



INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

AULA 05 – Aprendizado Supervisionado

O QUE IREMOS APRENDER

01

O QUE É APRENDIZADO SUPERVISIONADO?

02

ONDE É UTILIZADO?

03

CLASSIFICAÇÃO

04

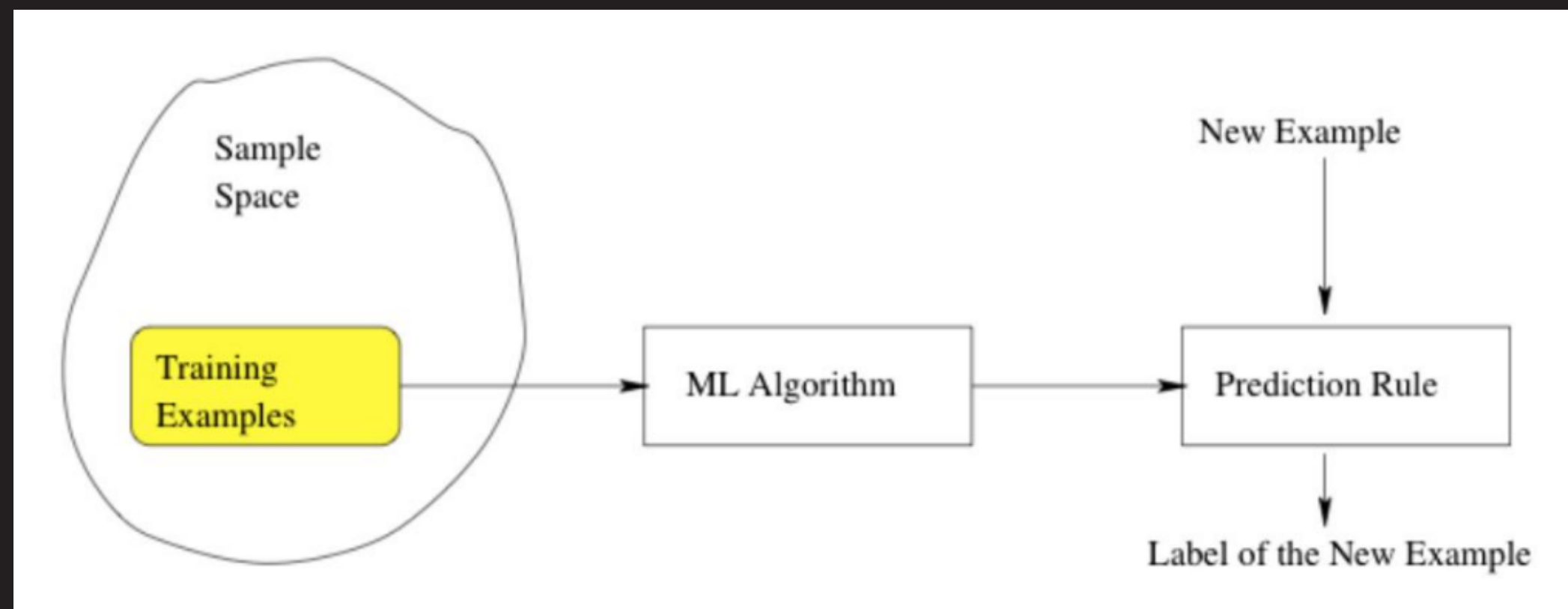
REGRESSÃO

O Que é Aprendizado Supervisionado?

O aprendizado supervisionado é um tipo de técnica de aprendizado de máquina em que um algoritmo é treinado em um conjunto de dados rotulados, ou seja, dados para os quais as respostas corretas já são conhecidas. Durante o treinamento, o algoritmo aprende a mapear as entradas para as saídas desejadas com base nos exemplos fornecidos.



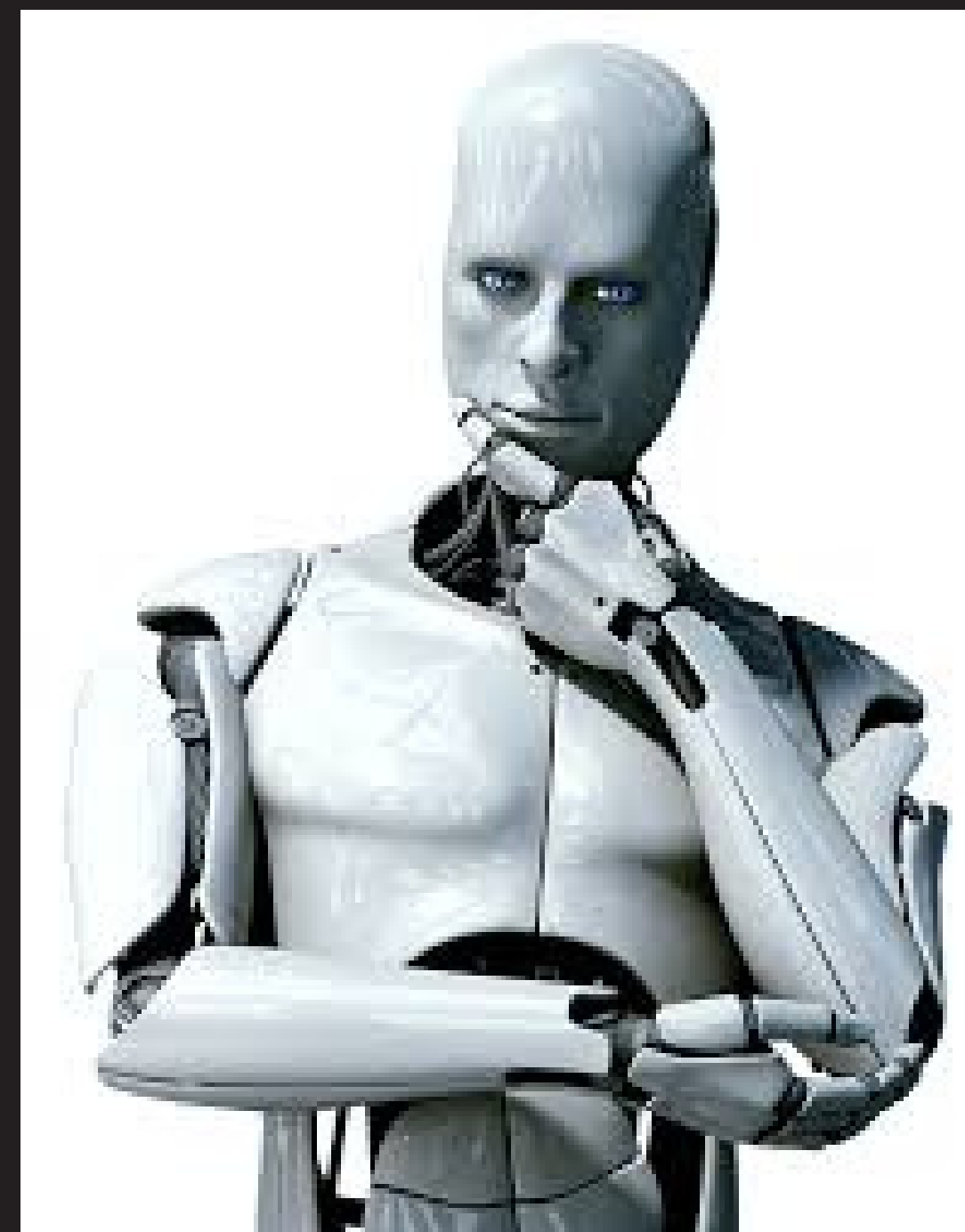
O Que é Aprendizado Supervisionado?



O processo de treinamento envolve alimentar o algoritmo com pares de entrada e saída, permitindo que ele ajuste seus parâmetros internos para minimizar a diferença entre as saídas previstas e as saídas reais. Uma vez treinado, o modelo pode ser usado para prever a saída correspondente para novos dados de entrada.

Onde é Utilizado?

É frequentemente utilizado em uma variedade de problemas de previsão e classificação, como reconhecimento de imagens, tradução automática, diagnóstico médico, detecção de fraudes, entre outros. É uma abordagem muito poderosa e amplamente utilizada em aprendizado de máquina devido à sua capacidade de fazer previsões precisas quando fornecido um conjunto de dados de treinamento adequado.



Onde é Utilizado?

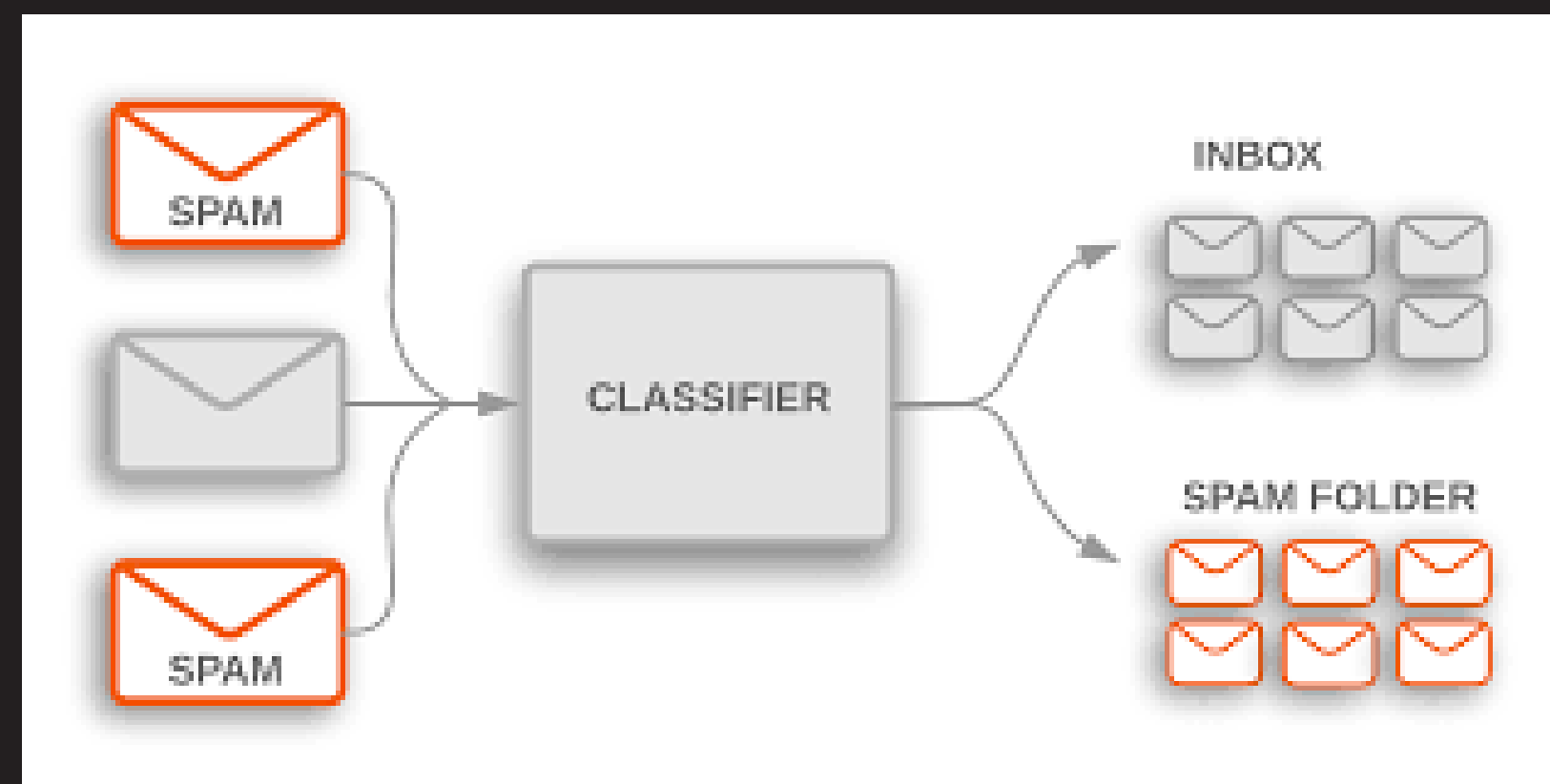
As grandes empresas fazem a utilização do aprendizado, são elas:

1. Google: Usa para melhorar a busca na internet e traduzir idiomas.
2. Facebook: Personaliza o que você vê no seu feed, identifica spam e reconhece rostos em fotos.
3. Amazon: Recomenda produtos com base no que você comprou antes e prevê o que as pessoas vão comprar.
4. Netflix: Sugere filmes e séries com base no que você assistiu antes.
5. Uber: Prevê quando e onde as pessoas vão precisar de um carro e detecta atividades suspeitas.



Classificação

A classificação é uma técnica usada para categorizar ou rotular dados em diferentes grupos ou classes. É como ensinar a um computador a reconhecer padrões e fazer previsões com base nessas informações. Por exemplo, ela pode ser usada para prever se um e-mail é spam ou não spam, se uma transação é fraudulenta ou legítima, ou até mesmo para diagnosticar uma doença com base nos sintomas de um paciente.



Classificação

Suponha que temos um conjunto de dados que contém informações sobre clientes, incluindo idade, renda e se eles compraram ou não um produto. Queremos construir um modelo de classificação que possa prever se um novo cliente comprará ou não o produto com base em sua idade e renda:



```
#Bibliotecas necessarias
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```


Classificação

Criação do Data Frame

```
import pandas as pd

# Criar um DataFrame de exemplo
data = {
    'idade': [25, 35, 45, 20, 30, 50, 40, 55, 60, 70],
    'renda': [3000, 4000, 5000, 2000, 3500, 6000, 5500, 7000, 8000, 9000],
    'compra': ['Nao', 'Nao', 'Sim', 'Nao', 'Sim', 'Sim', 'Sim', 'Sim', 'Sim', 'Sim']
}

data = pd.DataFrame(data)
```

Classificação

Treino do modelo que possui regressão logística que é um tipo de modelo de regressão usado para problemas de classificação, onde queremos prever a probabilidade de uma observação pertencer a uma determinada classe. Apesar do nome, ela é usada para tarefas de classificação, não para regressão

```
# Separar os dados em variáveis independentes (idade e renda) e variável dependente (compra)
X = data[['idade', 'renda']]
y = data['compra']

# Dividir os dados em conjunto de treinamento e teste
X_treino, X_teste, y_treino, y_teste = train_test_split(X, y, test_size=0.2, random_state=42)

# Instanciar e treinar o modelo de regressão logística
modelo = LogisticRegression()
modelo.fit(X_treino, y_treino)
```

Classificação

dividimos os dados em conjuntos de treinamento e teste, e então treinamos o modelo com os dados de treinamento. Depois, fazemos previsões nos dados de teste e avaliamos a precisão do modelo. Essa precisão nos dirá quão bem o modelo está prevendo se um cliente vai comprar ou não um produto com base em sua idade e renda.

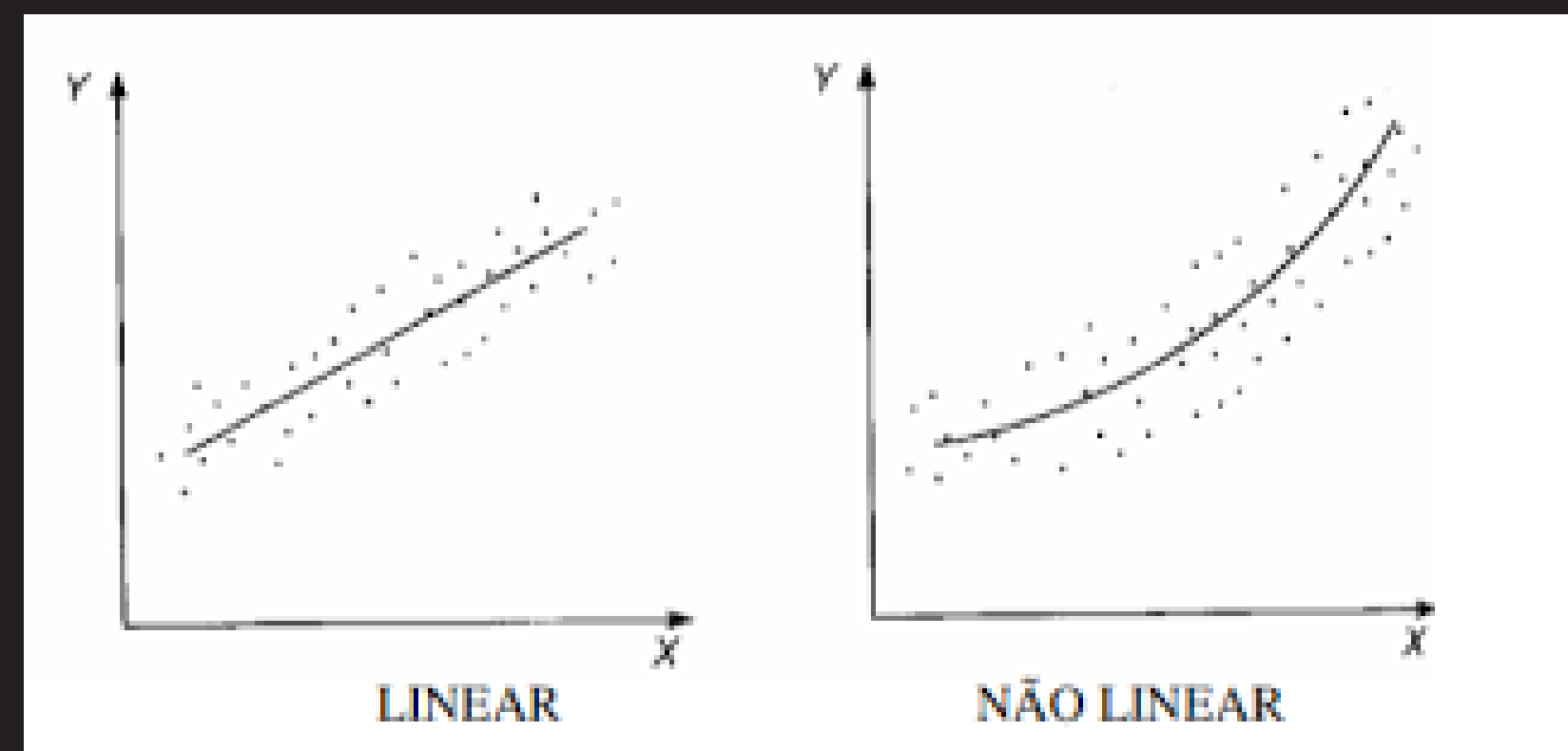
```
[10] # Fazer previsões no conjunto de teste
      previsoes = modelo.predict(X_teste)

      # Avaliar a precisão do modelo
      precisao = accuracy_score(y_teste, previsoes)
      print("Precisão do modelo:", precisao)
```

Regressão

A regressão é uma técnica estatística utilizada para entender a relação entre uma variável dependente (ou resposta) e uma ou mais variáveis independentes (ou preditoras). Seu principal objetivo é modelar essa relação de forma a fazer previsões ou estimativas da variável dependente com base nas variáveis independentes.

A regressão é amplamente utilizada no aprendizado supervisionado porque seu objetivo é prever ou estimar um resultado a partir de variáveis de entrada conhecidas.



Regressão

Linear

- É como uma linha que tenta se ajustar aos pontos de dados em um gráfico.
- Usada para prever ou estimar um valor com base em outra variável.
- Por exemplo, pode prever a altura de uma pessoa com base na idade dela.
- Encontra a linha que minimiza as diferenças entre os pontos reais e a linha.

Não Linear

- Funciona como a regressão linear, mas não se limita a uma linha reta.
- Pode se ajustar a curvas ou padrões mais complexos nos dados.
- Útil quando a relação entre as variáveis não é simplesmente linear.
- Por exemplo, pode modelar o crescimento de uma planta ao longo do tempo, que não é linear.
- É mais flexível, mas pode ser mais complexa de entender e interpretar.

Regressão

Vamos supor que temos uma loja de e-commerce e queremos prever as vendas com base nos gastos em marketing, temos aqui um exemplo de projeto:



```
#Importação de bibliotecas necessarias para resolução da ocasião
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
```


Regressão

Agora iremos fazer um Data Frame e separar as colunas em variaveis



```
# Criar um DataFrame com dados de gastos em marketing e vendas
data = {
    'Gastos_Marketing': [100, 200, 300, 400, 500],
    'Vendas': [50, 100, 150, 200, 250]
}
df = pd.DataFrame(data)

# Visualizar o DataFrame
print(df)

# Separar os dados em variáveis independentes (X) e variável dependente (y)
X = df[['Gastos_Marketing']]
y = df['Vendas']
```

Regressão

A conclusão obtida é que o modelo foi capaz de capturar a relação entre os gastos em marketing e as vendas da loja. As previsões do modelo são altamente precisas, como evidenciado pelo erro médio quadrático muito baixo. Com base nos resultados do modelo, podemos concluir que há uma forte relação entre os gastos em marketing e as vendas, e que investir mais em marketing provavelmente resultará em um aumento nas vendas.

```
[4] # Dividir os dados em conjunto de treinamento e teste
X_treino, X_teste, y_treino, y_teste = train_test_split(X, y, test_size=0.2, random_state=42)

# Instanciar e treinar o modelo de regressão linear
modelo = LinearRegression()
modelo.fit(X_treino, y_treino)

# Fazer previsões no conjunto de teste
previsoes = modelo.predict(X_teste)

# Avaliar o desempenho do modelo usando o erro médio quadrático (MSE)
mse = mean_squared_error(y_teste, previsoes)
print("Erro médio quadrático (MSE):", mse)
```

Regressão

O Erro Médio Quadrático (MSE) é uma medida que usamos para entender quão boas são as previsões feitas por um modelo em relação aos valores reais. O valor ideal para o MSE pode variar dependendo do contexto do problema e das características dos dados. Em geral, quanto mais próximo de zero, melhor é o desempenho do modelo. Se o (MSE) retornou 0.0, isso geralmente significa que o modelo foi capaz de fazer previsões perfeitas nos dados de teste. No entanto, isso é bastante incomum na prática e pode indicar algum tipo de problema ex: Overfitting.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

The equation is annotated with blue boxes and labels: 'Mean' above the fraction, 'Error' above the difference term, and 'Squared' above the exponent.

SE LIGA NO CONTEÚDO DA PRÓXIMA AULA!

AULA 09 – DATA SCIENCE PRO:
Aprendizado Não Supervisionado



IN

INFINITY SCHOOL

VISUAL ART CREATIVE CENTER



INFINITY SCHOOL

VISUAL ART CREATIVE CENTER