

Monitoramento e Previsão do Consumo em Edificações Inteligentes

"Na criação do código, temos a possibilidade de utilizar o Jupyter Notebook, aproveitando todos os seus recursos para melhorar a comunicação entre os envolvidos. Sempre com o objetivo de registrar as informações da melhor forma possível, tanto para os colegas e amigos da profissão quanto para o seu eu do futuro."

Sumário

- Entendimento do Negócio
- Objetivos
- Entendimento do Problema
- Público Alvo / Clientes Potenciais
- Importância e Relevância
- O Que Está Sendo Solucionado
 - Impacto Direto
- Oportunidades de Negócio e Mercado
- Benefícios e Melhorias Esperadas
- Referências

Entendimento do Negócio

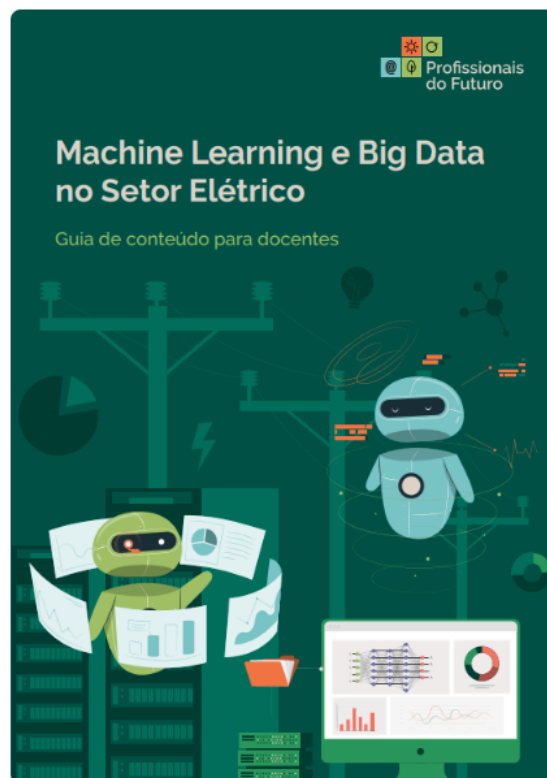
O consumo de energia é um dos principais desafios enfrentados por edificações modernas, impactando diretamente em custos operacionais, sustentabilidade e conforto dos ocupantes. Este projeto tem como finalidade **compreender e prever o comportamento energético** de um ambiente simulado, utilizando um conjunto de dados sintético que representa condições reais.

Com variáveis que envolvem fatores como temperatura, umidade, ocupação, uso de HVAC e iluminação, e energia renovável, é possível desenvolver modelos de aprendizado de máquina que antecipem padrões de consumo energético, auxiliando em estratégias de **eficiência e otimização**.

O público-alvo para essa solução inclui **empresas de energia, gestores prediais, startups de tecnologia limpa e instituições públicas**, interessados em reduzir gastos, automatizar decisões e melhorar o desempenho energético de suas instalações.

Uma outra aplicação famosa destes algoritmos é na **previsão de energia renovável**, principalmente fontes eólicas e fotovoltaicas. Através de dados históricos de medição e dados de operação, é possível prever com uma boa assertividade o quanto uma planta de geração será capaz de produzir em energia elétrica. **Esta é uma área que ainda existem muitos estudos sendo realizados.** [2]

As aplicações dos algoritmos de machine learning no setor elétrico são infinitas. A grande dificuldade é conseguir modelar o processo em termos computacionais para que eles possam representar com fidelidade os fenômenos que acontecem na realidade. Um outro impasse que o setor elétrico apresenta é a **disponibilidade de dados estruturados e de qualidade**. É comum algumas distribuidoras nem possuírem dados do processo que pretendem otimizar, precisando investir em uma etapa anterior, que é a de **instrumentação e aquisição de dados**. Esta etapa é custosa e muitas vezes **desencoraja os investimentos em tecnologia no setor elétrico.** [2]



Sumário

- Imports
 - Carregando o conjunto de dados
- Objetivos
- Entendimento do Conjunto de Dados
- Construção do dicionário dos dados
- Variável target
- Referências

Entendimento do Conjunto de Dados

Nesta seção, serão exploradas as principais características do conjunto de dados, como os tipos de variáveis, suas descrições, formatos e possíveis padrões. O objetivo é **obter uma visão geral** que guie as etapas seguintes da análise.



MRSIMPLE · UPDATED A YEAR AGO

70

<> Code

Download



Energy-consumption-prediction

Predict the Energy Consumption




Data Card


Code (10)

Discussion (0)

Suggestions (0)

Dataset: Energy-consumption-prediction

 **Descrição:** Predict the Energy Consumption

 **Autor:** MrSimple ·

 **Publicado em:** 1 ano atrás

 **Códigos disponíveis:** 10 notebooks |  **Visualizações:** 22.1K |  **Downloads:** 4558

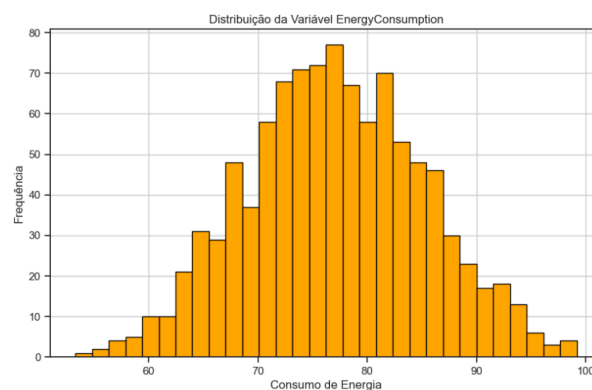
Observação

Informações obtidas no dia 24.04.2025

 **Acessar a Fonte dos Dados**

 Fonte do conjunto de dados: [Kaggle - Energy-consumption-prediction](#)

Variável target





Sumário

- 0. Imports
 - 0.1. Carregamento dos dados
- 1. Análise Inicial
 - 1.1. Análise da Quantidade de Dados
 - 1.2. Verificação da Existência de Nulos no Dataset
 - 1.3. Verificação do tipo de cada variável
 - 1.4. Criação de Novas Variáveis Utilizando a Variável Timestamp
 - 1.5. Verificação da Cardinalidade de Cada Variável
 - 1.6. Análise estatística
 - 1.7. Verificação de Outliers
- 2. Análise Bivariada
- 3. Análise Multivariada



Objetivos da Análise



1. Compreender o conteúdo dos dados:

- O que cada coluna representa
- Os tipos de dados (numéricos, categóricos, datas, booleanos)
- Como as variáveis podem estar relacionadas entre si



2. Identificar problemas de qualidade dos dados:

- Valores faltantes ou nulos
- Dados duplicados
- Outliers (valores muito fora do padrão esperado)



3. Visualizar padrões e distribuições:

- Ver como a energia é consumida ao longo do tempo
- Avaliar como temperatura, ocupação e uso de HVAC afetam o consumo
- Ver se há padrões diferentes em dias de semana vs. feriados



4. Guiar decisões de modelagem:

- A análise inicial ajuda a decidir que tipo de modelo usar (regressão? classificação?)
- Auxilia na engenharia de atributos (criação de novas colunas úteis)
- Pode mostrar a necessidade de transformar variáveis (como normalizar o SquareFootage, por exemplo)



5. Evitar erros futuros:

- Treinando modelos em dados sujos
- Tomando decisões com base em informações incompletas ou enganosas
- Ignorando variáveis-chave ou efeitos importantes



1.1. Análise da Quantidade de Dados

Nesta etapa, avaliamos o tamanho do conjunto de dados, verificando a quantidade total de registros e colunas disponíveis. Isso nos ajuda a entender a robustez da base e sua adequação para análises estatísticas ou treinamento de modelos.

```
In [30]: # dimensionalidade do dataframe:
df_analise.shape
```

```
Out[30]: (1000, 11)
```

Comentário:

O dataset possui uma dimensão de (1000, 11), ou seja, contém 1.000 registros e 11 variáveis. Dentre as 11 variáveis, 10 são descritivas e fornecem informações relevantes sobre o consumo de energia, ocupação, uso de HVAC, entre outros fatores.



Quantidade de valores na amostra:

A quantidade de 1000 registros pode limitar o rendimento do modelo, visto que, não temos uma quantidade alta de dados.

1.3. Verificação do tipo de cada variável

```
In [33]: # informações gerais do dataframe:
df_analise.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 11 columns):
#   Column              Non-Null Count  Dtype
---  -
0   Timestamp            1000 non-null   object
1   Temperature          1000 non-null   float64
2   Humidity             1000 non-null   float64
3   SquareFootage        1000 non-null   float64
4   Occupancy            1000 non-null   int64
5   HVACUsage            1000 non-null   object
6   LightingUsage        1000 non-null   object
7   RenewableEnergy      1000 non-null   float64
8   DayOfWeek            1000 non-null   object
9   Holiday              1000 non-null   object
10  EnergyConsumption    1000 non-null   float64
dtypes: float64(5), int64(1), object(5)
memory usage: 86.1+ KB
```

Comentários:

Podemos conduzir algumas coisas a partir dessa saída:

- **Timestamp:** Podemos extrair informações dessa variável, que serão mais interessantes para o modelo quando separadas em atributos distintos, como **dia**, **mês**, **ano**, **hora**, **período**, **feriado**, etc.
- **Variáveis Categóricas:** É necessário aplicar, na seção de pré-processamento, a conversão das variáveis categóricas para numéricas, como **HVACUsage**, **LightingUsage**, **DayOfWeek**, **Holiday** e outras que surgirem da fragmentação da variável **Timestamp**.
- **Normalização:** Identificamos que é necessário realizar a **normalização das variáveis** para melhorar a performance do modelo.
- **Tipo Correto:** Com relação aos **valores máximo e mínimo** de cada variável, principalmente as **numéricas do tipo float** podemos observar que, ao reduzir o intervalo de variação, podemos otimizar o espaço necessário para o armazenamento desses dados, tornando o conjunto de dados mais leve (isso é fundamental para projetos com grandes volumes de amostra).
- **Criação de Novas Variáveis:** Com as variáveis disponíveis no dataset, podemos criar novas métricas, como: **EUI – Energy Use Intensity** (Intensidade de Uso de Energia), **Consumo por ocupante**, **Eficiência do sistema HVAC** (Climatização), entre outras.

1.5. Verificação da Cardinalidade de Cada Variável

A verificação da cardinalidade de cada variável é importante para entender a diversidade de valores que cada uma pode assumir. Este processo nos ajuda a identificar variáveis com um número limitado de valores distintos, como variáveis categóricas, e a observar se alguma delas pode precisar de transformação para garantir que o modelo trabalhe de forma mais eficiente.

```
In [38]: # Função para extrair os metadata de um dataframe:
metadata_df = generate_metadata(df_analise)
metadata_df
```

```
Out[38]:
```

	nome_variavel	tipo	qt_nulos	percent_nulos	cardinalidade	min	max	sugestao_tipo
0	Timestamp	datetime64[ns]	0	0.0	1000	None	None	None
1	Fim_de_Semana	bool	0	0.0	2	False	True	bool
2	Hora	int32	0	0.0	24	0	23	uint8
3	Dia	int32	0	0.0	31	1	31	uint8
4	Mês	int32	0	0.0	2	1	2	uint8
5	Ano	int32	0	0.0	1	2022	2022	uint16
6	Occupancy	int64	0	0.0	10	0	9	uint8
7	Temperature	float64	0	0.0	1000	20.007565	29.998671	float32
8	Humidity	float64	0	0.0	1000	30.015975	59.969085	float32
9	SquareFootage	float64	0	0.0	1000	1000.512661	1999.982252	float32
10	RenewableEnergy	float64	0	0.0	1000	0.006642	29.965327	float32
11	EnergyConsumption	float64	0	0.0	1000	53.263278	99.20112	float32
12	DayOfWeek	object	0	0.0	7	None	None	None
13	LightingUsage	object	0	0.0	2	None	None	None
14	HVACUsage	object	0	0.0	2	None	None	None
15	Holiday	object	0	0.0	2	None	None	None
16	Período	object	0	0.0	4	None	None	None

PROJETO



MACHINE LEARNING
ENERGY CONSUMPTION

Monitoramento e Previsão do Consumo em Edificações Inteligentes




1 Veja como essa

fase foi construída do zero!

Código e os detalhes estão disponíveis no repositório:



[projeto_previsão_consumo_energia](#)

Seu comentário
é muito
importante! 

Elaborado por:

Yago Castro dos Reis