



UFRJ Analytica



UFRJ

Análise do Consumo de Energia elétrica no Estado do Rio de Janeiro

Gustavo Felicidade

UFRJ Analytica 2023.1

Introdução



Introdução

O consumo de energia elétrica no Rio de Janeiro.

O consumo de energia elétrica é um tema relevante para entendermos a infraestrutura e as necessidades energéticas do Rio de Janeiro.

Com a crescente demanda por energia elétrica nas últimas décadas, é importante analisarmos como esse consumo evoluiu ao longo do tempo. A partir dos dados disponibilizados pela Base dos Dados, é possível realizar uma análise exploratória que permita compreender o consumo de energia elétrica na cidade do Rio de Janeiro.

Motivação

Motivação

Através de Artigos na mídia
pude verificar a respeito do
consumo no estado do Rio de
Janeiro



Sobre a geração de energia no Rio de Janeiro:

Em termos de potência instalada na **geração distribuída**, o Rio de Janeiro lidera o ranking da cidade com maior volume de geração, com 17,7MW.

Fonte:

<https://www.portalsolar.com.br/noticias/materias/rio-de-janeiro-lidera-o-ranki-ng-de-producao-de-energia-solar-distribuida-no-pais>

Motivação

Através de Artigos na mídia
pude verificar a respeito do
consumo no estado do Rio de
Janeiro



<https://brasil61.com/n/rio-de-janeiro-possui-a-maior-tarifa-media-de-energia-eletrica-entre-os-estados-da-regiao-sudeste-pind212980>

**Rio de Janeiro possui a maior
tarifa média de energia elétrica
entre os estados da Região
Sudeste**



Motivação

Rio registra maior alta do país no consumo de energia elétrica

Por **Aislan Loyola** - 18:34 - 4 de maio de 2021

Levantamento da Câmara de Comercialização de Energia Elétrica – CCEE mostra que o consumo de energia no Estado do Rio de Janeiro cresceu 26% na primeira quinzena de abril, na comparação com o mesmo período de 2020. É o estado com a maior alta do país nestas duas primeiras semanas do mês.

<https://monitormercantil.com.br/rio-registra-maior-alta-do-pais-no-consumo-de-energia-eletrica/>

Então, é válido levantarmos questionamentos sobre isso...

1. Quais são os dados disponíveis sobre o consumo de energia elétrica no Rio de Janeiro?

Nossos dados foram extraídos do datario.com

Os seguintes arquivos .csv são filtrados pela chave 'Elétrica' no contexto do Município do Rio de Janeiro:

1 Tabela 1686 - Consumo total mensal de energia elétrica, segundo classe de serviço - Município do Rio de Janeiro - 2019

url =

<https://www.data.rio/documents/7c4d6a17d1c94ccf8c652488318d691b/about>

2 Tabela 1687 - Total mensal de unidades consumidoras de energia elétrica, por classe de consumo, no Município do Rio de Janeiro em 2019

url =

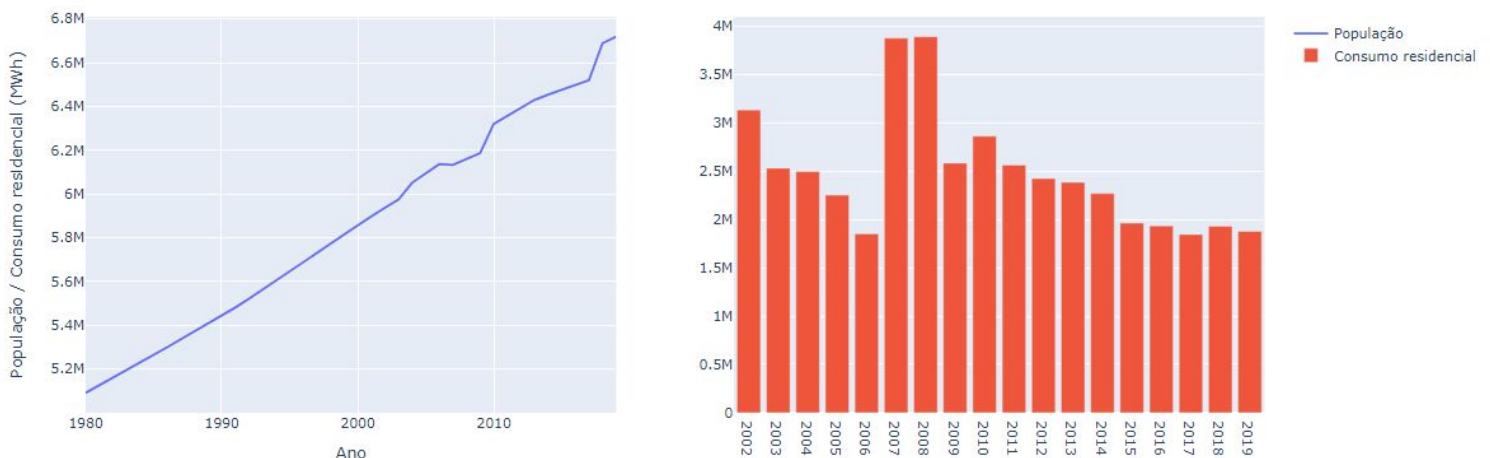
<https://www.data.rio/documents/c89710cd395b495a999ac84399fcc884/about>

2. Como o consumo de energia elétrica tem evoluído na cidade do Rio de Janeiro ao longo das últimas décadas?

Sim, Na minha análise pude verificar um aumento contínuo no consumo em função do crescimento da População.

No gráfico podemos ver:

População e consumo residencial do Rio de Janeiro (2002-2019)





3. Quais são as principais classes de serviço que consomem energia elétrica no Rio de Janeiro e como essa distribuição tem se modificado com o tempo?

Diversas classes de serviços são enquadradas como consumidoras dentre elas Residencial, Industrial, Comercial Rural e Poder Público, etc...

```
18 <class 'pandas.core.series.Series'>
[Jan 350121.439
Fev 360354.173
Mar 329053.693
Abr 460983.039
Mai 403608.164
Jun 362071.840
Jul 346843.060
Ago 359412.252
Set 353521.288
Out 383440.107
Nov 431789.763
Dez 461995.566
Name: 4603194.384, dtype: float64, Total
Jan 487581.090
Fev 482074.320
Mar 507677.940
Abr 403703.083
Mai 404002.814
Jun 360981.711
Jul 353784.829
Ago 357218.799
Set 338027.966
Out 382649.702
Nov 368786.479
Dez 440973.312
Name: 4887462.045000001, dtype: float64, Total
```

```
Ago 166069.652
Set 166969.980
Out 134774.379
Nov 173311.429
Dez 162866.421
Name: 1961873.716, dtype: float64, Total
Jan 133699.075
Fev 165112.309
Mar 158401.858
Abr 159576.196
Mai 169714.089
Jun 169067.312
Jul 164637.433
Ago 165400.908
Set 166023.530
Out 161235.587
Nov 159313.008
Dez 160667.466
Name: 1932928.770999999, dtype: float64, Total
Jan 126872.757
Fev 171943.845
Mar 159527.456
Abr 163652.080
Mai 153285.392
Jun 156027.387
Jul 137476.610
Ago 153755.985
Set 154440.211
Out 144245.307
Nov 161056.764
```

```
Ago 524640.763
Set 531730.346
Out 566401.765
Nov 579984.182
Dez 577785.391
Name: 6932628.609000001, dtype: float64, Total
Jan 623629.971
Fev 605636.532
Mar 621746.158
Abr 650178.147
Mai 570509.176
Jun 507464.002
Jul 497373.815
Ago 498047.224
Set 525495.279
Out 521008.617
Nov 537338.122
Dez 581880.926
Name: 6740307.969000001, dtype: float64, Total
Jan 615068.911000
Fev 625038.099000
Mar 625061.339000
Abr 573918.660000
Mai 590118.174000
Jun 503571.145000
Jul 471031.101000
Ago 482268.015000
Set 490924.233000
Out 500748.596129
Nov 528157.793144
Dez 559626.720354
```



4. Quais são as implicações do consumo de energia elétrica no meio ambiente na cidade do Rio de Janeiro?

O consumo de energia elétrica na cidade do Rio de Janeiro tem várias implicações no meio ambiente. A produção de energia elétrica geralmente envolve a queima de combustíveis fósseis, como o carvão e o petróleo, que emitem gases de efeito estufa, como dióxido de carbono e óxidos de nitrogênio, que contribuem para o aquecimento global e as mudanças climáticas.



5. Como é possível otimizar o consumo de energia elétrica na cidade do Rio de Janeiro?

Existem várias maneiras de otimizar o consumo de energia elétrica na cidade do Rio de Janeiro. Aqui estão algumas sugestões:

1. Promover o uso de fontes de energia renovável, como a energia solar, eólica e hidrelétrica, para reduzir a dependência de combustíveis fósseis.
2. Incentivar o uso de aparelhos elétricos mais eficientes em termos de energia, como lâmpadas LED, eletrodomésticos com classificação energética A e aparelhos eletrônicos com baixo consumo de energia.
3. Estimular a prática da economia de energia em casa, por meio de hábitos simples, como desligar os aparelhos da tomada quando não estão sendo usados, usar o ar condicionado de forma consciente e aproveitar a luz natural.
4. Investir em projetos de eficiência energética em edifícios públicos e privados, como a instalação de painéis solares e sistemas de iluminação LED, que podem reduzir significativamente o consumo de energia.
5. Criar campanhas de conscientização sobre o consumo consciente de energia elétrica e a importância da preservação do meio ambiente.

Modelo



Utilizando os dados abertos vamos agora aplicar os métodos analíticos...



```
import os
import csv
import pandas as pd
import numpy as np
import logging
import time
import matplotlib
import matplotlib.pyplot as plt
import plotly.graph_objs as go
import plotly.io as pio
from plotly.subplots import make_subplots

from tensorflow import keras
from keras.preprocessing.text import Tokenizer
from keras.utils import pad_sequences
from keras.models import Sequential
from keras.layers import Embedding, LSTM, Dense, Dropout
from keras.callbacks import EarlyStopping

from sklearn.metrics import accuracy_score, f1_score
```

▼ Configurando Log

```
logging.basicConfig(level=logging.INFO)
logger = logging.getLogger(__name__)
```

▼ Tratamento dos Dados

```
# Ler a página "2002" do arquivo
# Tabela 1686 - Consumo total mensal de energia elétrica, segundo classe de
# serviço - Município do Rio de Janeiro - 2019
df1 = pd.read_excel(
    r"/content/drive/MyDrive/Colab Notebooks/1686.xls",
    sheet_name="2002", header=6, skiprows=2, nrows=12)

# Ler a página "T 2257" do arquivo
# "Tabela 2257 - Consumo total, médio anual, mensal e diário
# de energia elétrica por habitante no Município do Rio de Janeiro
# entre 1980-2019"
df2 = pd.read_excel(
    r"/content/drive/MyDrive/Colab Notebooks/2257.xls",
    sheet_name="T 2257", header=6, skiprows=1, nrows=40)

# Imprimir os DataFrames
# print("Tabela 2257 - Consumo total, médio anual, mensal e diário \n de
# energia elétrica por habitante no Município do Rio de Janeiro entre 1980-2019")
# print(df2)
```




Extraindo Dataframe: Como o arquivo .xsl vem com informações além da tabela precisamos extrair o dataframe antes de analisá-los.

A função `collect_dataframes` fará esta extração retornando o dataframe armazenado em lista.

Note que o laço de repetição itera sobre os anos e assim obtemos os dataframes de cada aba:

```
[5] def collect_dataframes(file_path):  
    # criar uma lista vazia para armazenar os dataframes de cada ano  
    df_list = []  
    # percorrer os anos de 2002 a 2019  
    for year in range(2002, 2020):  
        # criar o nome da sheet com base no ano  
        sheet_name = str(year)  
        # ler o dataframe da sheet especificada  
        df = pd.read_excel(file_path, index_col=0, sheet_name=sheet_name, header=6, skiprows=2, nrows=12)  
        # adicionar o dataframe à lista  
        df_list.append(df)  
    # retornar a lista de dataframes  
    return df_list
```




Usaremos a função `collect_dataframes` para extrair as tabelas referentes ao consumo mensal de energia elétrica, segundo classe de serviço no Município do Rio de Janeiro - 2002 até 2019

```
file_path = r"/content/drive/MyDrive/Colab Notebooks/1686.xls"
df_list = collect_dataframes(file_path)
print(type(df_list))

# Extração de amostra do ano de 2002:
df_2002 = df_list[0]
print(type(df_2002))
```

Isso gera uma lista contendo DataFrames

```
<class 'list'>
<class 'pandas.core.frame.DataFrame'>
```



Vamos selecionar a coluna referente ao total de consumo por mês;

Para isso a função `print_second_column` é definida como:

```
[8] def print_second_column(df_list):  
    # percorrer cada dataframe na lista  
    for df in df_list:  
        # selecionar a segunda coluna e imprimir  
        print(df.iloc[:, 1])
```

A Função Consumo Mensal

Extração das colunas por consumo ao longo dos anos.

▼ A Função Consumo Mensal

Extração das colunas por consumo ao longo dos anos.

```
# Tabela 1686  
# Tabela 1686 - Consumo mensal de energia elétrica, segundo classe de serviço -  
# Município do Rio de Janeiro - 2002 a 2019  
def consumo_mensal(df_list):  
  
    # percorrer cada dataframe na lista e extraíndo  
    # colunas individualmente  
  
    # Coluna Consumo Total  
    consumo_total = []  
    # Coluna Consumo Residencial  
    consumo_residencial = []  
    # Coluna Consumo Industrial  
    consumo_industrial = []  
    # Coluna Consumo Comercial  
    consumo_comercial = []  
    # Coluna Consumo Rural  
    consumo_rural = []  
    # Coluna Consumo Poder Público  
    consumo_poder_publico = []  
    # Coluna Consumo Residencial  
    consumo_residencial = []  
    # Coluna Consumo Iluminação Pública  
    consumo_iluminacao_publica = []  
    # Coluna Consumo Serv. público  
    consumo_serv_publico = []  
    # Coluna Consumo Próprio  
    consumo_consumo_proprio = []  
  
    res = []
```

```
18  
<class 'pandas.core.series.Series'>
```

```
[11] # selecionar a coluna "Residencial" de todos os dataframes na lista  
residencial_list = [df.iloc[:, 2] for df in df_list]  
print("residencial_list")  
print(type(residencial_list))
```

Na Tabela 2257 - Consumo total, médio anual, mensal e diário de energia elétrica por habitante no Município do Rio de Janeiro entre 1980 e 2019:

```
populacao = df2.iloc[:, 5]
ano = df2.iloc[:, 0]
ano_populacao = df2.iloc[:, 0::5]
# print(f"Ano e populacao \n {ano_populacao}")
print(populacao)
```

Ano e populacao		
	Unnamed: 0	Unnamed: 5
0	1980	5.090790e+06
1	1981	5.125066e+06
2	1982	5.159573e+06
3	1983	5.194312e+06
4	1984	5.229285e+06
5	1985	5.264493e+06
6	1986	5.299939e+06
7	1987	5.335623e+06
8	1988	5.371547e+06
9	1989	5.407713e+06
10	1990	5.444123e+06
11	1991	5.480778e+06
12	1992	5.521452e+06
13	1993	5.562429e+06
14	1994	5.603709e+06
15	1995	5.645295e+06
16	1996	5.687191e+06
17	1997	5.729397e+06
18	1998	5.771916e+06
19	1999	5.814751e+06
20	2000	5.857904e+06
21	2001	5.897485e+06
22	2002	5.937253e+06
23	2003	5.974081e+06
24	2004	6.051399e+06
25	2005	6.094183e+06
26	2006	6.136652e+06
27	2007	6.132342e+06
28	2008	6.161047e+06
29	2009	6.186710e+06
30	2010	6.320446e+06
31	2011	6.355949e+06
32	2012	6.390290e+06
33	2013	6.429923e+06
34	2014	6.453682e+06
35	2015	6.476631e+06
36	2016	6.498837e+06
37	2017	6.520266e+06
38	2018	6.688927e+06
39	2019	6.718903e+06



Análise Exploratória dos Dados

Com o gráfico de consumo médio podemos analisar uma aumento considerável no consumo de energia no Rio de Janeiro

```
# Gráfico de linha para população
fig1 = go.Figure()
fig1.add_trace(go.Scatter(x=df2.iloc[:, 0], y=df2.iloc[:, 5], name='População'))

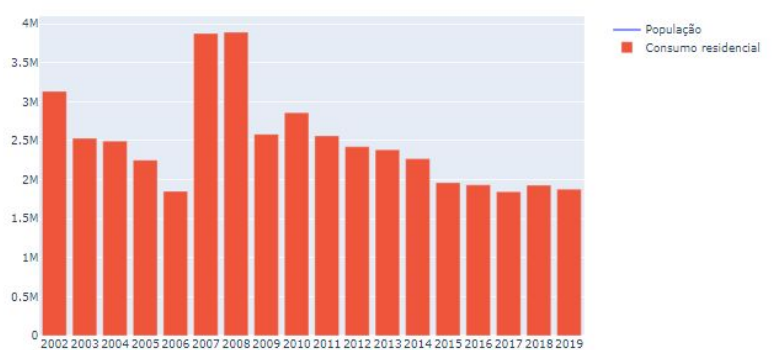
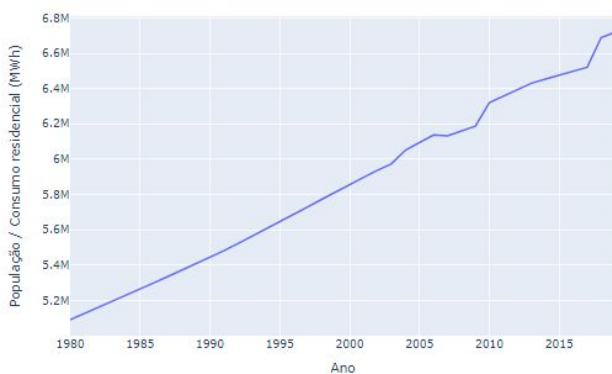
# Gráfico de barras para consumo residencial
fig2 = go.Figure()
fig2.add_trace(go.Bar(x=[str(year) for year in range(2002, 2020)], y=[df.iloc[:, 2].sum() for df in df_list], name='Consumo residencial'))

# Criar um mesmo gráfico com os dois plots
fig = make_subplots(rows=1, cols=2, shared_xaxes=True)
fig.add_trace(fig1.data[0], row=1, col=1)
fig.add_trace(fig2.data[0], row=1, col=2)

# Configurar o layout
fig.update_layout(title='População e consumo residencial do Rio de Janeiro (2002-2019)',
                  xaxis_title='Ano',
                  yaxis_title='População / Consumo residencial (MWh)')

# Mostrar o gráfico
fig.show()
```

População e consumo residencial do Rio de Janeiro (2002-2019)



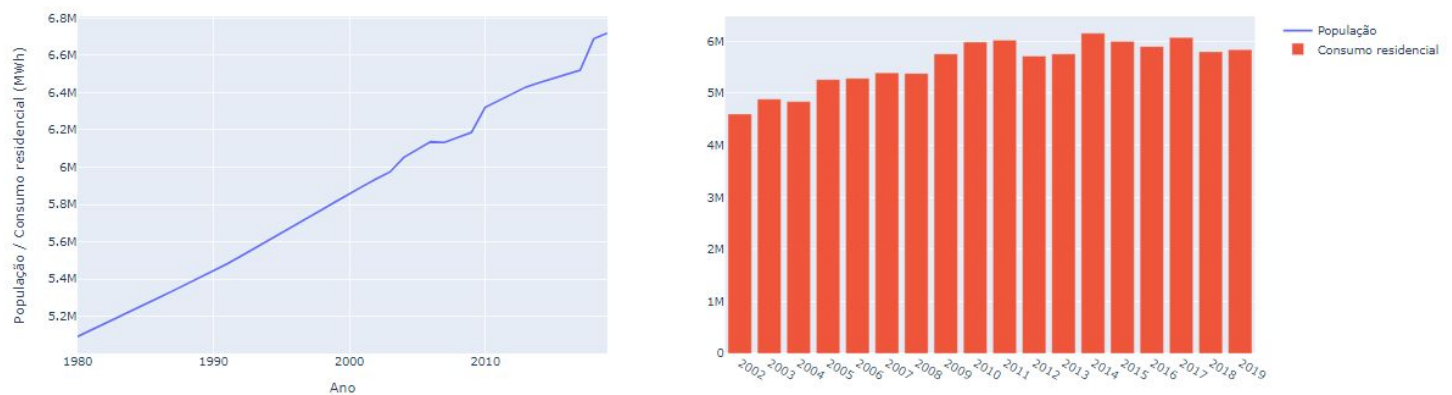
Análise Exploratória dos Dados

Sumários visuais do consumo de energia no
Rio de Janeiro

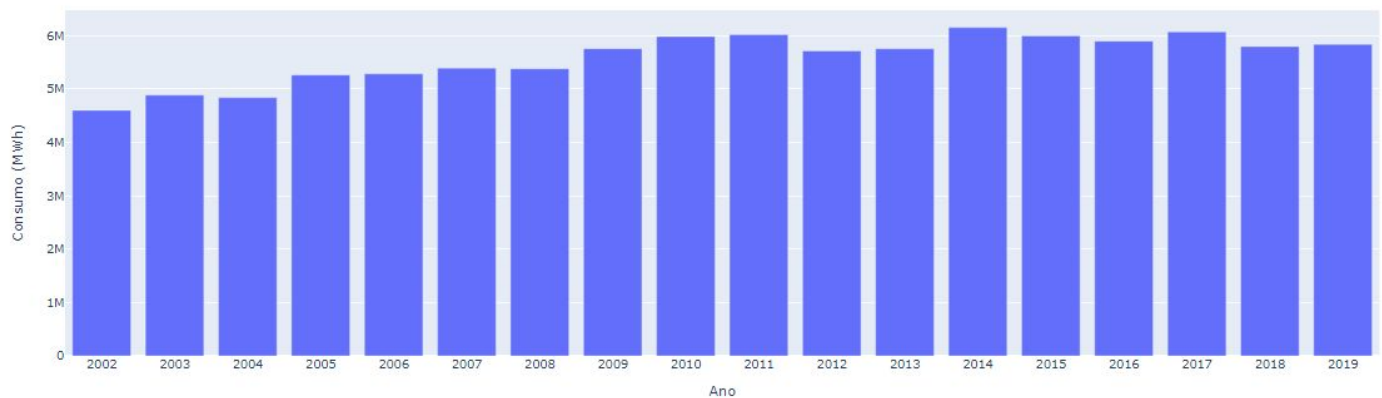
Análise Exploratória dos Dados

Sumários visuais do consumo de energia no Rio de Janeiro

População e consumo residencial do Rio de Janeiro (2002-2019)



Consumo de energia elétrica no Rio de Janeiro (2002-2019)

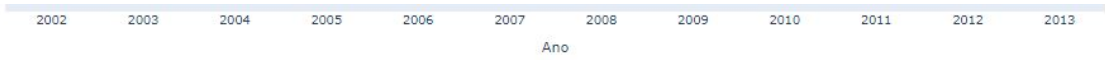


5

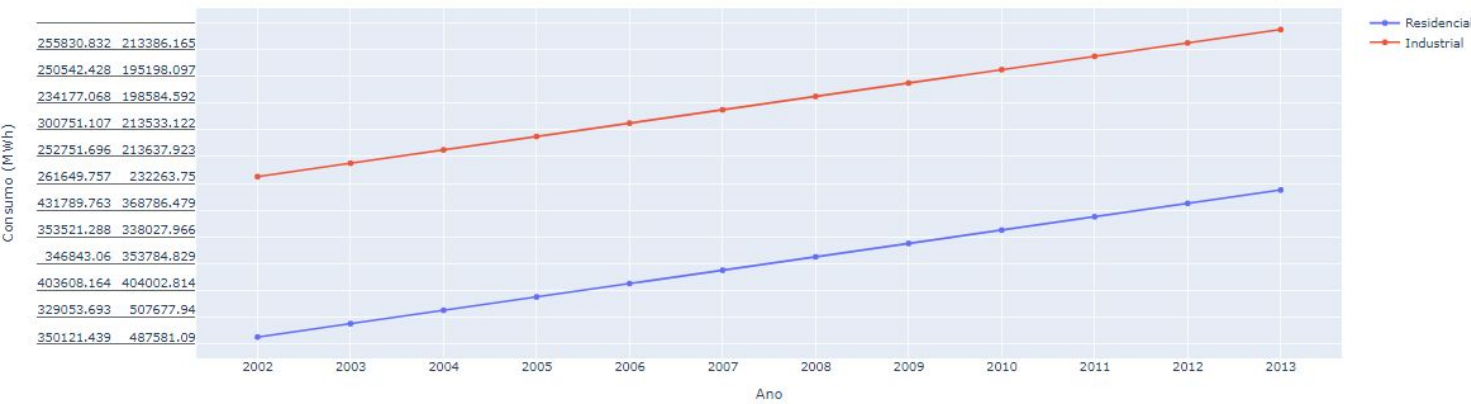




Análise Exploratória dos Dados



Consumo de energia elétrica no Rio de Janeiro (2002-2019)



Modelo



Utilizando Modelo LSTM para fazermos
predições...



Modelo LSTM

```
def prepare_data(res):  
    # Combine all the series in the list into one dataframe  
    df = pd.concat(res, axis=1)  
  
    # Transpose the dataframe to have each row represent a residency and each column a time step  
    df = df.T  
  
    # Convert the dataframe to a numpy array  
    data = df.values  
  
    # Normalize the data  
    data = (data - np.mean(data)) / np.std(data)  
  
    # Split the data into input (X) and output (y) sequences  
    X, y = data[:, :-1], data[:, -1]  
    print(type(X), len(X), type(y), len(y))  
    # Reshape X to be 3-dimensional (samples, time steps, features)  
    X = X.reshape(X.shape[0], X.shape[1], 1)  
  
    return X, y
```

```
def train_model(X_train, y_train, X_val, y_val):  
    model = create_model(input_shape=X_train.shape[1:])  
    early_stopping = EarlyStopping(monitor='val_loss', patience=5)  
    history = model.fit(X_train, y_train, epochs=15, batch_size=32, validation_data=(X_val, y_val))  
    return model, history
```

```
def plot_loss(history):  
    plt.plot(history.history['loss'], label='train')  
    # plt.plot(history.history['val_loss'], label='validation')  
    plt.xlabel('Epoch')  
    plt.ylabel('Loss')  
    plt.legend()  
    plt.show()
```



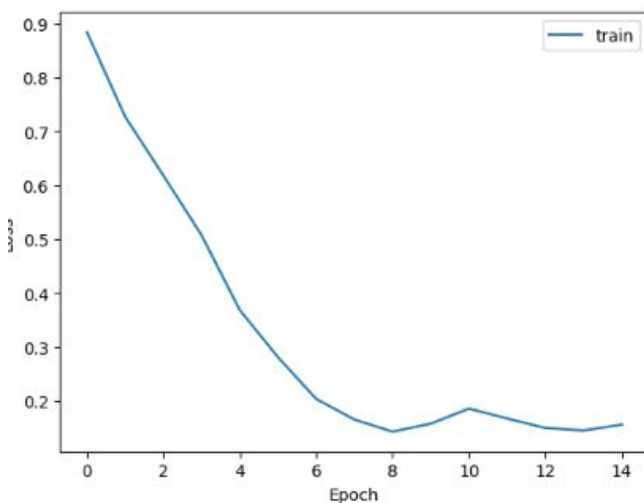
Exemplo de uso

```
X, y = prepare_data(res) # list of pandas series
X_train, y_train = X[:18], y[:18]
X_val, y_val = X[10:12], y[10:12]
X_test, y_test = X[12:], y[12:]
model, history = train_model(X_train, y_train, X_val, y_val)
plot_loss(history)
```

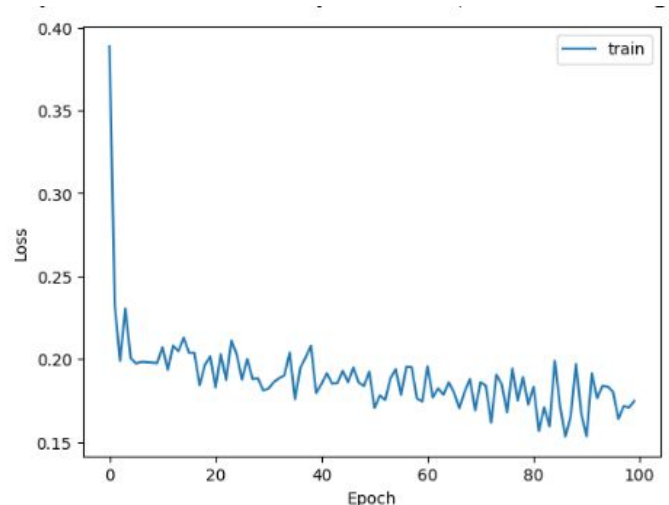
Como nosso dataset é pequeno testei diferentes *batch_size* e diferentes epochs para alcançar o melhor resultado possível,

```
<class 'numpy.ndarray'> 18 <class 'numpy.ndarray'> 18
Epoch 1/15
1/1 [=====] - 1s 519ms/step - loss: 0.8840 - val_loss: 0.0065
Epoch 2/15
1/1 [=====] - 0s 158ms/step - loss: 0.7273 - val_loss: 0.0043
Epoch 3/15
1/1 [=====] - 0s 169ms/step - loss: 0.6182 - val_loss: 0.0027
Epoch 4/15
1/1 [=====] - 0s 133ms/step - loss: 0.5074 - val_loss: 0.0016
Epoch 5/15
1/1 [=====] - 0s 144ms/step - loss: 0.3686 - val_loss: 0.0012
Epoch 6/15
1/1 [=====] - 0s 177ms/step - loss: 0.2811 - val_loss: 0.0016
Epoch 7/15
1/1 [=====] - 0s 149ms/step - loss: 0.2036 - val_loss: 0.0028
Epoch 8/15
1/1 [=====] - 0s 165ms/step - loss: 0.1653 - val_loss: 0.0051
Epoch 9/15
1/1 [=====] - 0s 151ms/step - loss: 0.1428 - val_loss: 0.0084
Epoch 10/15
1/1 [=====] - 0s 174ms/step - loss: 0.1577 - val_loss: 0.0128
Epoch 11/15
1/1 [=====] - 0s 173ms/step - loss: 0.1856 - val_loss: 0.0176
Epoch 12/15
1/1 [=====] - 0s 133ms/step - loss: 0.1672 - val_loss: 0.0219
Epoch 13/15
1/1 [=====] - 0s 163ms/step - loss: 0.1497 - val_loss: 0.0250
Epoch 14/15
1/1 [=====] - 0s 175ms/step - loss: 0.1450 - val_loss: 0.0260
Epoch 15/15
1/1 [=====] - 0s 137ms/step - loss: 0.1559 - val_loss: 0.0252
```

10 epochs



100 epochs



Métricas de Avaliação

Avaliando o resultado do nosso modelo...



Métricas de Avaliação

Avaliando o resultado do nosso modelo...

```
# Calcula a precisão (accuracy) para os dados de teste
def calculate_accuracy(y_true, y_pred):
    y_pred = np.round(y_pred)
    return accuracy_score(y_true, y_pred)
```

```
[21] # Calcula a medida F1 (F1-score) para os dados de teste
def calculate_f1_score(y_true, y_pred):
    y_pred = np.round(y_pred)
    return f1_score(y_true, y_pred)
```

```
import numpy as np
from sklearn.metrics import f1_score

# set threshold value
threshold = 0.5

# convert y_test to binary
y_test_binary = np.where(y_test >= threshold, 1, 0)

# make predictions
y_pred = model.predict(X_test)

# convert y_pred to binary
y_pred_binary = np.where(y_pred >= threshold, 1, 0)

# calculate accuracy and f1 score
accuracy = calculate_accuracy(y_test_binary, y_pred_binary)
f1 = f1_score(y_test_binary, y_pred_binary)

# print the results
print("Accuracy: {:.2f}%".format(accuracy * 100))
print("F1 Score: {:.4f}".format(f1))
```

Obtemos uma acurácia de 66.67% e um F1 Score de 0.7500

```
1/1 [=====] - 0s 495ms/step
Accuracy: 66.67%
F1 Score: 0.7500
```



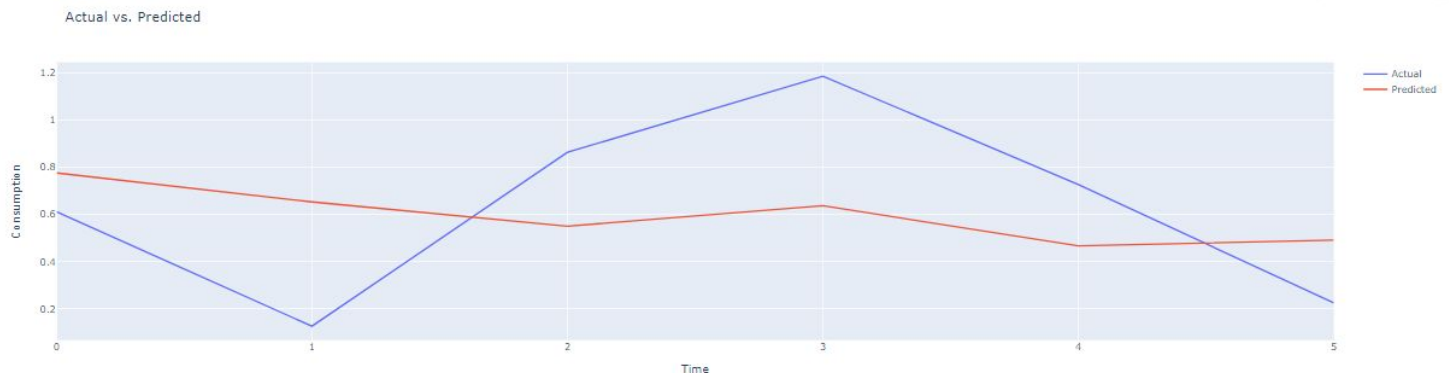
Métricas de Avaliação

Plotando as previsões

```
import plotly.graph_objs as go
from plotly.subplots import make_subplots

def plot_predictions(model, X_test, y_test):
    y_pred = model.predict(X_test)
    y_pred = y_pred.reshape(y_pred.shape[0],)
    fig = make_subplots(rows=1, cols=1)
    fig.add_trace(go.Scatter(x=list(range(len(y_test))), y=y_test, name='Actual', mode='lines'))
    fig.add_trace(go.Scatter(x=list(range(len(y_pred))), y=y_pred, name='Predicted', mode='lines'))
    fig.update_layout(title='Actual vs. Predicted', xaxis_title='Time', yaxis_title='Consumption')
    fig.show()
```

Os valores preditos são as estimativas do consumo de energia elétrica para o período de teste (ou seja, os próximos 6 meses) com base nos dados históricos fornecidos ao modelo. Esses valores representam a saída (ou "y_pred") do modelo treinado.





Métricas de Avaliação

Avaliando o resultado do nosso modelo...

Os valores de MAE (Mean Absolute Error) e MSE (Mean Squared Error) são medidas de desempenho do modelo em relação aos dados de teste. O MAE é a média da diferença absoluta entre as previsões e os valores reais, enquanto o MSE é a média dos quadrados das diferenças entre as previsões e os valores reais. Essas métricas permitem avaliar o quão bem o modelo está ajustando aos dados e quanto erro ele está cometendo em relação aos valores reais. No seu caso, o modelo parece ter um bom desempenho, já que o valor de MAE é relativamente baixo (0.23) e o valor de MSE também é baixo (0.11).

```
from sklearn.metrics import mean_absolute_error, mean_squared_error

mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)

print('MAE:', mae)
print('MSE:', mse)
```

MAE: 0.3460727441582632
MSE: 0.13992042805749247

MAE: 0.3460727441582632
MSE: 0.13992042805749247



Através da análise dos dados referentes ao consumo de energia elétrica disponibilizados pelo data rio fomos capaz de extrair informações práticas com o uso de ferramentas de linguagem de programação, no caso python e técnicas de tratamento de dados, análise exploratória e técnicas de predição usando redes neurais