



# Predição do Preço de Energia com ML

Aplicação de modelo supervisionado em  
problemas de regressão

Kaique Moraes da Silva | 10410548

# Desenvolvimento do Projeto na Pipeline de Modelos de ML

\*Baseada em KDD e CRISP-DM



## Mercado de Energia

Como funcionam os preços de energia?



## Dados dos Preços de Energia

Quais foram os dados utilizados?



## Processamento dos dados

Como os dados foram processados para alimentar o modelo?



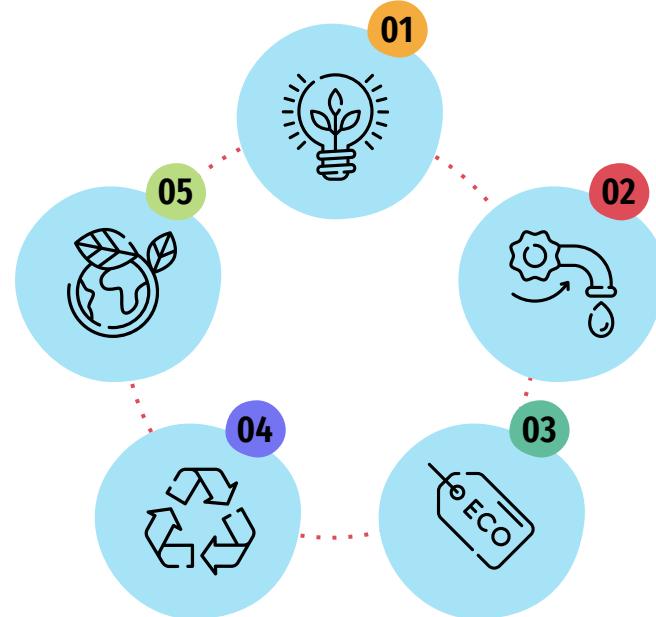
## Modelos utilizados

O que os modelos aplicados nos mostraram?



## Considerações finais

O que aprendemos com esses modelos?



# Como funcionam os preços de energia?

## 01 Mercado de Energia

Ambientes diferentes, múltiplas variáveis e streaming de dados

## 02 Séries Temporais

Dados diários, coleção sequencial complexa e tendências & sazonalidades



## 03 Volatilidade e Frequência

Oferta e demanda, dados de telemetria e fluxos contínuos

## 04 Desvio de Conceito

Alteração de sentido, perda de relação entre os dados e retreinos.

03

04

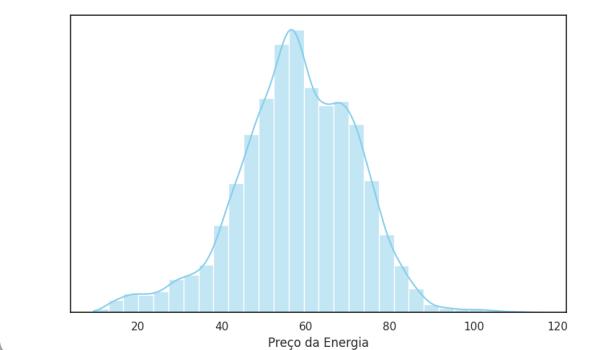
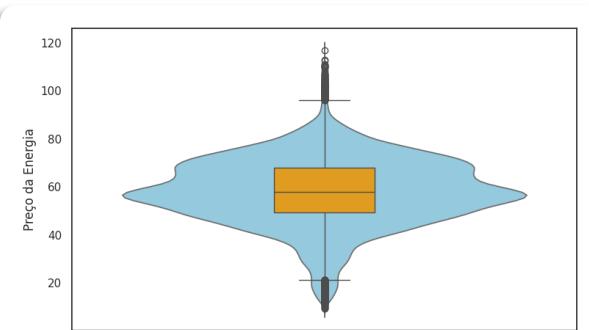
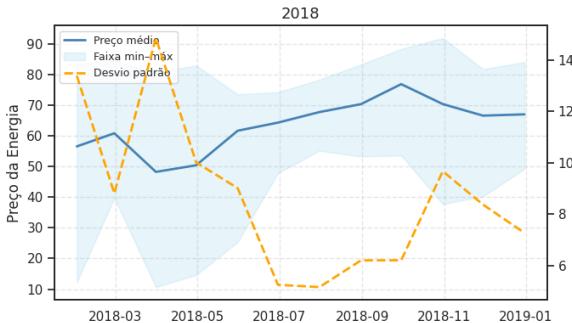
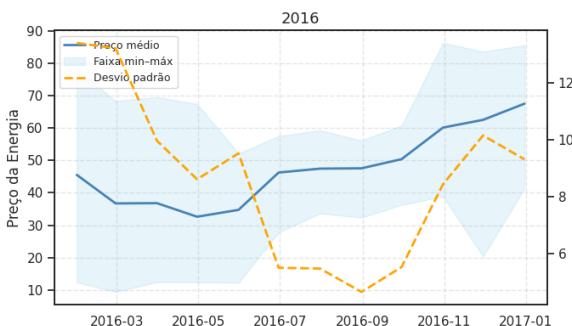
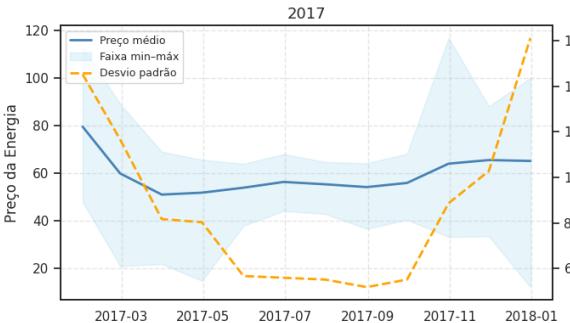
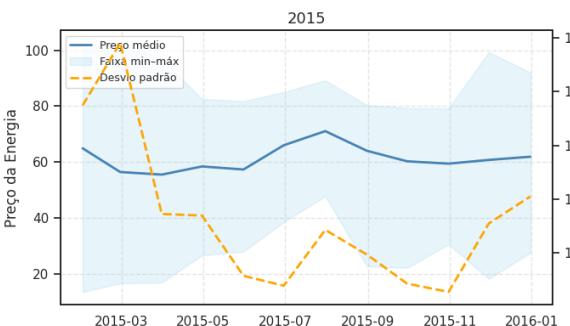
# Avaliando a variável alvo

Minímo: € 14.53

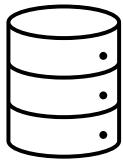
Média: € 57.90

Máximo: € 98.98

Coeficiente de Variação: 20.85%



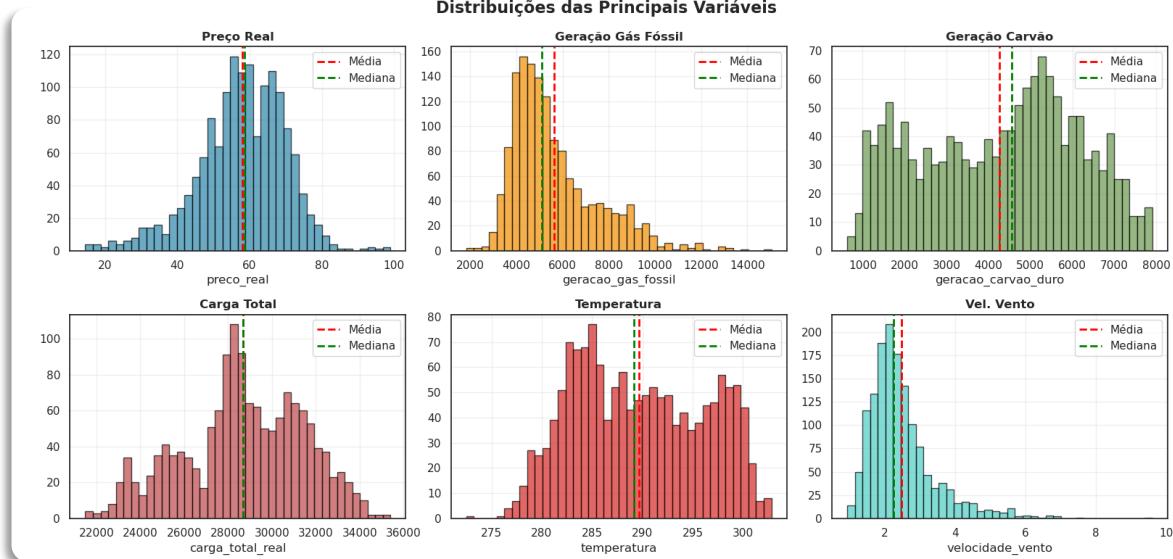
# Quais foram os dados utilizados?

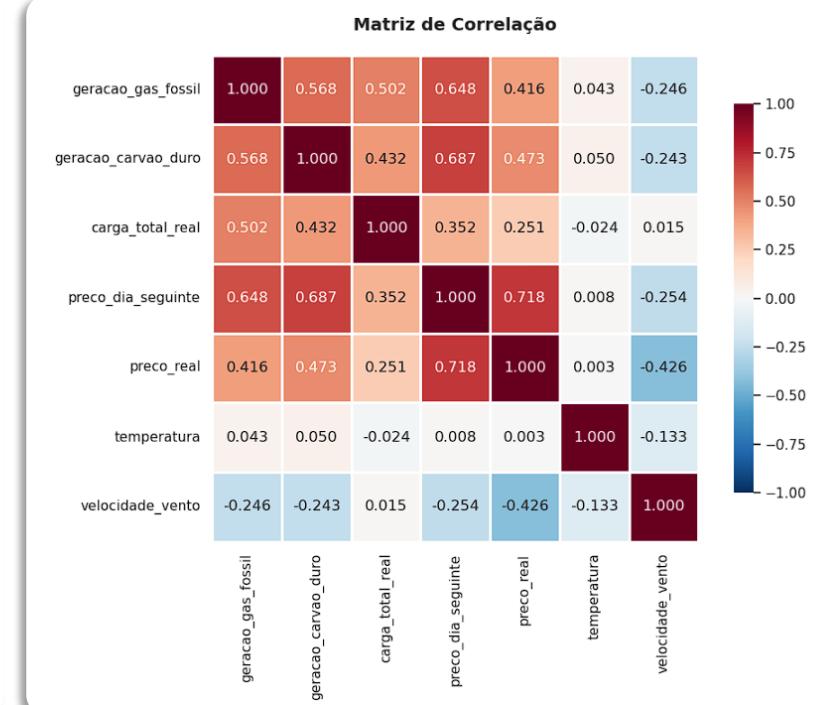
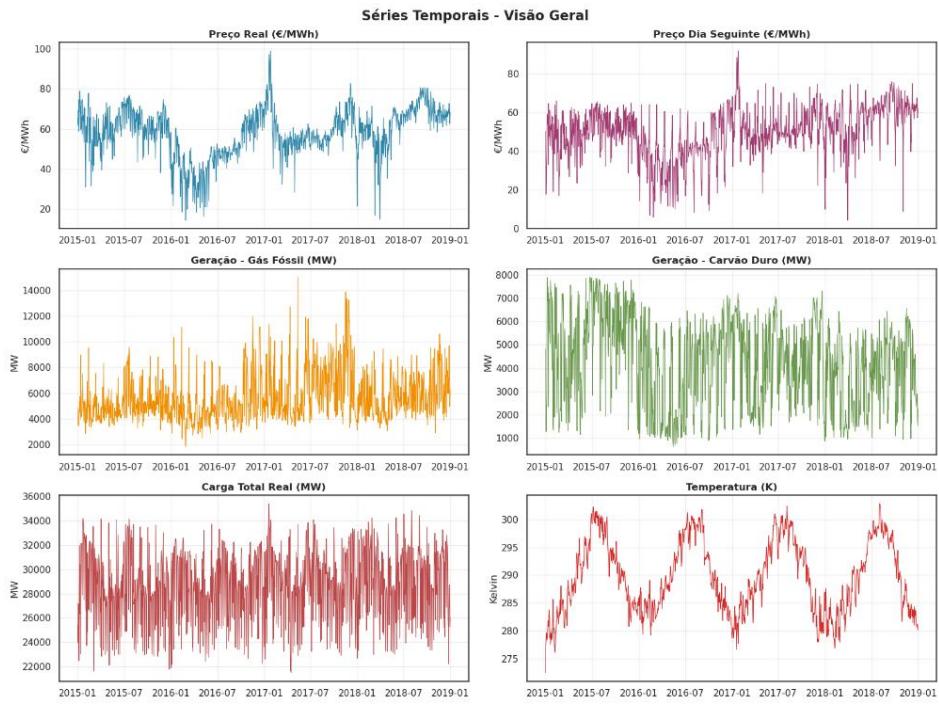


Total de variáveis: **46**  
Total de linhas: **35064**  
Linhas duplicadas: **0**

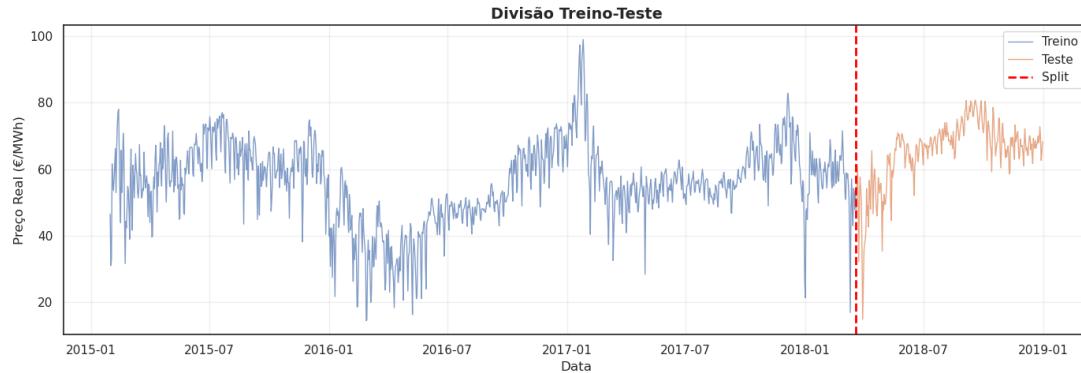


Total de variáveis: **7**  
Total de linhas: **1462**  
Linhas duplicadas: **0**





# Como os dados foram processados para alimentar os modelos?



Período dos dados: 2015 a 2018  
Divisão dos dados: 80% treino,  
10% teste e 10% validação

```
Normalização de variáveis numéricas

# StandardScaler
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Converter para DataFrame
X_train_scaled = pd.DataFrame(X_train_scaled, columns=X_train.columns,
index=X_train.index)
X_test_scaled = pd.DataFrame(X_test_scaled, columns=X_test.columns,
index=X_test.index)

print("Dados normalizados (StandardScaler)")
print("\nEstatísticas após normalização (5 primeiras features):")
print(X_train_scaled.iloc[:, :5].describe().loc[['mean', 'std']])
```

	geracao_gas_fossil	geracao_carvao_duro	carga_total_real	temperatura
mean	0.0000	0.0000	0.0000	-0.0000
std	1.0004	1.0004	1.0004	1.0004

	velocidade_vento
mean	-0.0000
std	1.0004

# O que os modelos aplicados nos mostraram?



## Regressão Linear

### Regressão Linear - TREINO

```
=====  
MAE:           €2.5273/MWh  
RMSE:          €3.3510/MWh  
MSE:            11.2289  
R2:            0.9207  
R2 Ajustado:   0.9176  
MAPE:           5.1164%  
MPE:            -0.5886%  
=====
```

### Regressão Linear - TESTE

```
=====  
MAE:           €2.8670/MWh  
RMSE:          €3.8699/MWh  
MSE:            14.9764  
R2:            0.8382  
R2 Ajustado:   0.8104  
MAPE:           4.9363%  
MPE:            2.3788%  
=====
```

## Árvores Aleatórias

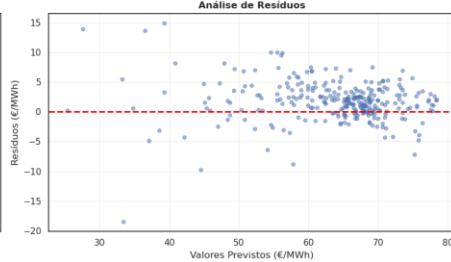
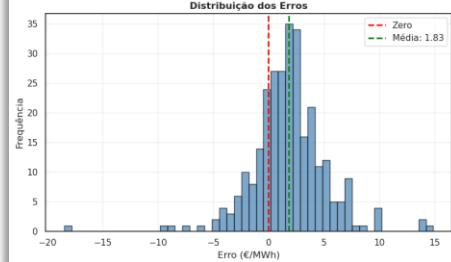
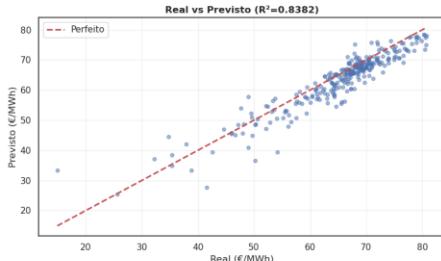
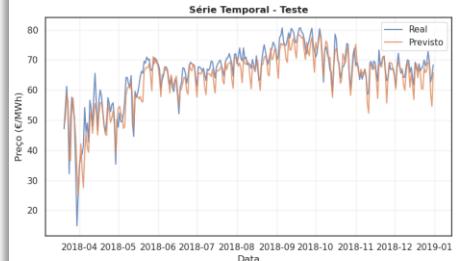
### Random Forest - TREINO

```
=====  
MAE:           €1.9697/MWh  
RMSE:          €2.8641/MWh  
MSE:            8.2033  
R2:            0.9420  
R2 Ajustado:   0.9398  
MAPE:           4.1822%  
MPE:            -0.9487%  
=====
```

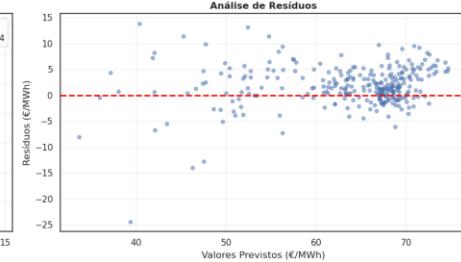
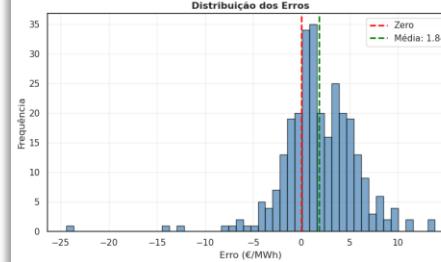
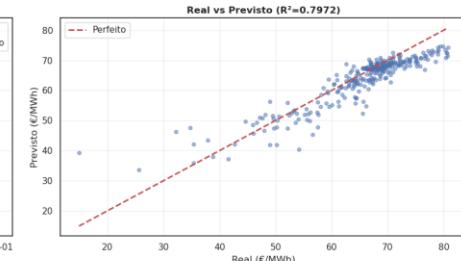
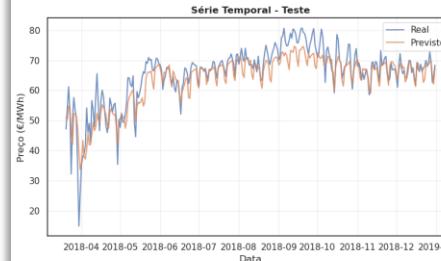
### Random Forest - TESTE

```
=====  
MAE:           €3.1715/MWh  
RMSE:          €4.3330/MWh  
MSE:            18.7753  
R2:            0.7972  
R2 Ajustado:   0.7623  
MAPE:           5.5146%  
MPE:            1.9297%  
=====
```

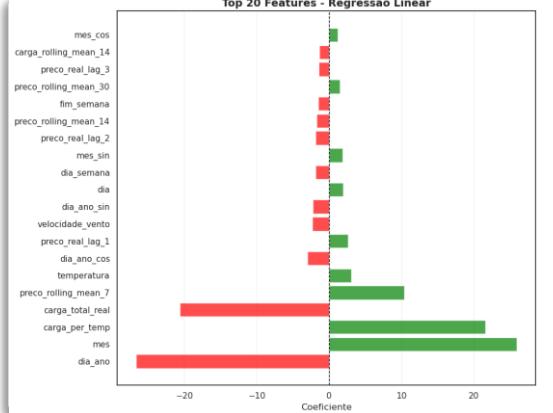
**R**egressão Linear - Análise de Previsões



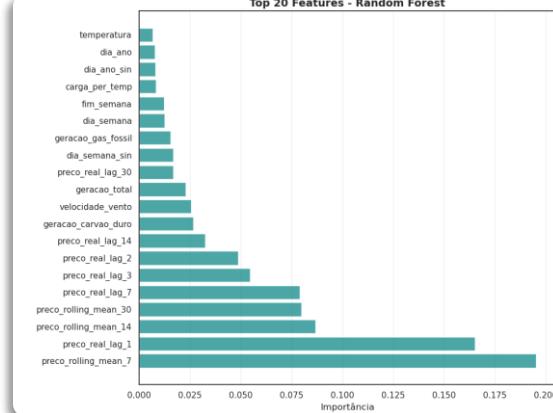
Random Forest - Análise de Previsões



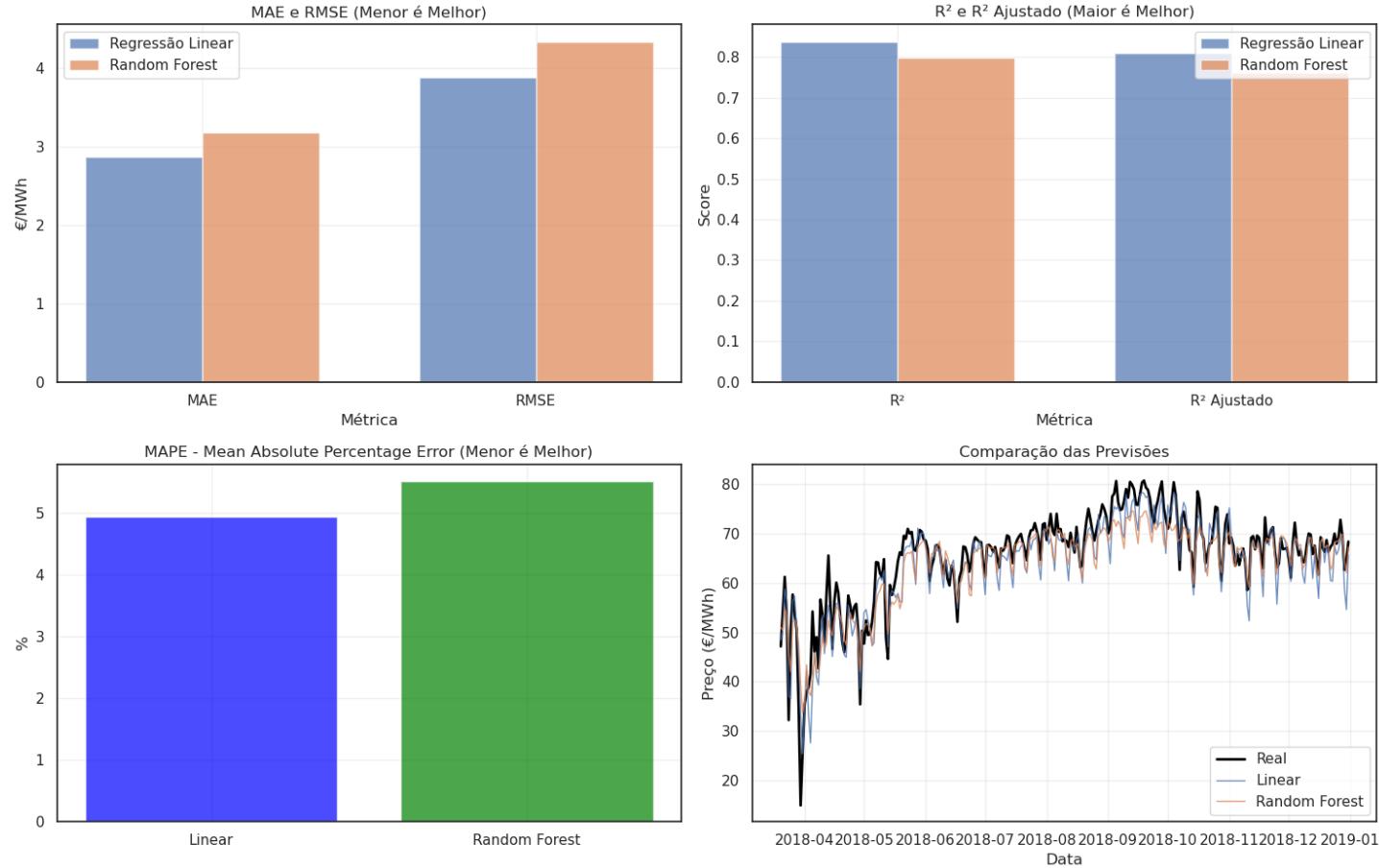
Top 20 Features - Regressão Linear



Top 20 Features - Random Forest



## Comparação dos Modelos - Desempenho no Teste



# O que aprendemos com esses modelos?

01

Os dois tiveram bons resultados, como escolher?

02

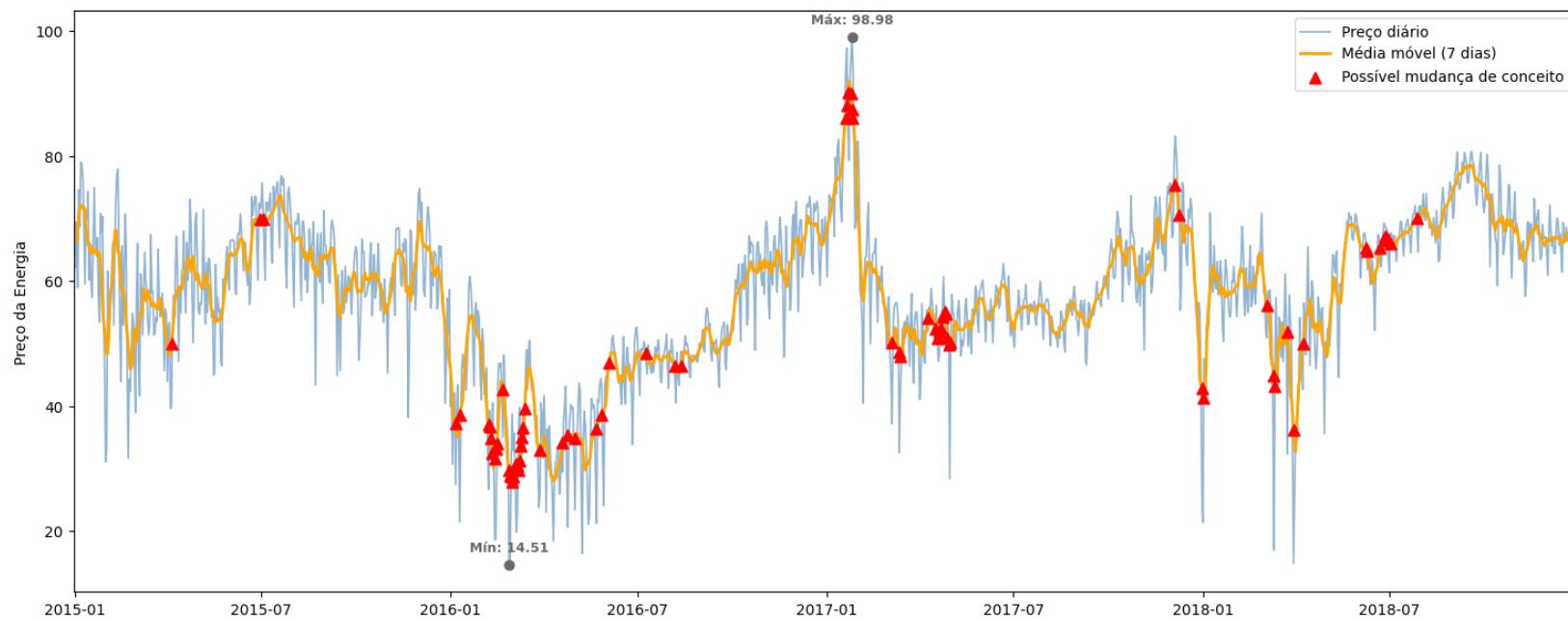
O pré-processamento dos dados é a parte mais relevante?

03

Dados, negócio ou problemas?



# Aplicação de Árvores de Hoeffding para Predição de Preços de Energia sob desvio de conceito



# Obrigado pela atenção!

**Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow**

Géron A. - 2<sup>a</sup> ed. O'Reilly Media, 2019.

**Regressão Linear e Regressão Logística**

<https://www.adit.io/posts/2016-02-20-Linear-Regression-in-Pictures.html>

**Energy Consumption Generation Prices and Weather Dataset**

<https://www.kaggle.com/datasets/nicholasjhana/energy-consumption-generation-prices-and-weather>.

**Algoritmos de Regressão**

<https://arunp77.medium.com/regression-algorithms-29f112797724>

**Conceitos de Preço**

<https://www.ccee.org.br/web/guest/precos/conceitos-precos>

**CRISP-DM, SEMMA e KDD**

<https://www.dataimd.com/post/entenda-as-principais-metodologias-de-minera-o-de-dados-crisp-dm-semma-e-kdd/>

