

# Aula 30 – Engenharia de Features para Séries Temporais

Imagine que você está tentando prever o clima de amanhã. Olhar apenas para a temperatura de hoje pode não ser suficiente, certo? Você provavelmente consideraria a temperatura de ontem, a média da semana, se é verão ou inverno, e até mesmo se houve algum evento climático incomum recentemente. No mundo dos dados, especialmente com séries temporais, essa intuição é a base da **Engenharia de Features**. É a arte e a ciência de transformar dados brutos em informações mais ricas e preditivas, que os modelos de Machine Learning podem realmente "entender" e usar para fazer previsões mais precisas.

Em muitos cenários, desde a previsão de vendas até a manutenção preditiva de equipamentos, a capacidade de antecipar o futuro com base em padrões passados é um diferencial competitivo enorme. No entanto, os dados de séries temporais, por sua natureza sequencial, apresentam desafios únicos. Eles não são apenas um conjunto de pontos independentes; cada ponto está intrinsecamente ligado ao que veio antes. É aqui que a engenharia de features se torna não apenas útil, mas essencial, atuando como a ponte entre os dados brutos e um modelo preditivo de alta performance.

Nesta aula, nosso objetivo é desvendar as técnicas fundamentais para extrair o máximo de valor de seus dados de séries temporais. Você aprenderá a construir features de defasagem (lag), que capturam a memória do passado; features de janela móvel (rolling window), que sumarizam tendências e volatilidade recentes; e features baseadas em calendário, que codificam a sazonalidade e outros padrões temporais. Ao final, você estará apto a aplicar essas estratégias para enriquecer seus conjuntos de dados e construir modelos preditivos mais robustos e interpretáveis, preparando o terreno para as aplicações de Machine Learning que veremos a seguir.

# O Coração da Previsão: Por Que Features?



## O Desafio do Modelo

Algoritmos não "entendem" padrões humanos naturalmente - precisam de tradução explícita



## O Papel do Detetive

Features são como pistas que ajudam o modelo a construir um quadro completo da situação



## O Fator Decisivo

Features bem elaboradas superam algoritmos sofisticados com features ruins

Quando olhamos para uma série temporal, como o registro diário de vendas de uma loja, o que vemos é uma sequência de números. Para nós, humanos, é fácil identificar padrões: "As vendas caem nos fins de semana", "Há um pico antes do Natal". Mas para um algoritmo de Machine Learning, esses números são apenas isso: números. O modelo não "sabe" que sábado e domingo são dias de descanso, nem que dezembro é um mês de festas. Ele precisa que essas informações sejam explicitamente apresentadas a ele, em um formato que possa processar.

**É exatamente essa a função da engenharia de features: traduzir o conhecimento de domínio e os padrões implícitos nos dados para uma linguagem que o modelo entenda.**

Pense nisso como um detetive que, ao invés de apenas olhar para uma cena de crime, coleta impressões digitais, analisa padrões de pegadas, e entrevista testemunhas. Cada uma dessas ações gera uma "feature" – uma peça de informação valiosa que, combinada, ajuda a construir um quadro mais completo e a chegar a uma conclusão mais precisa. Sem essa etapa, o modelo estaria tentando resolver um quebra-cabeça com apenas algumas peças espalhadas aleatoriamente.

A criação de features relevantes é, muitas vezes, o fator mais determinante para o sucesso de um modelo preditivo. Um algoritmo sofisticado com features ruins pode ter um desempenho inferior a um algoritmo simples com features bem elaboradas. É um processo iterativo que exige criatividade, conhecimento do negócio e uma boa compreensão das características dos dados. Ao transformar o tempo em variáveis significativas, abrimos um leque de possibilidades para que nossos modelos capturem a complexidade do mundo real.

# Desvendando o Passado: Features de Lag (Defasagem)

A intuição mais básica ao prever o futuro em séries temporais é que o passado recente tem uma forte influência. Se as vendas de café foram altas ontem, há uma boa chance de que sejam razoavelmente altas hoje. Essa "memória" do sistema é capturada pelas **features de lag**, ou defasagem. Elas são simplesmente os valores anteriores da própria série temporal ou de outras séries relacionadas. É como olhar para o espelho retrovisor enquanto dirige: você usa a informação do que acabou de passar para tomar decisões sobre o que está por vir.

Para criar uma feature de lag, pegamos o valor de uma variável em um ponto no tempo  $t$  e o associamos ao ponto no tempo  $t+k$ , onde  $k$  é o número de períodos de defasagem. Por exemplo, se queremos prever as vendas de hoje, podemos usar as vendas de ontem (lag 1), da semana passada (lag 7) ou até do mesmo dia do ano anterior (lag 365). A escolha do  $k$  é crucial e geralmente depende da periodicidade dos dados e dos padrões que se deseja capturar.

## Lags Comuns

- **Lag 1:** Dependência diária
- **Lag 7:** Padrão semanal
- **Lag 28:** Ciclo mensal
- **Lag 365:** Sazonalidade anual

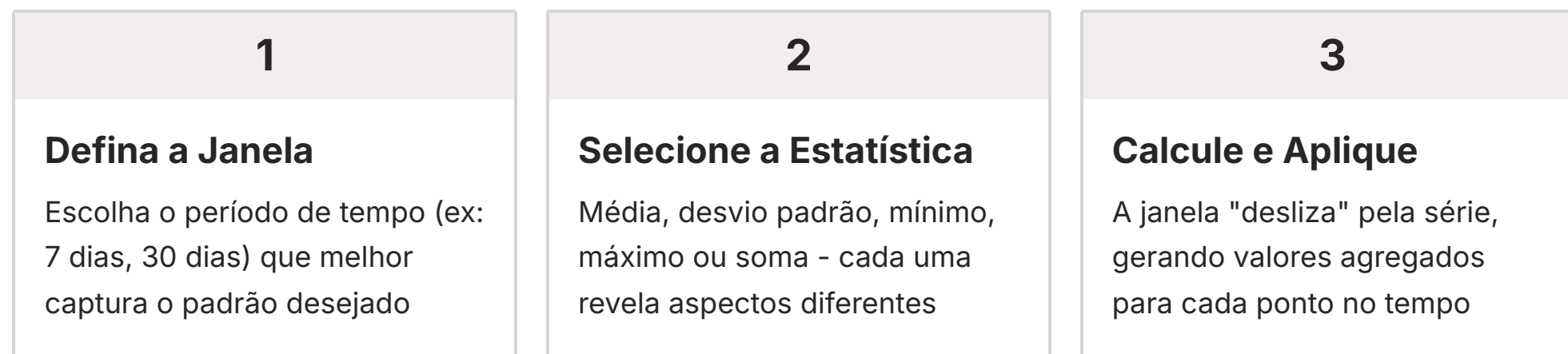
---

## Exemplo Prático: Consumo de Energia

Vamos considerar um exemplo prático. Se você está prevendo o consumo de energia elétrica, saber o consumo da hora anterior (lag 1 hora) ou do mesmo horário no dia anterior (lag 24 horas) pode ser extremamente preditivo. Essas features de lag fornecem ao modelo uma visão direta da dinâmica temporal da série, permitindo que ele aprenda a relação entre o passado e o presente/futuro. É a forma mais direta de incorporar a autocorrelação, ou seja, a correlação de uma série com suas próprias versões defasadas, que é uma característica fundamental das séries temporais.

# A Visão Ampla: Features de Janela Móvel (Rolling Window)

Enquanto as features de lag nos dão pontos específicos do passado, muitas vezes precisamos de uma visão mais agregada. Um único valor de ontem pode ser um ruído; o que realmente importa é a tendência ou a volatilidade dos últimos dias. É aqui que entram as **features de janela móvel**, também conhecidas como rolling window features. Elas calculam estatísticas (como média, desvio padrão, mínimo, máximo, soma) sobre um período de tempo contínuo que "se move" junto com a série temporal. Pense nisso como uma lupa que se desloca sobre os dados, revelando padrões locais.



Imagine que você está monitorando o desempenho de um atleta. Olhar apenas para o resultado da última corrida (lag) é útil, mas saber a média dos últimos cinco resultados (rolling mean) ou a variação entre eles (rolling standard deviation) oferece uma perspectiva muito mais rica sobre sua forma atual.

Essas estatísticas de janela móvel ajudam o modelo a entender a dinâmica de curto e médio prazo da série, como tendências emergentes, picos de volatilidade ou períodos de estabilidade. Elas são excelentes para suavizar o ruído e destacar os sinais subjacentes.

Para criar uma feature de janela móvel, você define o tamanho da "janela" (por exemplo, 7 dias, 30 dias) e a estatística a ser calculada. Por exemplo, um "rolling mean de 7 dias" para as vendas diárias calcularia a média das vendas dos últimos 7 dias para cada ponto no tempo. Um "rolling standard deviation de 30 dias" capturaria a volatilidade mensal. Essas features são particularmente poderosas em cenários como a previsão de preços de ações, onde a média móvel pode indicar tendências e o desvio padrão móvel, o risco. Elas fornecem um contexto estatístico que os valores de lag isolados não conseguem oferecer.

# O Ritmo da Vida: Features Baseadas em Calendário

As séries temporais são frequentemente influenciadas por ciclos previsíveis que se alinham com o calendário. O volume de tráfego é diferente em um dia de semana e em um fim de semana. As vendas de sorvete aumentam no verão e caem no inverno. Esses padrões sazonais e cíclicos são informações valiosas que podem ser codificadas através de **features baseadas em calendário**. Elas transformam componentes da data e hora em variáveis numéricas ou categóricas que o modelo pode aprender a associar a diferentes comportamentos da série.



## Dia da Semana

Segunda = 1, Domingo = 7. Captura padrões de rotina semanal e diferenças entre dias úteis e fins de semana.



## Mês do Ano

Janeiro = 1, Dezembro = 12. Identifica sazonalidade anual e variações climáticas ao longo do ano.



## Feriados e Eventos

Variáveis binárias (0/1) para datas especiais como Black Friday, Natal, ou feriados nacionais.



## Hora do Dia

0-23 horas. Essencial para dados intradiários, capturando picos de atividade em horários específicos.

---

## Aplicação Prática

Pense na sua própria rotina. Ela muda drasticamente dependendo do dia da semana, do mês, ou se é um feriado. Para um modelo, é a mesma coisa. Se ele souber que um determinado ponto de dados corresponde a uma segunda-feira, a um mês de dezembro, ou a um feriado nacional, ele pode ajustar sua previsão de acordo. As features de calendário incluem o dia da semana (segunda-feira = 1, domingo = 7), o mês do ano, o dia do mês, o dia do ano, a hora do dia, e até mesmo indicadores binários para feriados ou eventos especiais.

Um exemplo clássico é a previsão de demanda em varejo. Saber que é "Black Friday" ou "Natal" é crucial para prever picos de vendas. Da mesma forma, em serviços públicos, prever o consumo de água ou energia exige considerar se é um dia útil ou fim de semana. Essas features de calendário são uma forma eficaz de capturar a **sazonalidade** e os **efeitos de calendário** que são tão prevalentes em muitos conjuntos de dados de séries temporais, permitindo que o modelo aprenda padrões que se repetem em intervalos regulares.

# Combinando Estratégias e Desafios Comuns

## A Magia da Combinação

A verdadeira magia da engenharia de features para séries temporais reside na combinação inteligente das técnicas. Raramente um único tipo de feature será suficiente.

Um modelo de previsão de vendas, por exemplo, pode se beneficiar de: vendas do dia anterior (lag), média de vendas da última semana (rolling mean), desvio padrão de vendas do último mês (rolling std), além do dia da semana e um indicador de feriado (calendar features). A combinação dessas informações cria um vetor de características muito mais rico e informativo para o modelo.

## Desafios a Superar

### 1. Seleção de Features

Nem toda feature é útil. Features redundantes ou irrelevantes podem confundir o modelo ou levar a overfitting. É preciso discernimento para escolher as melhores.

### 2. Dados Ausentes

A criação de lags e janelas móveis pode gerar valores ausentes no início da série. Estratégias de imputação ou remoção precisam ser aplicadas cuidadosamente.

### 3. Vazamento de Dados (Data Leakage)

**CRÍTICO:** Nunca use informações do futuro para criar features do passado. Ao calcular médias móveis, use apenas dados anteriores ou iguais ao ponto de previsão.

Pense em um chef preparando um prato complexo. Ele não joga todos os ingredientes na panela de uma vez. Ele seleciona os melhores, prepara-os cuidadosamente (cortando, temperando), e os adiciona no momento certo para que cada um contribua para o sabor final. Da mesma forma, na engenharia de features, é preciso discernimento para escolher as features certas, pré-processá-las adequadamente e combiná-las de forma que potencializem o desempenho do modelo, sem introduzir vieses ou informações enganosas.

# A Revolução da Automação: AutoML e Engenharia de Features

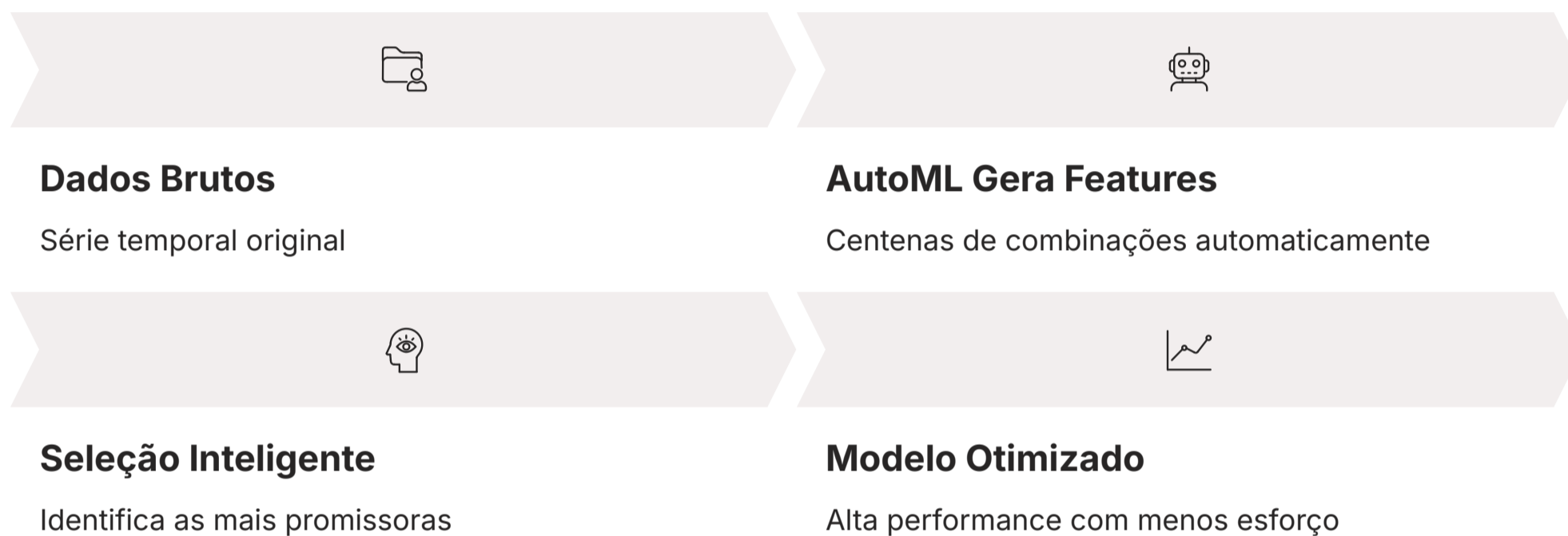
## O Desafio Tradicional

A engenharia de features, embora poderosa, é um processo que demanda:

- Tempo significativo de desenvolvimento
- Conhecimento profundo de domínio
- Muita experimentação e iteração
- Expertise técnica especializada

## A Solução AutoML

**Automação de Machine Learning (AutoML)** promete revolucionar esse processo, automatizando desde a geração até a seleção de features.



Imagine ter um assistente inteligente que, ao receber seus dados de séries temporais, automaticamente gera centenas de features de lag, rolling window e calendário, testa diferentes combinações, e até mesmo cria features mais complexas, como interações entre elas. Isso é o que o AutoML busca fazer. Ele utiliza algoritmos para explorar o espaço de features, identificando as mais promissoras e otimizando sua seleção, liberando o cientista de dados para focar em problemas de negócio mais complexos e na interpretação dos resultados.

## Ferramentas Populares

### featuretools

Python - Deep Feature Synthesis para geração automática

### Google Cloud AI

Plataforma completa com AutoML integrado

### H2O.ai

Open source com capacidades avançadas de AutoML

Essas ferramentas não apenas geram features, mas também podem realizar a seleção e o pré-processamento, garantindo que as features sejam escaladas corretamente e que o vazamento de dados seja minimizado. Embora o conhecimento humano ainda seja insubstituível para a validação e interpretação, o AutoML atua como um poderoso multiplicador de força, permitindo que equipes menores construam modelos mais sofisticados em menos tempo.

# Além da Previsão: A Importância da XAI na Engenharia de Features

## Por que o modelo fez essa previsão?

Com a crescente complexidade dos modelos de Machine Learning, especialmente aqueles que utilizam muitas features geradas automaticamente ou que são intrinsecamente "caixas-pretas" (como redes neurais e gradient boosting), surge uma necessidade crítica: entender **por que** um modelo faz uma determinada previsão. É a área da **Inteligência Artificial Explicável (XAI - Explainable AI)**, e ela tem uma conexão profunda com a engenharia de features.

### Conformidade Legal

Em setores regulados (finanças, saúde), a interpretabilidade não é luxo - é exigência legal e ética

### Confiança no Modelo

Stakeholders precisam entender as decisões para confiar e adotar o sistema de IA

### Depuração e Insights

Identificar se o modelo aprende padrões lógicos ou apenas "decora" ruídos dos dados

## Técnicas XAI Principais

### SHAP (SHapley Additive exPlanations)

Atribui valores de contribuição para cada feature, mostrando seu impacto na previsão. Baseado em teoria dos jogos, oferece explicações consistentes e justas.

- ❏ **Exemplo:** "O lag de 7 dias contribuiu +15% para a previsão, enquanto a média móvel de 30 dias contribuiu -8%"

### LIME (Local Interpretable Model-agnostic Explanations)

Cria explicações locais aproximando o modelo complexo com um modelo simples e interpretável ao redor de uma previsão específica.

- ❏ **Exemplo:** "Para esta previsão específica, o dia da semana foi o fator mais importante"

Ao analisar os valores SHAP, por exemplo, podemos ver se o lag de 7 dias ou a média móvel de 30 dias teve um impacto maior na previsão de hoje, e em que direção. Isso nos ajuda a validar a relevância das features que criamos e a identificar se o modelo está aprendendo padrões lógicos ou apenas "decorando" ruídos. A engenharia de features, combinada com a XAI, não só melhora a precisão, mas também a transparência e a confiabilidade dos sistemas de IA.

# Boas Práticas e Armadilhas a Evitar

A engenharia de features é uma ferramenta poderosa, mas como toda ferramenta, pode ser mal utilizada. Para garantir que suas features realmente melhorem seu modelo e não introduzam problemas, é fundamental seguir algumas boas práticas e estar ciente das armadilhas comuns.

1

## Evite Vazamento de Dados

**REGRA DE OURO:** Nunca use informações que não estariam disponíveis no momento da previsão. Features de janela móvel devem usar apenas dados *anteriores* ao ponto previsto.

2

## Cuidado com Overfitting

Criar muitas features específicas pode fazer o modelo memorizar ruído. Use validação cruzada temporal: conjunto de validação sempre *depois* do treinamento.

3

## Normalize as Escalas

Algoritmos como redes neurais e SVMs são sensíveis à escala. Padronize ou normalize features para que todas tenham escala semelhante.

4

## Priorize Interpretabilidade

Mesmo com AutoML, entenda as features importantes. Se algo parece ilógico, investigue - pode ser erro ou insight inesperado.

---

**Pense em um chef preparando um prato complexo.** Ele não joga todos os ingredientes na panela de uma vez. Ele seleciona os melhores, prepara-os cuidadosamente (cortando, temperando), e os adiciona no momento certo para que cada um contribua para o sabor final.

Da mesma forma, na engenharia de features, é preciso discernimento para escolher as features certas, pré-processá-las adequadamente e combiná-las de forma que potencializem o desempenho do modelo, sem introduzir vieses ou informações enganosas.

### **Lembre-se**

A engenharia de features é um **ciclo contínuo** de criação, teste, análise e refinamento. Não é um processo único, mas uma jornada iterativa de melhoria constante.

# Ferramentas e Bibliotecas para Engenharia de Features

A boa notícia é que você não precisa reinventar a roda para implementar a engenharia de features. O ecossistema de Python, em particular, oferece uma vasta gama de bibliotecas que simplificam enormemente esse processo. A escolha da ferramenta certa pode acelerar seu trabalho e garantir a robustez das suas features.

1

## Pandas

### A Base Fundamental

Indispensável para manipulação de dados. Crie lags com `.shift()`, calcule rolling com `.rolling().mean()`, acesse atributos de data com `.dt.dayofweek`

2

## scikit-learn

### Pré-processamento Robusto

Ferramentas para escalonamento, codificação de variáveis categóricas e seleção de features aplicáveis após criação das features temporais

3

## tsfresh

### Automação Poderosa

Extraí centenas de features automaticamente: estatísticas, espectrais, complexidade. Inclui seleção inteligente das mais relevantes

4

## featuretools

### Deep Feature Synthesis

AutoML para geração automática de features a partir de dados relacionais e séries temporais usando síntese profunda

---

## Exemplo Rápido com Pandas

```
# Criar feature de lag
df['vendas_lag_1'] = df['vendas'].shift(1)

# Criar rolling mean de 7 dias
df['vendas_media_7d'] = df['vendas'].rolling(window=7).mean()

# Extrair dia da semana
df['dia_semana'] = df['data'].dt.dayofweek
```

Essas ferramentas, combinadas com seu conhecimento de domínio, formam um arsenal poderoso para transformar dados brutos em insights preditivos. Dominar o uso dessas bibliotecas é um passo fundamental para se tornar um engenheiro de features eficaz.

# Estudo de Caso: Previsão de Demanda com Features Avançadas

Vamos aplicar o que aprendemos em um cenário comum: a previsão de demanda de um produto em uma loja de varejo. Nosso objetivo é prever as vendas diárias para os próximos 7 dias.

## Contexto do Problema

**Dados Iniciais:** Histórico de vendas diárias (vendas\_diarias) e a data correspondente

**Objetivo:** Prever vendas dos próximos 7 dias com alta precisão

01

### Features de Lag

- **vendas\_lag\_1:** Vendas do dia anterior
- **vendas\_lag\_7:** Vendas da semana anterior (padrão semanal)
- **vendas\_lag\_14:** Vendas de duas semanas atrás

02

### Features de Janela Móvel

- **media\_movel\_7d:** Média dos últimos 7 dias (tendência curto prazo)
- **std\_movel\_7d:** Desvio padrão dos últimos 7 dias (volatilidade)
- **max\_movel\_30d:** Venda máxima dos últimos 30 dias (picos recentes)

03

### Features Baseadas em Calendário

- **dia\_da\_semana:** 0=segunda, 6=domingo
- **mes\_do\_ano:** Número do mês (1-12)
- **dia\_do\_mes:** Dia do mês (1-31)
- **eh\_fim\_de\_semana:** Binária (1 se sábado/domingo)
- **eh\_feriado:** Binária (1 se feriado nacional)

## O Poder da Combinação

Ao combinar essas features, transformamos uma única coluna de "vendas" em um conjunto de dados multidimensional que oferece ao modelo uma visão rica e contextualizada.

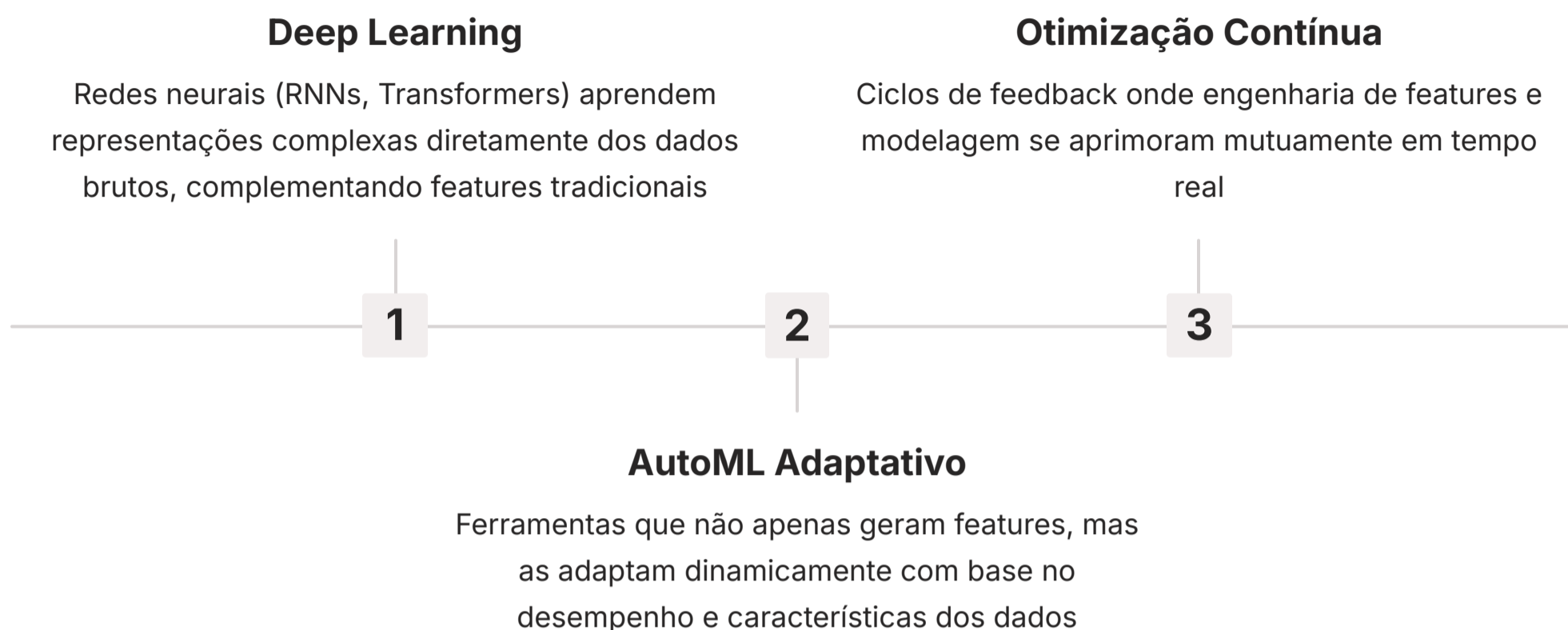
O modelo agora pode aprender que:

- As vendas são geralmente **menores nos fins de semana** (eh\_fim\_de\_semana)
- Há um **aumento gradual antes de feriados** (eh\_feriado + vendas\_lag\_X)
- A **tendência recente** (media\_movel\_7d) é um forte indicador do futuro
- Picos de **volatilidade** (std\_movel\_7d) podem indicar eventos especiais

Essa abordagem sistemática é a chave para construir modelos de previsão de demanda altamente eficazes, transformando dados simples em previsões acionáveis para o negócio.

# O Futuro da Engenharia de Features

A engenharia de features, longe de ser uma área estática, está em constante evolução. Com o avanço da pesquisa em Machine Learning e a crescente disponibilidade de dados, novas abordagens e ferramentas surgem regularmente, prometendo tornar o processo ainda mais eficiente e poderoso.



## Tendências Emergentes

### Engenharia Baseada em Deep Learning

Uma das tendências mais significativas é a **engenharia de features baseada em Deep Learning**. Redes neurais, especialmente as recorrentes (RNNs) e transformadores, têm a capacidade de aprender representações complexas e hierárquicas diretamente dos dados brutos, incluindo padrões temporais. Em vez de engenheiros humanos criarem features manualmente, o próprio modelo de Deep Learning pode "aprender" as features mais relevantes. Isso não elimina a necessidade de features tradicionais, mas complementa-as, especialmente em cenários com grandes volumes de dados e complexidade intrínseca.

### Automação Inteligente

Outra área de desenvolvimento é a **engenharia de features automatizada e adaptativa**. Ferramentas de AutoML estão se tornando mais sofisticadas, não apenas gerando features, mas também adaptando-as dinamicamente com base no desempenho do modelo e nas características dos dados. Isso inclui a otimização de parâmetros para features de janela móvel (qual o tamanho ideal da janela?) e a identificação de interações não lineares entre features.

**O futuro aponta para uma colaboração cada vez maior entre a inteligência humana e a artificial.** O conhecimento de domínio do especialista continuará sendo crucial para guiar o processo e interpretar os resultados, enquanto as ferramentas automatizadas e os modelos de Deep Learning assumirão o trabalho pesado de exploração e otimização.

Em última análise, o futuro da engenharia de features aponta para uma colaboração cada vez maior entre a inteligência humana e a artificial. O conhecimento de domínio do especialista continuará sendo crucial para guiar o processo e interpretar os resultados, enquanto as ferramentas automatizadas e os modelos de Deep Learning assumirão o trabalho pesado de exploração e otimização, permitindo que alcancemos níveis de precisão e insights nunca antes vistos.

# Consolidação e Próximos Passos

Nesta aula, mergulhamos no universo da Engenharia de Features para Séries Temporais, uma etapa crucial para transformar dados brutos em informações preditivas valiosas. Exploramos como as features de lag capturam a memória do passado, as features de janela móvel sumarizam tendências e volatilidade, e as features baseadas em calendário codificam padrões sazonais. Vimos também a importância do AutoML para automatizar e acelerar esse processo, e como a XAI nos ajuda a entender a contribuição de cada feature, construindo modelos mais transparentes e confiáveis.

## Em Prática

1. Comece com features simples (lag 1, dia da semana)
2. Monitore o desempenho do modelo
3. Adicione complexidade gradualmente
4. Use validação cruzada temporal
5. Aplique conhecimento de domínio

## Próxima Aula

### Aula 31 – Usando Modelos de Machine Learning para Previsão

Você aprenderá a aplicar os modelos de Machine Learning mais adequados para séries temporais, utilizando as features que você aprendeu a construir hoje.

## Autoavaliação

1. **Qual o principal objetivo da Engenharia de Features em séries temporais?**
  - o a) Reduzir o número de dados para acelerar o processamento.
  - o b) Transformar dados brutos em informações mais preditivas para modelos de Machine Learning.
  - o c) Apenas visualizar padrões sazonais nos dados.
  - o d) Eliminar completamente a necessidade de modelos de Machine Learning.
2. **Uma feature de lag de 7 dias para uma série temporal diária de vendas significa:**
  - o a) A média das vendas dos últimos 7 dias.
  - o b) O valor das vendas de 7 dias no futuro.
  - o c) O valor das vendas do mesmo dia da semana anterior.
  - o d) A soma das vendas dos últimos 7 dias.
3. **Qual tipo de feature seria mais adequado para capturar a volatilidade recente de uma série temporal?**
  - o a) Features de calendário (ex: mês do ano).
  - o b) Features de lag (ex: valor do dia anterior).
  - o c) Features de janela móvel (ex: desvio padrão móvel).
  - o d) Features de texto (ex: descrição do evento).
4. **A principal preocupação ao criar features para evitar o "data leakage" é:**
  - o a) Usar apenas features de calendário.
  - o b) Garantir que as features sejam calculadas apenas com dados disponíveis no momento da previsão.
  - o c) Criar o máximo de features possível.
  - o d) Ignorar a ordem temporal dos dados.

## Gabarito

1. b) | 2. c) | 3. c) | 4. b)

## Questão Discursiva

Explique como a integração de técnicas de Inteligência Artificial Explicável (XAI) pode beneficiar o processo de Engenharia de Features para Séries Temporais, especialmente em contextos onde a interpretabilidade do modelo é crucial.

## Recursos Adicionais

- **Documentação Pandas:** Para aprofundar nas funções de manipulação de séries temporais.
- **Artigos sobre tsfresh e featuretools:** Para explorar a automação da engenharia de features.
- **Introdução a SHAP e LIME:** Para entender como interpretar a contribuição das suas features.

**NOTA IMPORTANTE:** As informações regulatórias/legais/técnicas desta aula estão atualizadas até 2025. Consulte sempre fontes oficiais para verificar alterações.