

Criando uma máquina de aprendizado simples

```
import random
import math
import json

def sigmoide(x):
    return 1.0 / (1.0 + math.exp(-x))

class ModeloDeAprendizado:
    def __init__(self, num_entradas=None, num_saidas=None):
        if num_entradas is not None and num_saidas is not None:
            self.num_entradas = num_entradas
            self.num_saidas = num_saidas
            self.pesos = []
            for _ in range(num_saidas):
                self.pesos.append([random.uniform(-1.0, 1.0) for _ in range(num_entradas)])
            self.vieses = [random.uniform(-1.0, 1.0) for _ in range(num_saidas)]

    def prever(self, entradas):
        if len(entradas) != self.num_entradas:
            raise ValueError("O número de entradas fornecido não bate com a arquitetura.")
        saidas = []
        for i in range(self.num_saidas):
            soma = self.vieses[i]
            for j in range(self.num_entradas):
                soma += entradas[j] * self.pesos[i][j]
            saidas.append(sigmoide(soma))
        return saidas

    def salvar_modelo(self, nome_arquivo):
        dados_do_modelo = {
            "num_entradas": self.num_entradas,
            "num_saidas": self.num_saidas,
            "pesos": self.pesos,
            "vieses": self.vieses
        }
        with open(nome_arquivo, 'w') as arquivo:
            json.dump(dados_do_modelo, arquivo, indent=4)

    def carregar_modelo(self, nome_arquivo):
        with open(nome_arquivo, 'r') as arquivo:
            dados_do_modelo = json.load(arquivo)

            self.num_entradas = dados_do_modelo["num_entradas"]
            self.num_saidas = dados_do_modelo["num_saidas"]
            self.pesos = dados_do_modelo["pesos"]
            self.vieses = dados_do_modelo["vieses"]

meu_modelo = ModeloDeAprendizado(num_entradas=3, num_saidas=2)
meu_modelo.salvar_modelo("meu_primeiro_modelo.json")
print("Modelo salvo com sucesso no arquivo 'meu_primeiro_modelo.json'!")
```

```
modelo_recuperado = ModeloDeAprendizado()
modelo_recuperado.carregar_modelo("meu_primeiro_modelo.json")

leitura_sensores = [0.8, 0.2, 0.9]
decisao = modelo_recuperado.prever(leitura_sensores)
print(f"Decisão do modelo carregado: {[round(d, 3) for d in decisao]}")
```