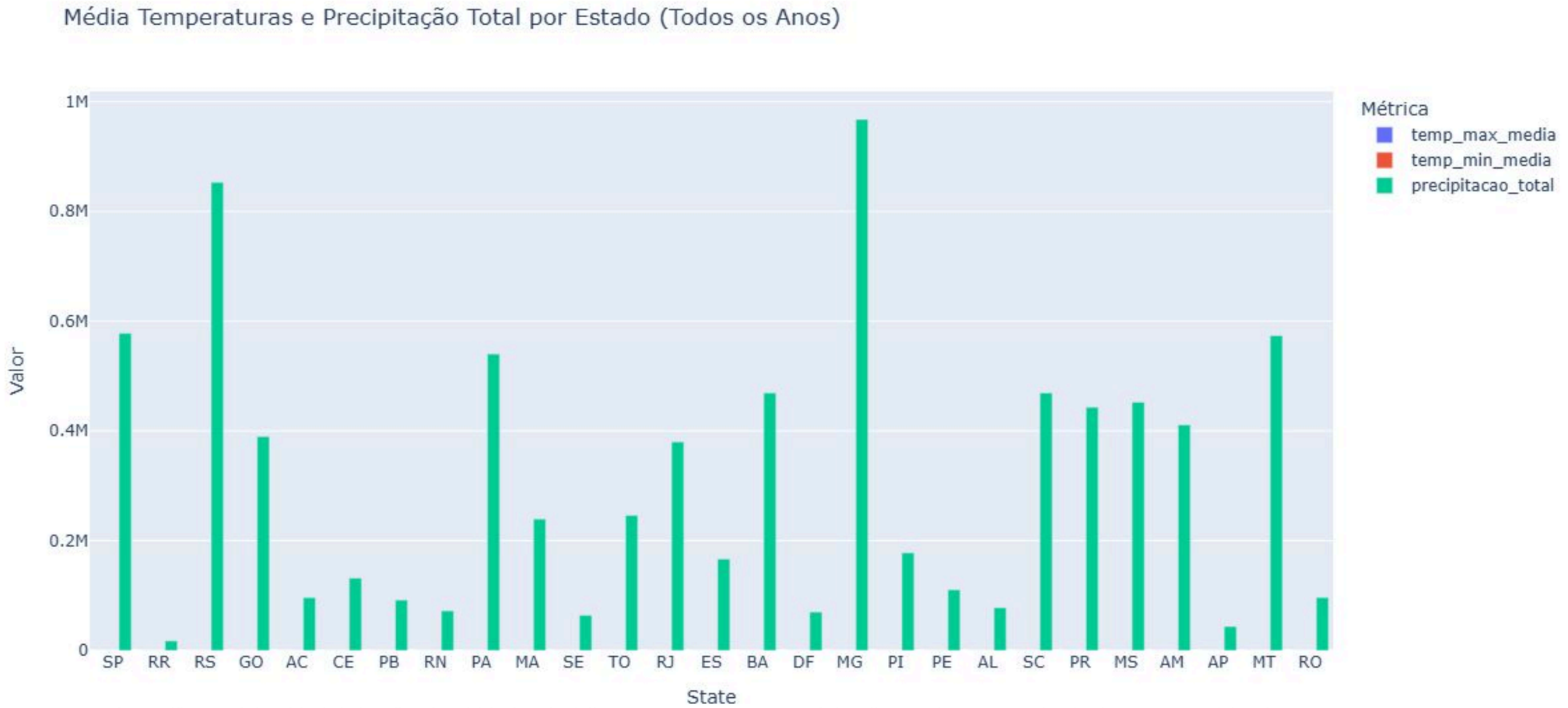
Dask



https://lucasnizio.github.io/grafico-html-bigdata/grafico_precipitacao_anual_por_regiao.html

Temperatura Máxima e Mínima Média e Precipitação Total de todos os anos (2000-2021) por estados do Brasil

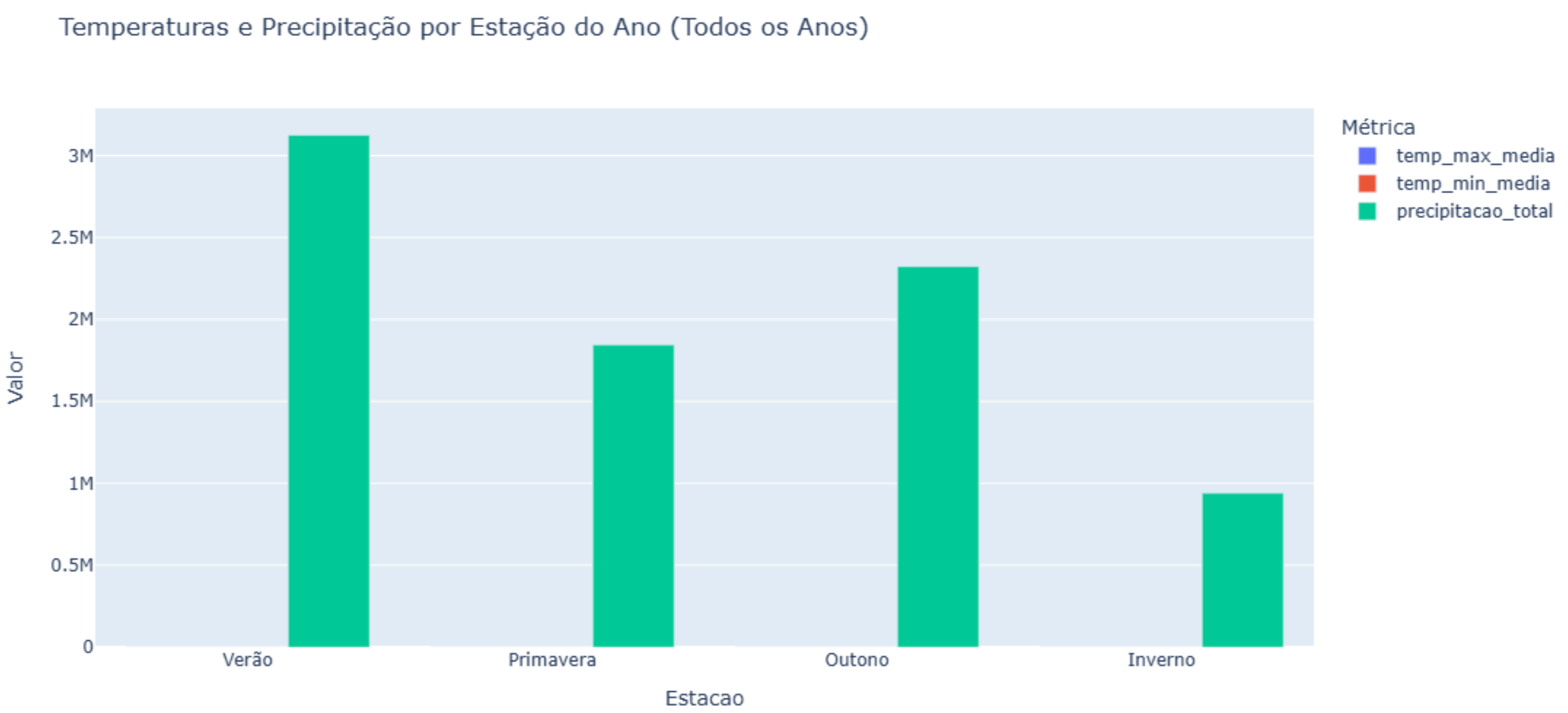
Dask



https://lucasnizio.github.io/grafico-html-bigdata/média_temperaturas_e_precipitação_total_por_estado_(todos_os_anos).html

Temperatura Máxima e Mínima Média e Precipitação Total de todos os anos (2000-2021) por estações do ano

Dask



https://lucasnizio.github.io/grafico-html-bigdata/temperaturas_e_precipitação_por_estação_do_ano_(todos_os_anos).html

Dask

Regressão da temperatura do ar

- **Objetivo:** Criação de um modelo de regressão para predizer o valor da coluna: “TEMPERATURA DO AR - BULBO SECO, HORARIA (°C)”, utilizando os outros atributos presentes na base de dados.
- **Metodologia:** Utilizou-se Dask, DaskML, DaskDataframe e a biblioteca LightGBM.
- **Leitura e Limpeza(Dask e DaskDataframe):** Feita com Dask e DaskDataframe, fizemos a leitura dos dados, remoção de colunas relacionadas a outras medidas de temperatura, filtragem de valores indesejados, codificação de valores categóricos para valores numéricos.

Dask

Regressão da temperatura do ar

- **Treinamento e avaliação do modelo(DaskML e LightGBM):** Inicialmente, foi realizado uma divisão dos dados, em treino e teste com a intenção de treinar o modelo na primeira e avalia-lo, usando a segunda.
- **Escolha do modelo:** Por conta do tamanho do dataset, optamos por um modelo mais robusto, que fosse capaz de identificar com maior facilidade os padrões no grande número de dados.
- **Avaliação do modelo:** Feita com a métrica erro médio absoluto, que fornece um valor facilmente interpretável.

Dask

Resultados da Regressão

- **Máquina utilizada:**
  - **Processador:** 13th Gen Intel(R) Core(TM) i7-13700F
  - **Memória:** 64GB de RAM
- **Hiperparâmetros do LGBM:**
  - **n_estimators:** 1000
  - **learning_rate:** 0.01

- **Tempo de Treinamento:** 23 minutos e 52 segundos
- **Tempo para predição:** 442 milisegundos
- **Erro médio absoluto obtido:** 1.45

Dask

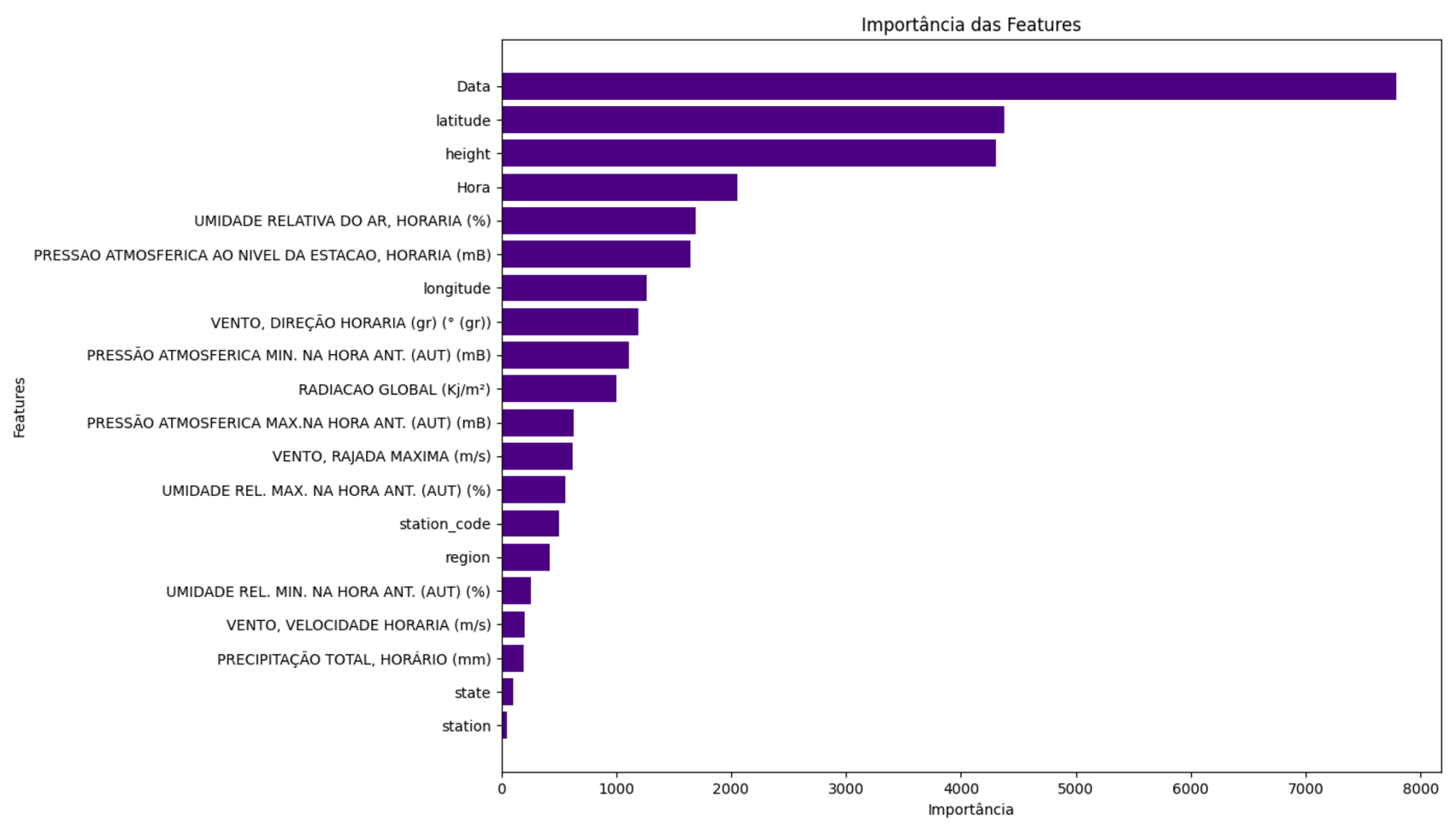
Resultados da Regressão

Amostra de Resultados da Predição

Valores Reais	Valores Preditos	Erro absoluto
26,3000	27,7195	1,4195
26,9000	26,4218	0,4782
10,5000	15,7538	5,2538
25,6000	26,5004	0,9004
31,2000	30,8340	0,3660
17,5000	20,1653	2,6653
29,5000	29,1203	0,3797
20,7000	20,3536	0,3464
23,0000	22,3361	0,6639
19,5000	17,9173	1,5827

Dask

Resultados da Regressão



Dask

Dockerfile

```
FROM ghcr.io/dask/dask-notebook:latest

USER root

RUN apt-get update \
 && apt-get install -y --no-install-recommends \
      build-essential \
      cmake \
      wget \
 && rm -rf /var/lib/apt/lists/*

USER jovyan
WORKDIR /home/jovyan

RUN pip install --no-cache-dir \
      dask-ml \
      lightgbm \
      scikit-learn \
      pandas \
      numpy \
      matplotlib

RUN mkdir -p /home/jovyan/work
WORKDIR /home/jovyan/work

EXPOSE 8888 8786 8787
```

Dask

Inicialização do docker swarm

```
if ! docker info | grep -q 'Swarm: active'; then
  echo "Initializing Docker Swarm..."
  docker swarm init
else
  echo "Swarm já ativo."
fi

echo "Deploying stack '$STACK_NAME'..."
docker stack deploy -c $COMPOSE_FILE $STACK_NAME

echo "Deployment completo!"

echo "Acesse JupyterLab: http://localhost:8888"
```

docker

Dask

docker-compose

```yaml
jupyter:
  image: custom-dask-notebook:latest
  ports:
    - "8888:8888"
  environment:
    - JUPYTER_ENABLE_LAB=yes
    - JUPYTER_TOKEN=token
  volumes:
    - /home/fundecc/TrabalhoFinalBigData/bin:/home/jovyan/work/bin
    - /home/fundecc/TrabalhoFinalBigData/datasample:/home/jovyan/work/datasample
    - /home/fundecc/TrabalhoFinalBigData/misc:/home/jovyan/work/misc
    - /home/fundecc/TrabalhoFinalBigData/src:/home/jovyan/work/src
  depends_on:
    - dask-scheduler
  networks:
    - dask-net
```

Dask

docker-compose

```yaml
services:
  dask-scheduler:
    image: custom-dask-notebook:latest
    command: dask-scheduler
    ports:
      - "8786:8786"
      - "8787:8787"
    networks:
      - dask-net
    healthcheck:
      test: ["CMD", "curl", "--fail", "http://localhost:8787/status"]
      interval: 10s
      timeout: 3s
      retries: 5
```

```yaml
dask-worker:
  image: custom-dask-notebook:latest
  deploy:
    replicas: 6
  command: >
    dask-worker tcp://dask-scheduler:8786
    --nthreads 4
    --memory-limit 10GB
  networks:
    - dask-net
  volumes:
    - /home/fundecc/TrabalhoFinalBigData/dataset:/home/bigdata/work/dataset
```

OBRIGADO!!!!!!!!