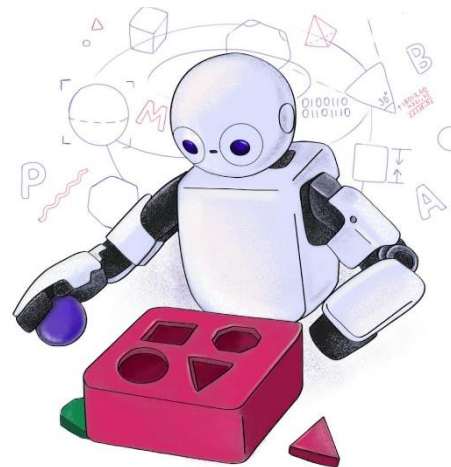


TP557 - Tópicos avançados em IoT e  
Machine Learning:

# *O Paradigma do Aprendizado de Máquina*



***Inatel***

Felipe Augusto Pereira de Figueiredo  
felipe.figueiredo@inatel.br

# O que vamos ver?

- Neste tópico vamos explorar o que o **aprendizado de máquina** realmente é em um nível mais fundamental.
- Basicamente, o que vamos discutir é um **novο paradigma**, onde **ao invés de programar uma solução** para um determinado problema, vamos **ensinar um computador a aprender a solução** através de experiências prévias.
- Esse novo paradigma tem o potencial para **resolver problemas que não podem ou são muito difíceis de serem resolvidos programaticamente**.

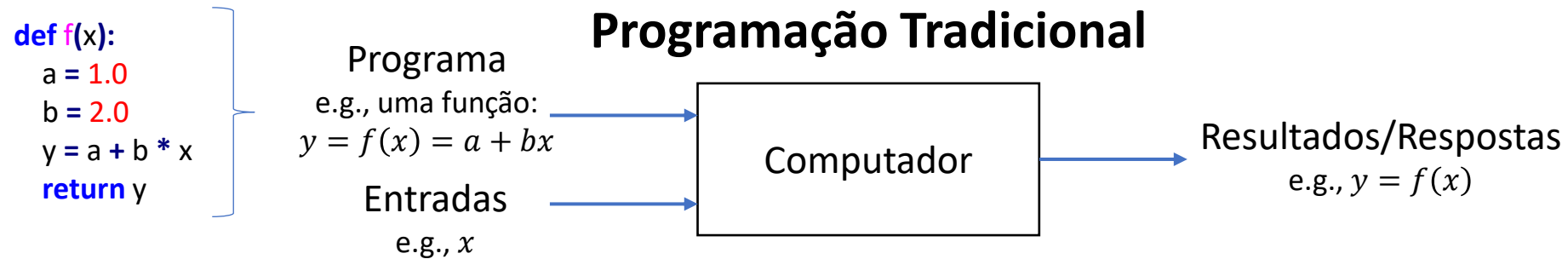
# O que é o aprendizado de máquina?



- É uma das subáreas da inteligência artificial.
- O termo foi cunhado em 1959, pelo cientista da computação Arthur Samuel, que o definiu como o  
*“Campo de estudo que dá aos computadores a habilidade de **aprender sem serem explicitamente programados.**”*
- Mas como eles aprendem?
  - Através de **experiências prévias, induz-se** conhecimento nas máquinas.
- Algoritmos de ML são **orientados a dados**, i.e., eles **aprendem automaticamente** (através de treinamento) uma **solução geral** a partir de **conjuntos de dados** fornecidos a eles.

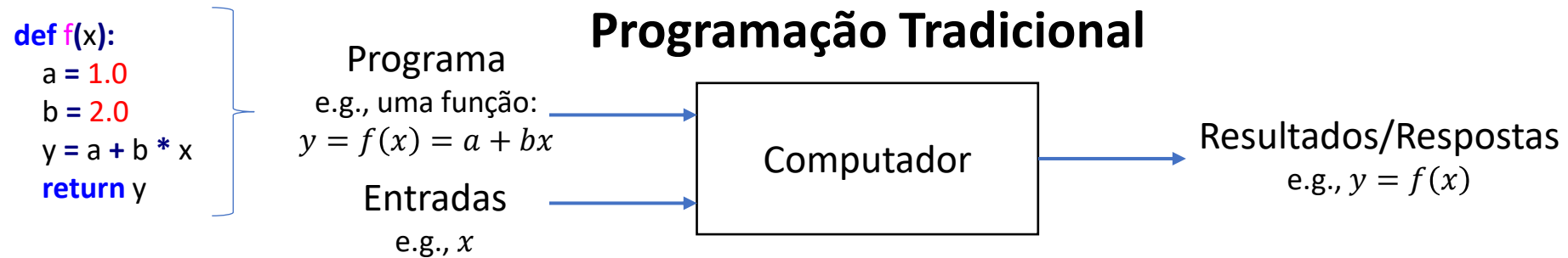


# Programação tradicional



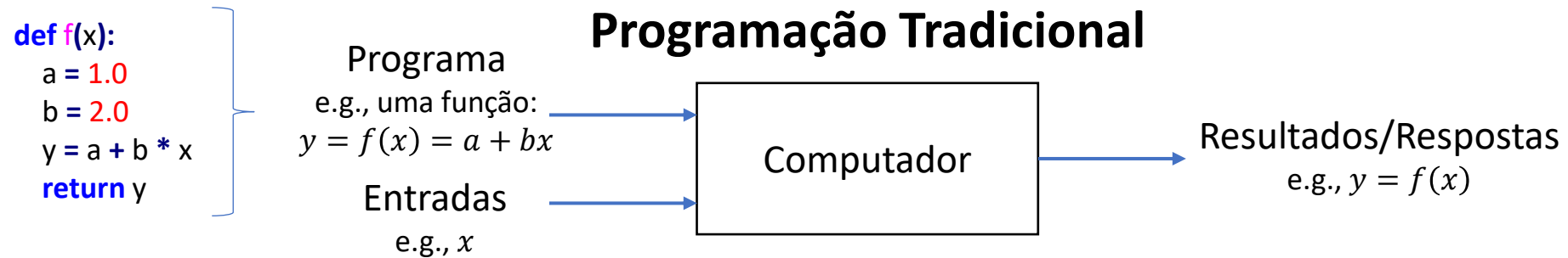
- Na programação tradicional, o **programador analisa um problema e cria um código** (ou programa) para resolvê-lo.
  - Código: sequência de regras que definem o comportamento do programa.
- Na sequência, o **computador recebe o código e os dados** (i.e., entradas), **o aplica aos dados e retorna os valores de saída**.

# Programação tradicional



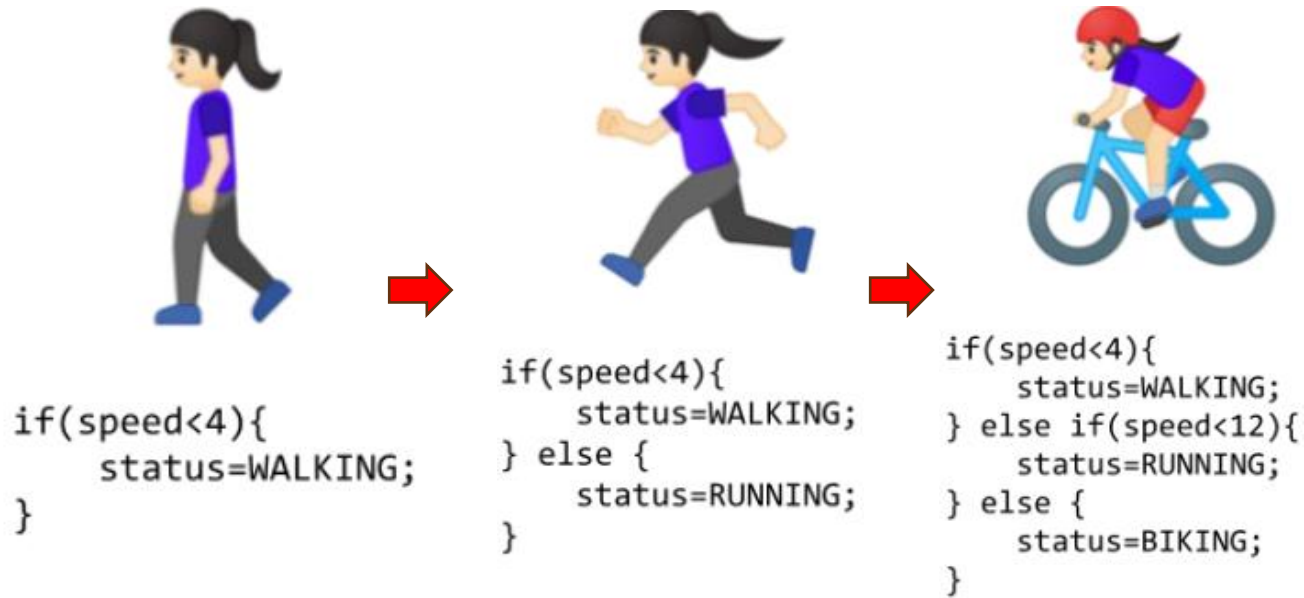
- Em outras palavras, na programação tradicional, o **programador cria as regras** (i.e., programa) que **mapeiam as entradas nas saídas**.

# Programação tradicional



- Porém, em alguns casos é muito difícil criar um código para solucionar um problema de forma geral.
- Vamos ver uma situação onde esse paradigma não funciona bem.

# Programação tradicional



- Vamos supor que queremos criar uma aplicação para celular/*smart watch* que ***detecte atividades*** como ***andar***, ***correr*** e ***pedalar***.
- Nesse caso, o programador analisaria o problema e diria que podemos usar a ***velocidade*** e criar algumas ***regras*** para ***diferenciar as atividades***.

# Programação tradicional



```
if(speed<4){  
    status=WALKING;  
}
```



```
if(speed<4){  
    status=WALKING;  
} else {  
    status=RUNNING;  
}
```



```
if(speed<4){  
    status=WALKING;  
} else if(speed<12){  
    status=RUNNING;  
} else {  
    status=BIKING;  
}
```



// ???

- Mas e se quisermos estender a aplicação para identificar que o usuário está *jogando golfe*, como poderíamos identificar essa atividade?



# Programação tradicional



```
if(speed<4){  
    status=WALKING;  
}
```



```
if(speed<4){  
    status=WALKING;  
} else {  
    status=RUNNING;  
}
```



```
if(speed<4){  
    status=WALKING;  
} else if(speed<12){  
    status=RUNNING;  
} else {  
    status=BIKING;  
}
```

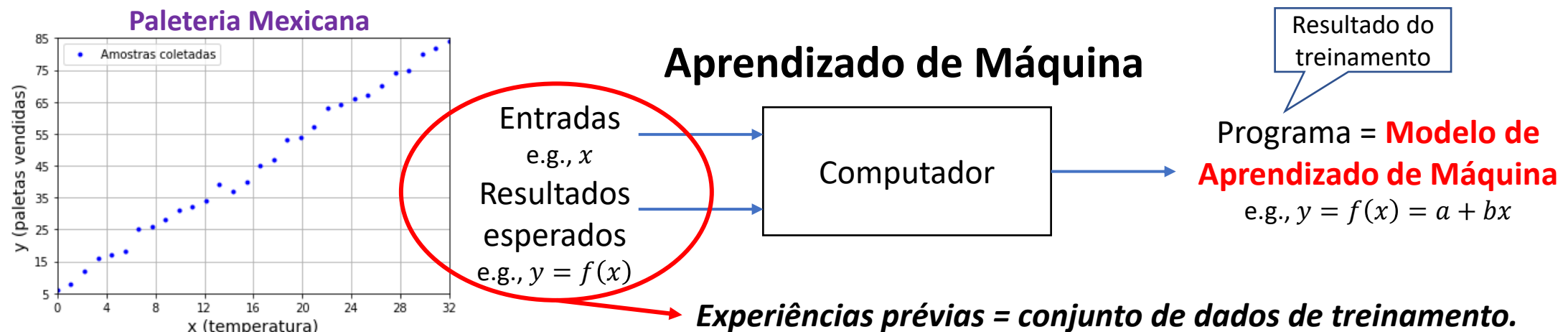


// ???

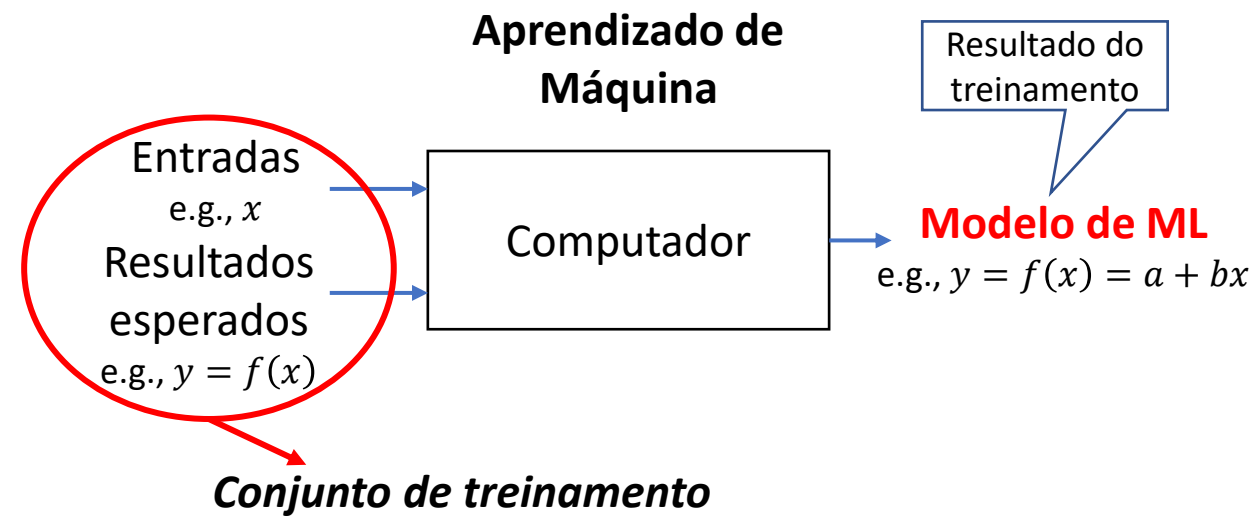
- Além disso, vejam que as *regras* que criamos são bem *simples e com certeza gerariam vários erros de identificação*.
  - Por exemplo, podemos correr ladeira abaixo mais rápido do que pedalamos ladeira acima.
- Aprendizado de máquina pode nos ajudar a resolver este problema.

# O paradigma do aprendizado de máquina

- “... *aprender sem serem explicitamente programados*.”
- Esse trecho pode ser entendido se reorganizarmos a figura anterior.
  - E se ao invés de tentarmos **descobrir as regras que agem sobre os dados para gerar as respostas**, fizermos o **contrário**? Ou seja, **fornecer as respostas e os dados a um computador e deixar que ele descubra as regras** que geram as saídas (i.e., um mapeamento das entradas nas respostas)?



# O paradigma do aprendizado de máquina



- Como a máquina aprende?
  - Através de treinamento com um **conjunto de experiências prévias** (chamado de **conjunto de treinamento**).
  - Durante o treinamento, o **conjunto de treinamento** é **apresentado ao modelo diversas vezes** (chamada de **épocas**).
  - Com isso, o modelo vai **iterativamente se atualizando** e **aprendendo** um **mapeamento** das entradas nas saídas esperadas.

# O paradigma do aprendizado de máquina



```
0101001010100101010  
1001010101001011101  
0100101010010101001  
0101001010100101010
```

Label = WALKING



```
1010100101001010101  
0101010010010010001  
0010011111010101111  
1010100100111101011
```

Label = RUNNING



```
1001010011111010101  
1101010111010101110  
1010101111010101011  
1111110001111010101
```

Label = BIKING



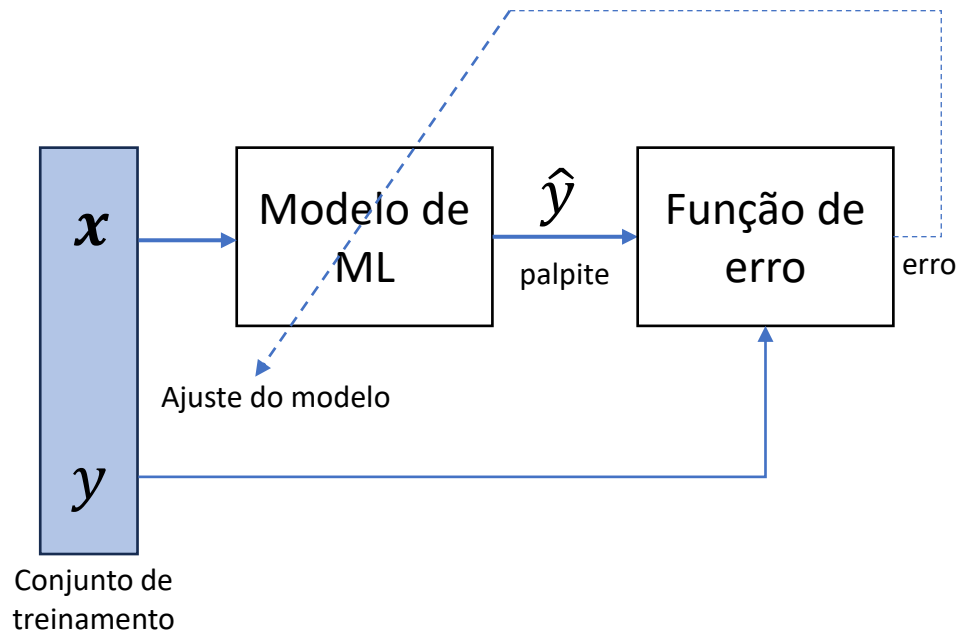
```
1111111111010011101  
0011111010111110101  
0101110101010101110  
1010101010100111110
```

Label = GOLFING

O computador aprende, através do seu treinamento, *padrões* nos dados de diferentes sensores que *podem ser mapeados nas atividades*.

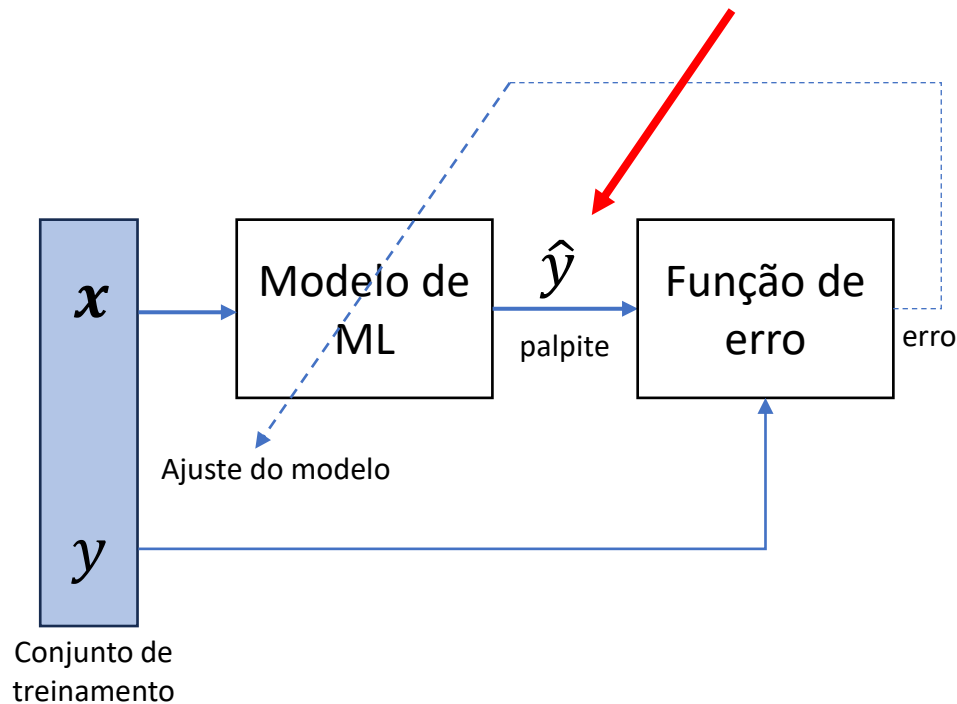
- No caso da nossa aplicação, podemos *coletar informações de sensores* diferentes e *rotulá-las* (i.e., saídas esperadas) com a atividade do usuário.
- Usando este *conjunto de dados* (i.e., dados dos sensores e rótulos), o computador pode *descobrir regras que identificam as atividades*: caminhar, correr, pedalar e até mesmo jogar golfe.

# Treinamento da máquina (ou modelo de ML)



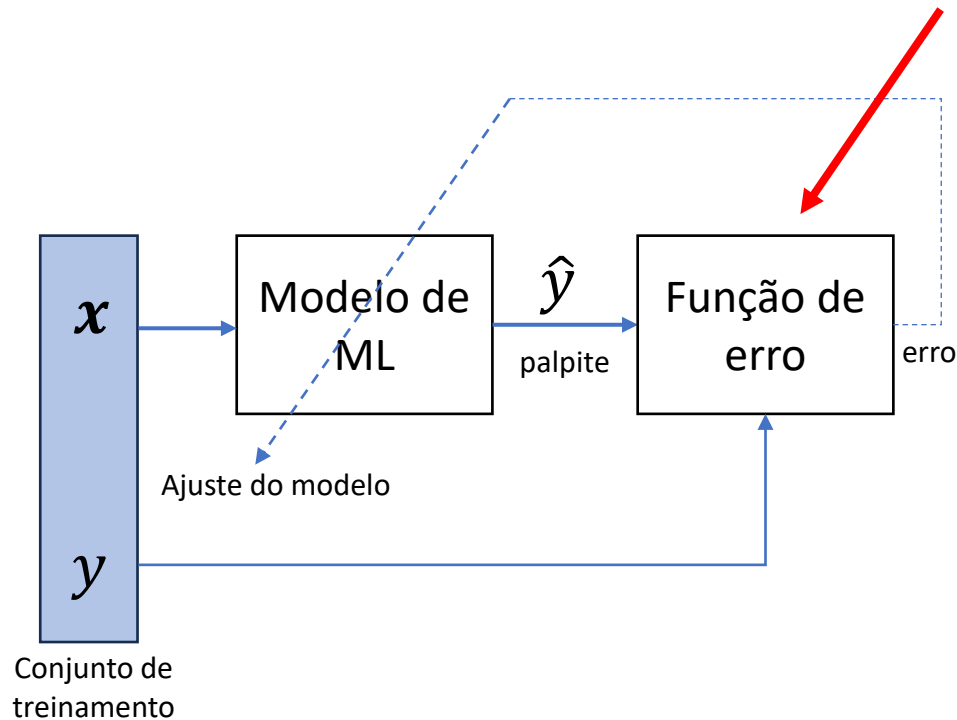
- Vamos ver como o treinamento da máquina (i.e., dos modelos) funciona em alto nível.
- Essa explicação em alto nível nos **dará a base** para o entendimento do **algoritmo** por trás do **treinamento de muitos modelos de ML**.

# Treinamento da máquina (ou modelo de ML)



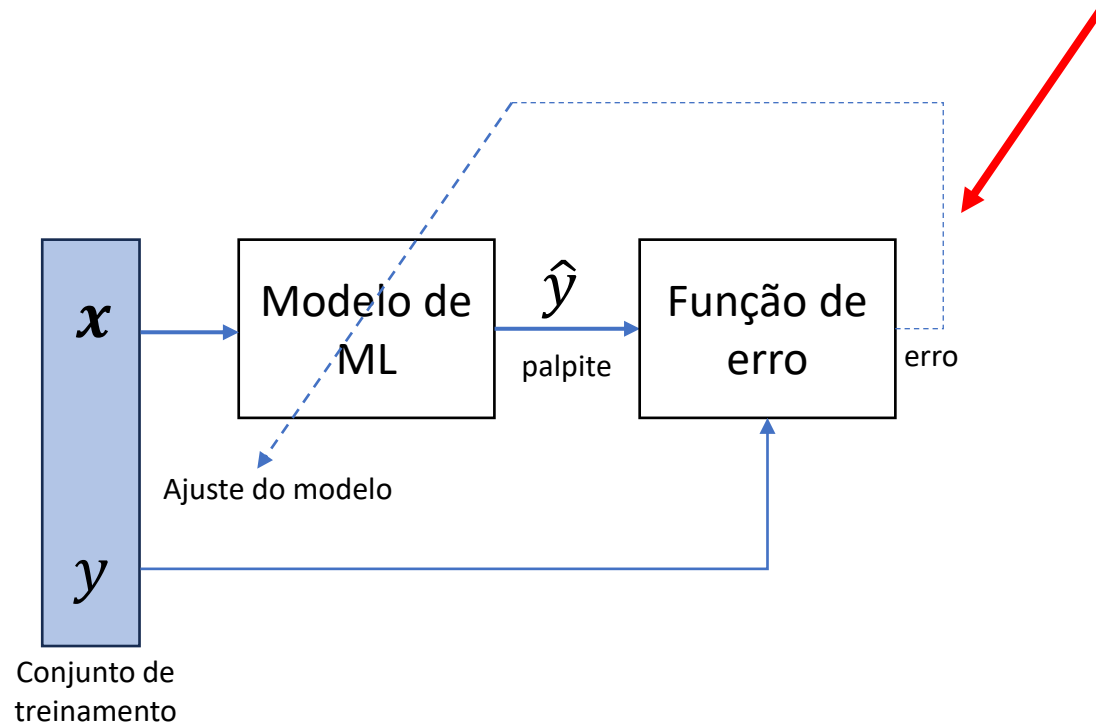
- Primeiro, o computador (ou modelo de ML) faz um **mapeamento aleatório** da entrada,  $x$ , em um valor de saída,  $\hat{y}$ .
- Ou seja, o modelo dá um **palpite** sobre qual deve ser a saída para aquela entrada.

# Treinamento da máquina (ou modelo de ML)



- Em seguida, usando as saídas esperadas,  $y$ , (i.e., rótulos), *mede-se o quão bom ou ruim foi esse palpite.*
- Medimos a qualidade do palpite usando uma função chamada de *função de perda, erro ou custo.*

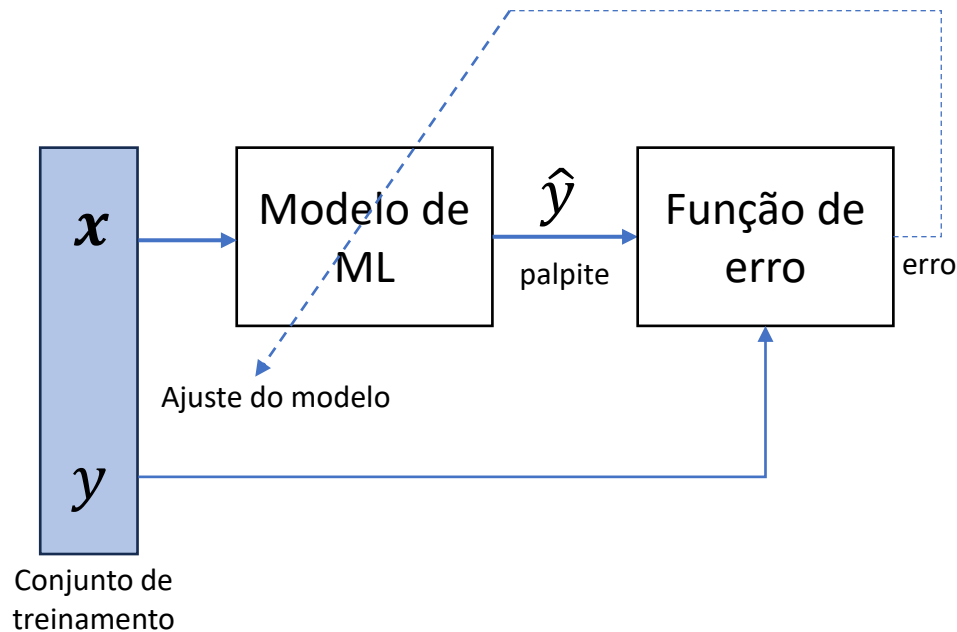
# Treinamento da máquina (ou modelo de ML)



- Na sequência, o **erro** é usado para **otimizar o modelo** e, com isso, **melhorar o próximo palpite**.
  - Por exemplo, em uma rede neural, atualizar seus pesos.
- Como veremos, a informação fornecida pelo **erro** é de **suma importância para o treinamento de modelos de aprendizado supervisionados**.

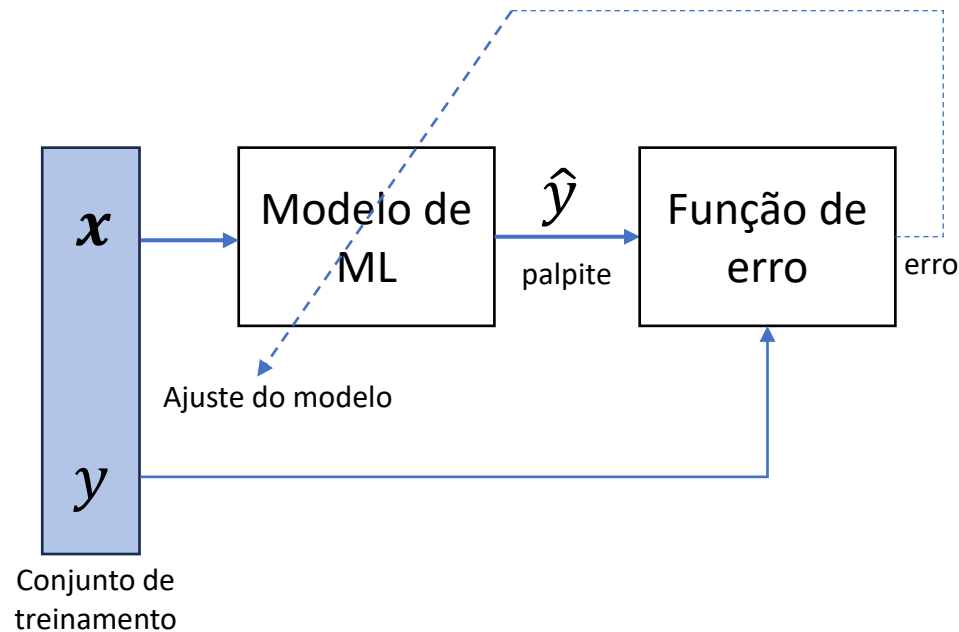


# Treinamento da máquina (ou modelo de ML)



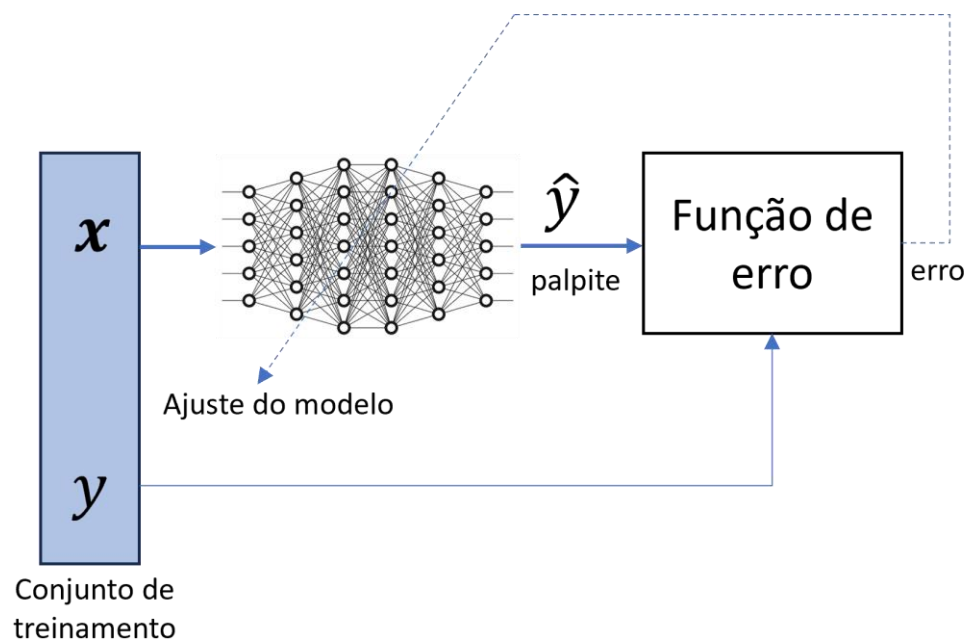
- Esse processo é repetido até que o erro seja **minimizado**.
- A **ideia** é que a cada **repetição** (ou **iteração**), o **novo palpite** se torne **melhor do que o anterior**, fazendo com que o **erro diminua** e o **modelo se torne mais preciso**.

# Treinamento da máquina (ou modelo de ML)



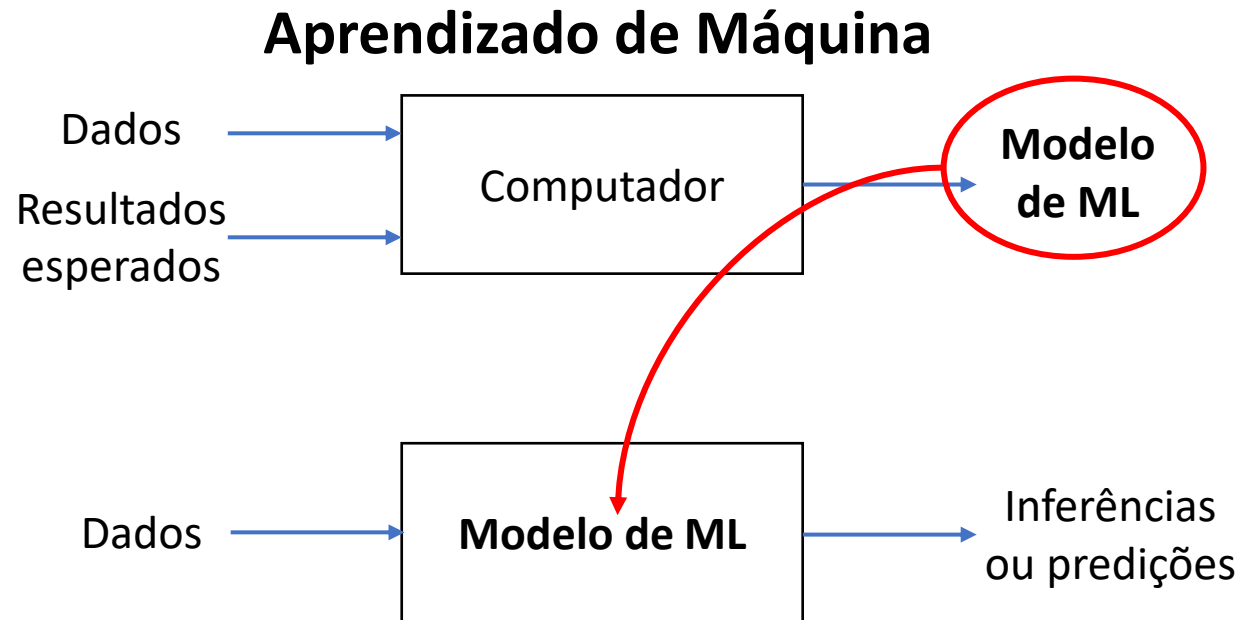
- Percebam que a ***solução*** para o problema é ***encontrada com base em experiências prévias***, ou seja, com o conjunto de treinamento (i.e., entradas,  $x$ , e saídas esperadas,  $y$ ).
- Isso é chamado de ***raciocínio indutivo***, que é um processo pelo qual chega-se a ***conclusões gerais*** a partir de ***experiências passadas***.

# Treinamento da máquina (ou modelo de ML)



- Por exemplo, o modelo de ML pode ser uma **rede neural** onde o **processo de treinamento atualiza** os seus **pesos** (sinápticos e de bias) a fim de **minimizar o erro** e, consequentemente, **resolver problemas classificação** (e.g., de imagens), **regressão** (i.e., aproximação de curvas), etc.

# Inferência



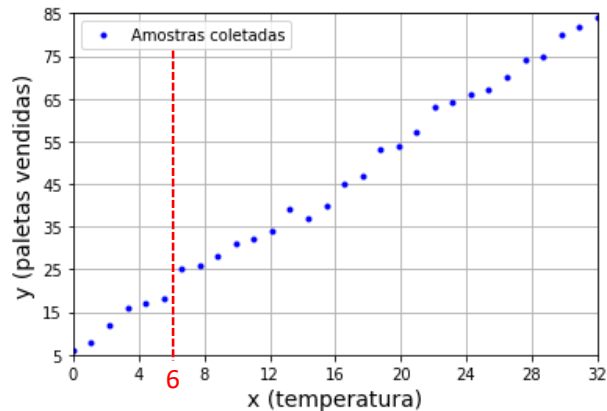
- Após o treinamento do modelo, o usamos para fazer *inferências*.
- Ou seja, o usamos para fazer *previsões* (i.e., valores de saídas) *para entradas inéditas*.

# Generalização

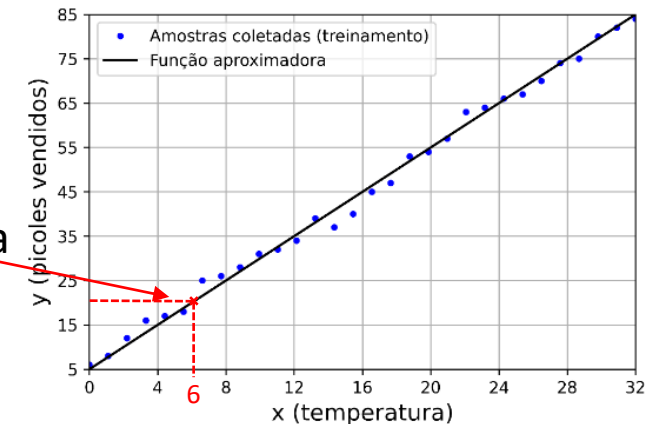
- Porém, não basta que o algoritmo de ML treine um modelo que faça um bom mapeamento ***apenas para os dados do conjunto de treinamento***.
- O ***algoritmo*** de ML deve ***treinar*** um ***modelo*** que ***aprenda*** uma ***solução geral***, ou seja, um modelo que ***faça previsões precisas para entradas não vistas durante o treinamento***.
- Essa extrapolação do conhecimento adquirido durante o treinamento é chamado de ***generalização***.



# Generalização: Paeteria Mexicana

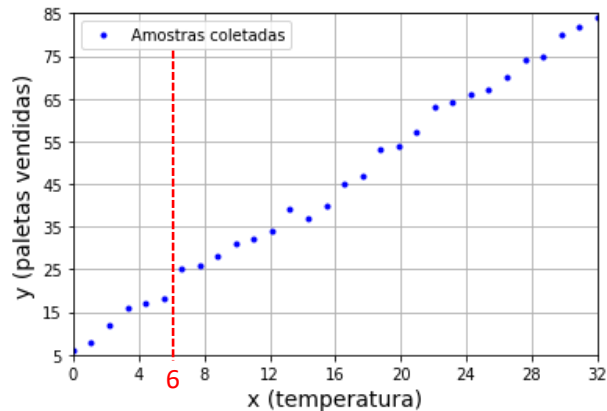


*Qual é a estimativa de paletas vendidas quando a temperatura é de  $\approx 6$  graus (valor não visto durante o treinamento)?*



*A partir do mapeamento aprendido (reta), o modelo gera como saída o valor 21, que é coerente com o restante dos dados.*

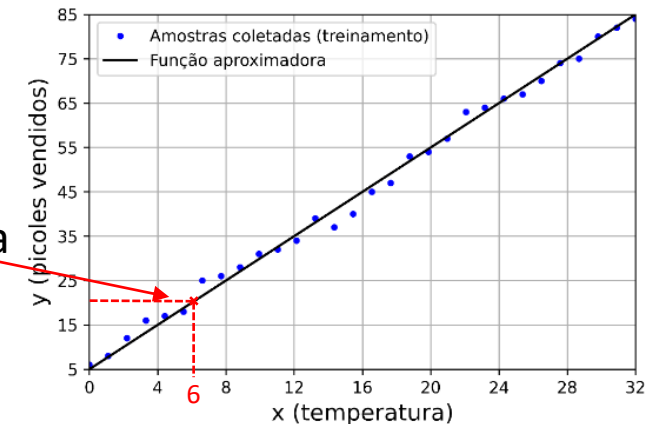
# Generalização: Paeteria Mexicana



Qual é a estimativa de paletas vendidas quando a temperatura é de  $\approx 6$  graus (valor não visto durante o treinamento)?



O modelo aprendeu que uma reta é um bom mapeamento.



A partir do mapeamento aprendido (reta), o modelo gera como saída o valor 21, que é coerente com o restante dos dados.

No sentido da minimização do erro **E** de ter uma boa capacidade de generalização.

# Atividades

- Quiz: “***TP557 - O Paradigma do Aprendizado de Máquina***”.



Perguntas?

Obrigado!

