PYTHON

Com exemplos em AI

OBJETIVOS

- Introduzir a linguagem python
- Como usar o python para ler arquivos
- Usar os dados extraídos para executar modelos de ML
- Base para futuros estudos

ROTEIRO

SINTAXE PYTHON — Introdução do básico de sintaxe de Python.

INTRODUÇÃO À ANÁLISE DE DADOS COM PYTHON — Introdução as principais ferramentas de análise de dados

INTRODUÇÃO A MACHINE LEARNING COM PYTHON — Introdução à modelos básicos de aprendizado de máquina em python

DESAFIO PRÁTICO — Usar o aprendizado minicurso para construir o próprio modelo

O QUE É PYTHON

Alto Nível

Único produto especificamente dedicado a este nicho de mercado

Interpretada

O código em python é executado linha por linha

Orientada à Objetos

Possibilita o uso de classes, objetos, herança.

Tipagem dinâmica

Tipos de variáveis podem mudar e não precisam ser explícitos

ONDE É USADA?

Desenvolvimento web

Linguagem de Backend fácil de testar e com diversos frameworks de API

Ciência de Dados

Adotada por diversos tipos de pesquisadores pela sua simplicidade e capacidade de trabalhar com grande volume de dados

Scripting/Automação

Pela simplicidade, permite a automação de tarefas repetitivas, dentro da própria indústria.

SINTAXE PYTHON

SINTAXE

Variáveis

- Tipos básicos
  - Bolean
  - Int
  - Float
  - String
  - List
  - Dict
  - Tupla
  - Set

```python
a = True # Boolean

b = 1 # Int

c = 1.0 # Float

d = 'a' # String

e = [1, 2, 3] # List

f = {'a': 1} # Dict

g = (1,2) # Tupla

h = set([1, 2, 2]) # Set
```

SINTAXE

Variáveis: Boolean

- True ou False
- Usada como controle

```
a = True # Boolean

type(a)
```

SINTAXE

Variáveis: Int

- Numero inteiro positivo ou negativo
- Nem toda operação de inteiro resulta em inteiro.

```
type(10/2)
```

```
a = 1 # Int
type(a)
```

SINTAXE

Variáveis: Float

- Numero Real positivo ou negativo
- Qualquer operação entre inteiros e Float resulta em floa

```
type(1 + 1.0)
type(1/2)
type(1 * 2.0)
type(2 ** 2.0)
type(2 % 3.0)
```

```
a = 1.1 # Float

type(a)
```

SINTAXE

Variáveis: String

- Sequencias de caracteres
- Limitado pela memória
- Interpoladas
- Iteradas

```
a = 'A' # String
type(a)
```

```
# Interpolação
nome = 'Marcio'

print(f'Olá {nome}')
```

```
# Iteradas
nome = 'Marcio'

for i in a:
  print(i)
```

SINTAXE

Variáveis: String - Funções

- capitalize() | lower() | upper()
- count()
- find()
- split()
- ...

```
# Dir retorna os possíveis métodos
dir(str())
# Help retonar manual sobre o método
help(str().split)
```

SINTAXE

Variáveis: List

- Lista de elementos
- Iteraveis
- Mutaveis

```
a = [1, 2, 3, 4]
for i in a:
  print(a)

print(a[2])

b = ['M', 'a', 'r',
'c', 'i', 'o']
print(b)
```

```
a = [1 ,2] # List
type(a)

b = [1,'2'] # List
type(b)
```

SINTAXE

Variáveis: List - Funções

- append() | insert()
- pop()
- copy()
- sort()
- ...

```
a = [3, 2, 10, 4]

a.append(5)

a.sort()

print(a)
```

SINTAXE

Variáveis: List - Operações

- Soma (list)
- Multiplicação (Int)
- Slice [:]

```
[0] * 10
```

```
a = [1, 2, 3, 4]
a[:2]
```

```
a = [1, 2]
b = [3, 4]

a + b
a.append(b)
```

```
a = [1, 2]
b = a

b[1] = 10
a[1]
```

SINTAXE

Variáveis: Dict

- Sequencia de dados desordenados
- Chave-Valor
- Podem servir como structs

```
a = {'teste1': 10,
'teste2': 12}

a['teste1']
```

```
a = {'teste1': 10,
'teste2': 12}
b = {'teste1': 11,
'teste2': 12}

c = {'a': a, 'b':
b}

c
```

```
a = {'a': "bcd"} #
Dict
type(a)
```

SINTAXE

Variáveis: Dict - Funções

- keys()
- values()

```
a = {'teste1': 10,
'teste2': 12}

a.keys()
a.values()
```

SINTAXE

Variáveis: Dict - Operações

- Busca
- Iteração

```
a = {'teste1': 10, 'teste2': 12}

for i in a:
  print(i)
```

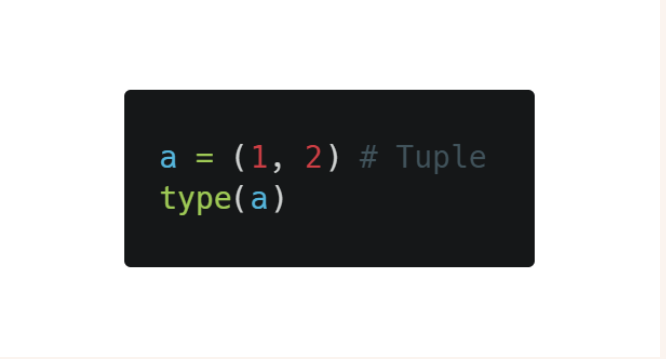
```
a = {'teste1': 10, 'teste2': 12}

print(list(a.keys())
[list(a.values()).index(12)])
```

SINTAXE

Variáveis: Tuple

- Sequencia de dados
- Imutavel



SINTAXE

Variáveis: Set

- Sequencia ordenada de dados

```
a = set([1, 2, 2]) # Set
type(a)
print(a)
```

SINTAXE

Condicionais

- If
- Else
- Elif
- Ternário

```
a = 1
if (a % 2 == 0):
  print('Par')
elif (a > 10):
  print('Maior 10')
else:
  print('Impar menor que 10')
```

```
print(1) if 1 < 10 else print('untrue')
```

```
a = 1
if (a % 2 == 0):
  print('Par')
else:
  print('Impaar')
```

SINTAXE

Loops

- For
- While
- Comprehension

```python
a = [1, 2, 3, '10']

[i if type(i) == int else 0 for i in a]
[i for i in a if type(i) == int]
```

```python
a = 10

while(a > 0):
  print(a)
  a -= 1
```

```python
a = 'Nome'
for i in a:
  print(i)
for i in range(len(a)):
  print(i)
for i,v in enumerate(a)
    print(i)
    print(v)
```

- Podem retornar nenhum ou uma tupla de valores
- Aceitam nenhum ou N parâmetros
  - *args
  - *kwargs

```
def fun(*args):
  for i in args:
    for i in range(i):
      print('*', end='')
    print('')

fun(1,2,3,4,5)
```

```
def fun(**kwargs):
  args = kwargs.get('args')
  for i in args:
    for i in range(i):
      print('*', end='')
    print('')

fun(1,2,3,4,5)
```

```
def recursao(a, b, c):
    if b < 10:
      a.append(c)
      b += 1
      recursao(a, b, c)
    return a
```

SINTAXE

Funções - Built-in

- Documentação
  - dir()
  - help()
- Debug
  - breakpoint()

```
abs(-1)
len([1, 2, 3])
range(10)
enumerate([1,2,3])
```

SINTAXE

IO

- Input
- Print
- Open

```
filename = 'arquivo.txt'

with open(filename) as f:
  # Print the file
  print(f.read())
  # Get the iterator of lines
  lines = f.readlines()
  # Execute operations on those lines
  for line in lines:
    print(line)
```

SINTAXE

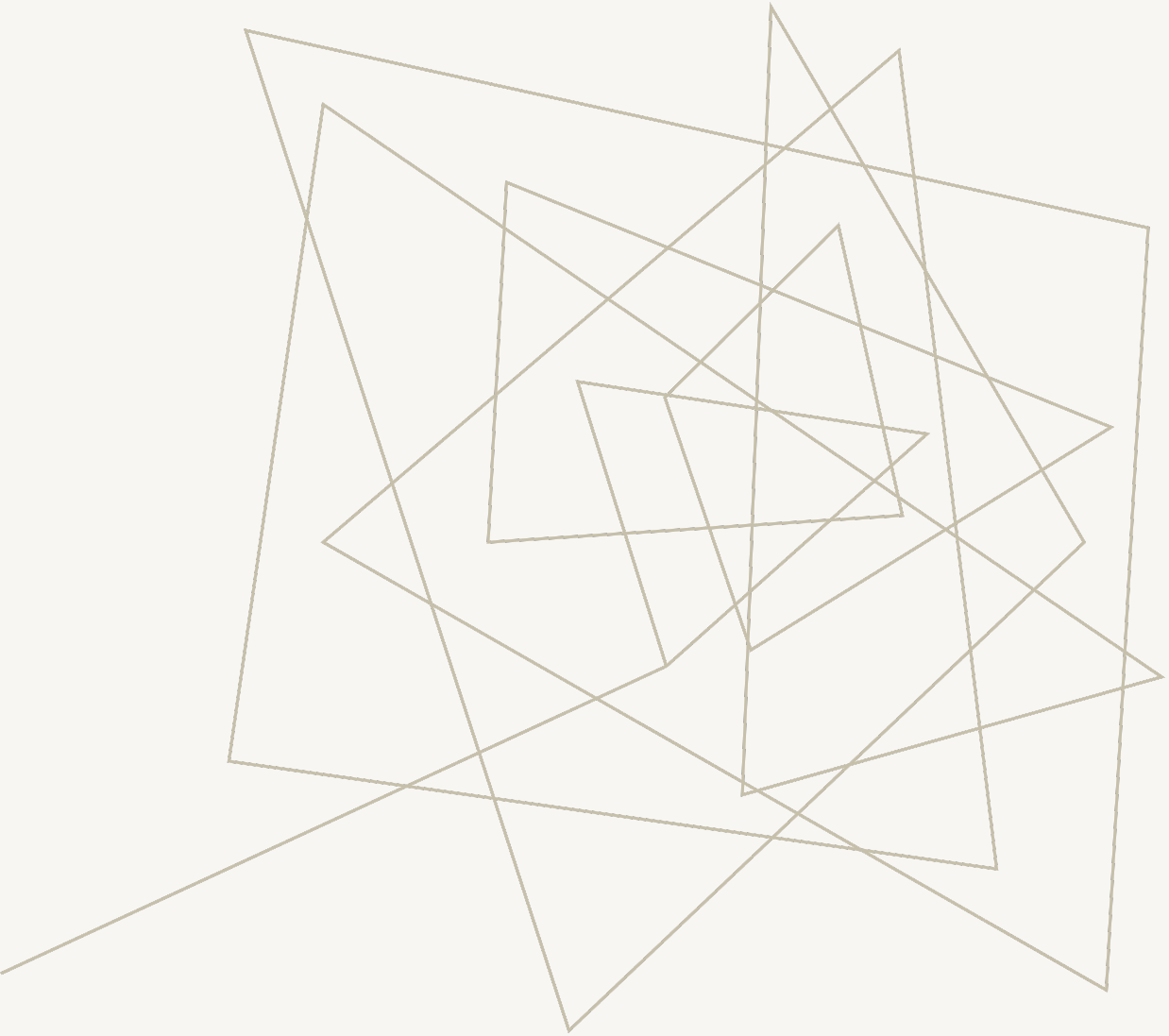
Outras sintaxes importantes

- Funções de código
- Criação de classes
- Import de libs

```
import numpy as np
```

```
class Relogio:
    hora = 1
    minuto = 12
    def __init(self, sec)__:
      self.segundo = sec

    def que_horas_sao(self):
      print(f'{self.hora}:
{self.minuto}')
```



INTRODUÇÃO À ANÁLISE DE DADOS COM PYTHON

INTRODUÇÃO À ANÁLISE DE DADOS COM PYTHON

Numpy

- Array de objetos
- Operações matemáticas
  - Álgebra linear
  - Transformadas
  - Gerações de números aleatórios

```
import numpy as np
```

INTRODUÇÃO À ANÁLISE DE DADOS COM PYTHON

Numpy - Array

- Homogêneo
- Eficiente
- Tamanho fixo
- Otimizado para operações numéricas e matemáticas

```
np.array([1,2,3,4])
```

INTRODUÇÃO À ANÁLISE DE DADOS COM PYTHON

Numpy – Geração de arrays

- Zeros
- Ones
- Arranjos
- Linespace
- Aleatorios

```
zeros_array = np.zeros(5)
ones_array = np.ones(5)
arranjo_array = np.arrange(0, 10, 2)
linear_array = np.linspace(0, 1, 5)
random_array = np.random.random(5)
```

INTRODUÇÃO À ANÁLISE DE DADOS COM PYTHON

Numpy – Operações

- Operações são executadas elemento por elemento
- Possuem o conceito de broadcast

```
a = np.array([1,2,3])
b = np.array([4,5,6])

print(a + b)
print(a - b)
print(a * b)
print(a / b)
print(a ** 2)
np.append(a,b)
np.concatenate((a,b))
```

INTRODUÇÃO À ANÁLISE DE DADOS COM PYTHON

Matplotlib

- Visualização de dados
- Geração de imagens
- Geração de gráficos iterativos

```
import matplotlib.pyplot as plt
```

```
plt.plot(x, y)
plt.scatter(x, y)
plt.show()
```

INTRODUÇÃO À ANÁLISE DE DADOS COM PYTHON

Matplotlib

```
x = np.arrange(0,5,0.1)
y = np.sin(x)
plt.plot(x,y)
plt.show()
```

INTRODUÇÃO À ANÁLISE DE DADOS COM PYTHON

Desafio

- Gerar dados de x aleatórios
- Usar a função para calcular os valores de y
- Fazer o plot dos dados

```
def polinomial(x):
  y = 1.3 x ** 3 + 10 * x ** 2 - 6 * x - 10
```

INTRODUÇÃO À ANÁLISE DE DADOS COM PYTHON

Pandas

- Manipulação de dados
- Análise de dados
- Filtragem
- Plots

```
import pandas as pd
```

INTRODUÇÃO À ANÁLISE DE DADOS COM PYTHON

CSV

```
import pandas as pd

dados = pd.read_csv("dados.csv")

dados
```

```
import csv

with open('file.csv') as arq_csv:
  leitor = csv.reader(arq_csv, delimiter=';')
  for linha in leitor:
    print(','.join(row))
```

INTRODUÇÃO À ANÁLISE DE DADOS COM PYTHON

Filtragem

- Select como em banco de dados
- Grouping
- Cálculos

```
cars = kagglehub.load_dataset(
    KaggleDatasetAdapter.PANDAS,
    "mohammedadham45/cars-data",
    "cars_data.csv",
)
```

```
cars
cars.query('transmission == "Automatic"')
```

INTRODUÇÃO À ANÁLISE DE DADOS COM PYTHON

KaggleHub

- Datasets diversos
- Notebooks
- Desafios pagos

INTRODUÇÃO À ANÁLISE DE DADOS COM PYTHON

KaggleHub

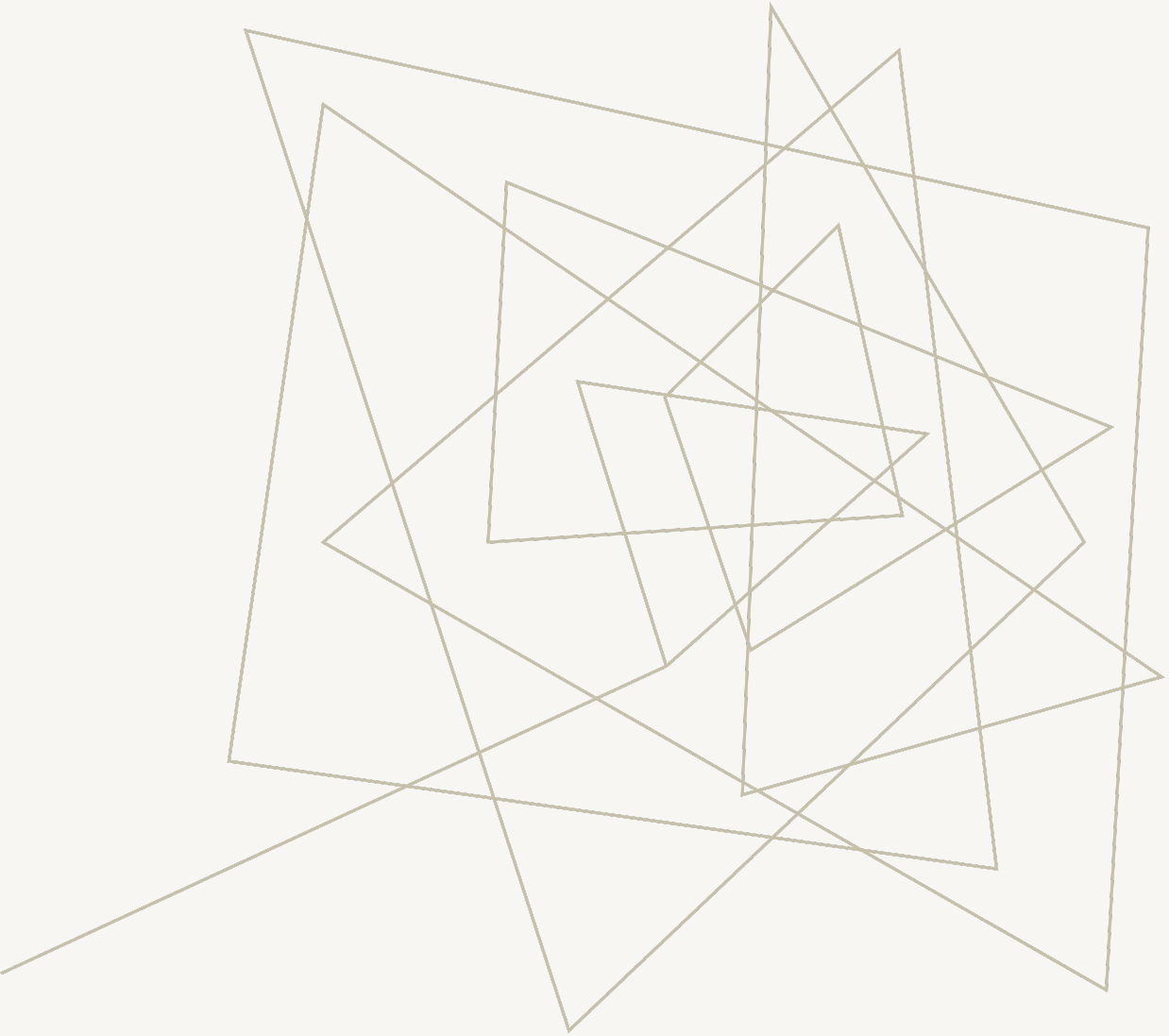
- Datasets diversos
- Notebooks
- Desafios pagos

INTRODUÇÃO À ANÁLISE DE DADOS COM PYTHON

Desafio prático

- Pegar a média de preço de carros manuais antes de 2020.
- Plotar o gráfico dos dados.

```
cars = kagglehub.load_dataset(
    KaggleDatasetAdapter.PANDAS,
    "mohammedadham45/cars-data",
    "cars_data.csv",
)
```



INTRODUÇÃO À MACHINE LEARNING COM PYTHON

INTRODUÇÃO À ANÁLISE DE DADOS COM PYTHON

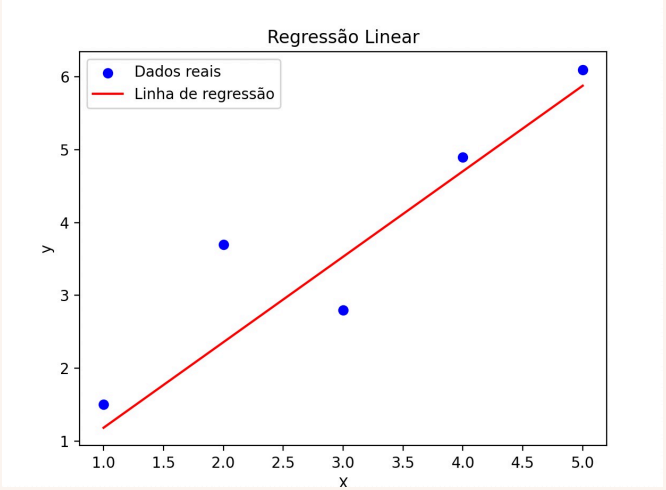
Regressão X Classificação

- Supervisionado
- Regressão
  - A variável de resposta e contínua
  - Pode ser linear ou não linear
- Classificação
  - A variável de resposta é categórica
  - Divide em N Grupos

INTRODUÇÃO À MACHINE LEARNING COM PYTHON

Regressão

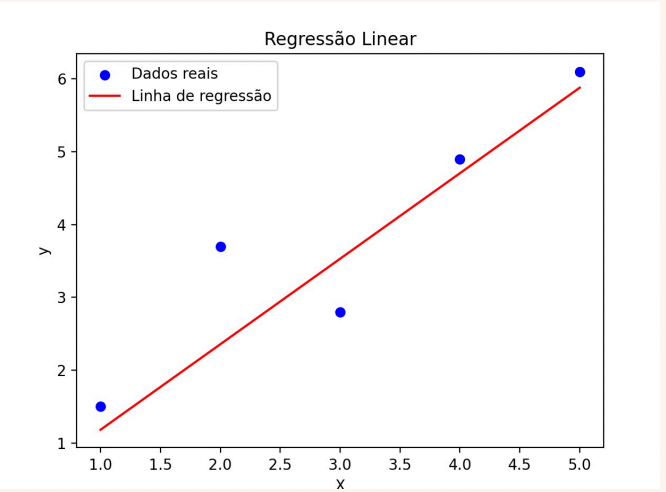
- Prever o comportamento de uma variável baseado em outra
- É um problema de otimização
- Pode ser linear ou não linear



INTRODUÇÃO À MACHINE LEARNING COM PYTHON

Regressão - Linear

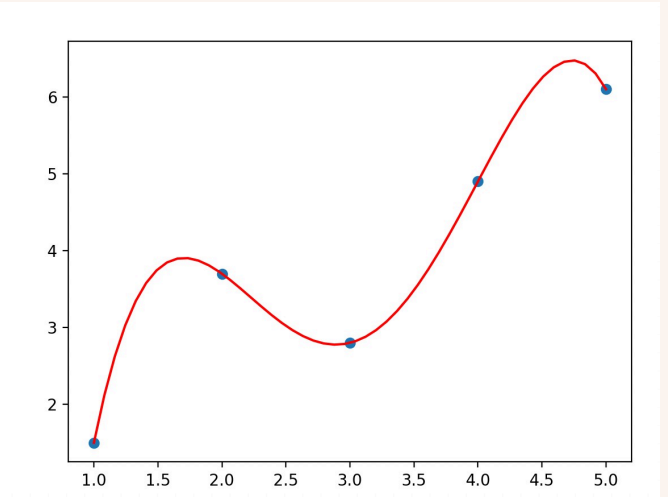
- $f(x) = a\,x + b$



INTRODUÇÃO À MACHINE LEARNING COM PYTHON

Regressão - Não Linear

- Polinomial:
  $f(x) = a_n\, x^n + a_{n-1} * x^{n-1} + \ldots + a_0$
- Exponencial: $f(x) = ae^{-bx} + c$
- Logística: $f(x) = \dfrac{1}{1 + e^{-(ax - b)}}$



INTRODUÇÃO À MACHINE LEARNING COM PYTHON

Regressão - Scipy

- Curve_fit

```
x = np.array([1, 2, 3, 4, 5])
y = np.array([2.3, 4, 5, 1.9, 3])
```

```
from scipy.optmize import
curve_fit
```

```
def function(x, a, b):
  return a*x + b
```

```
params, cov = curve_fit(func, x, y)
```

INTRODUÇÃO À ANÁLISE DE DADOS COM PYTHON

Desafio prático

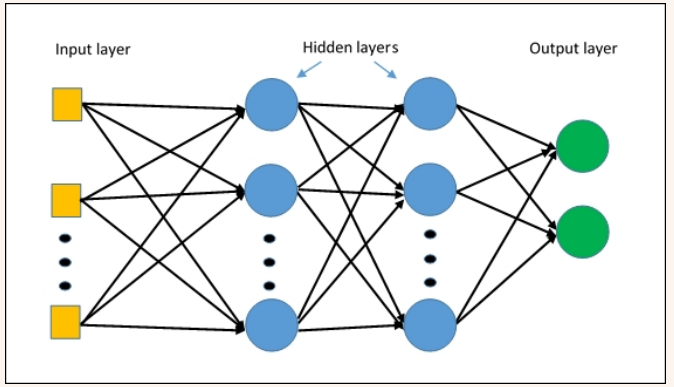
- Pegar a média de preço de carros manuais antes de 2020.
- Plotar o gráfico dos dados.
- Criar uma regressão polinomial.
- Gerar valores de X baseado na função e plotar os gráficos

```
cars = kagglehub.load_dataset(
    KaggleDatasetAdapter.PANDAS,
    "mohammedadham45/cars-data",
    "cars_data.csv",
)
```

INTRODUÇÃO À ANÁLISE DE DADOS COM PYTHON

MLP

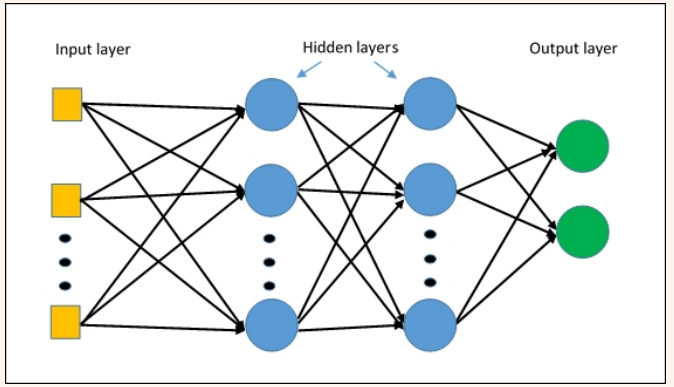
- Entrada
  - Valores de X e Y
- Camadas ocultas
  - Cada neurônio recebe dados das camadas de entrada e passa a diante multiplicando por um peso
  - Aplica uma função de ativação
- Saída
  - Multiplexa as saídas da camada oculta e gera resultados
  - Podem ter N neurônios de saída, desde que respeitem o tipo de saída, e.g. binário um neurônio



INTRODUÇÃO À ANÁLISE DE DADOS COM PYTHON

MLP

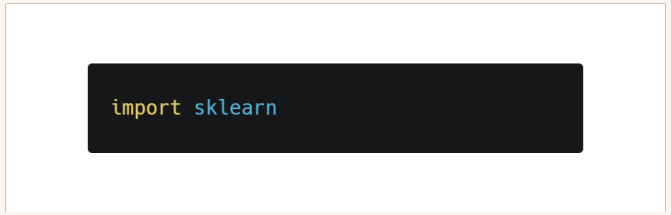
- Treinamento
  - Ajuste dos pesos de a uma função de gradiente
  - Utiliza uma função para determinar a convergência da saída de acordo com o resultado esperado
  - Feedforward
    - Propagação dos dados
  - Backpropagation
    - Propagação dos pesos



INTRODUÇÃO À ANÁLISE DE DADOS COM PYTHON

MLP – Scikit-learn

- Classificação
- Regressão
- Clustering
- Pré-processamento



INTRODUÇÃO À ANÁLISE DE DADOS COM PYTHON

MLP - Scikit-learn

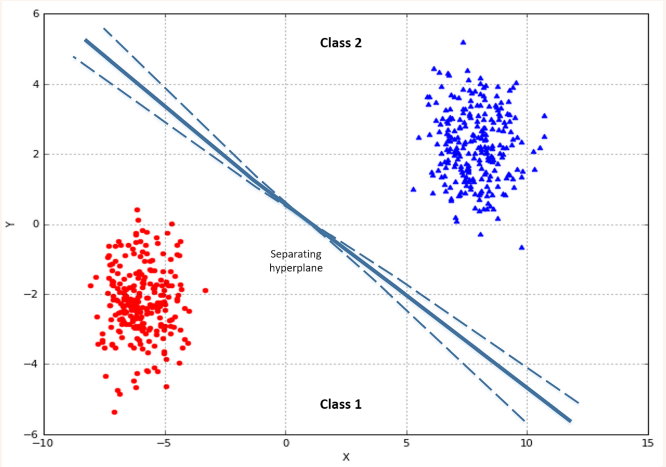
```
mlp = MLPRegressor(
    hidden_layer_sizes=(15,),
    activation='relu',
    solver='adam',
    learning_rate='adaptive',
    n_iter_no_change=30,
    max_iter=100000,
    learning_rate_init=0.01,
    alpha=0.01,
    verbose=True,
    tol=1e-4
)
mlp.fit(X_training, y_training) # training
y_pred=mlp.predict(X_testing) # prediction
```

```
from sklearn.neural_network import MLPRegressor
```

INTRODUÇÃO À ANÁLISE DE DADOS COM PYTHON

MLP – Classificação

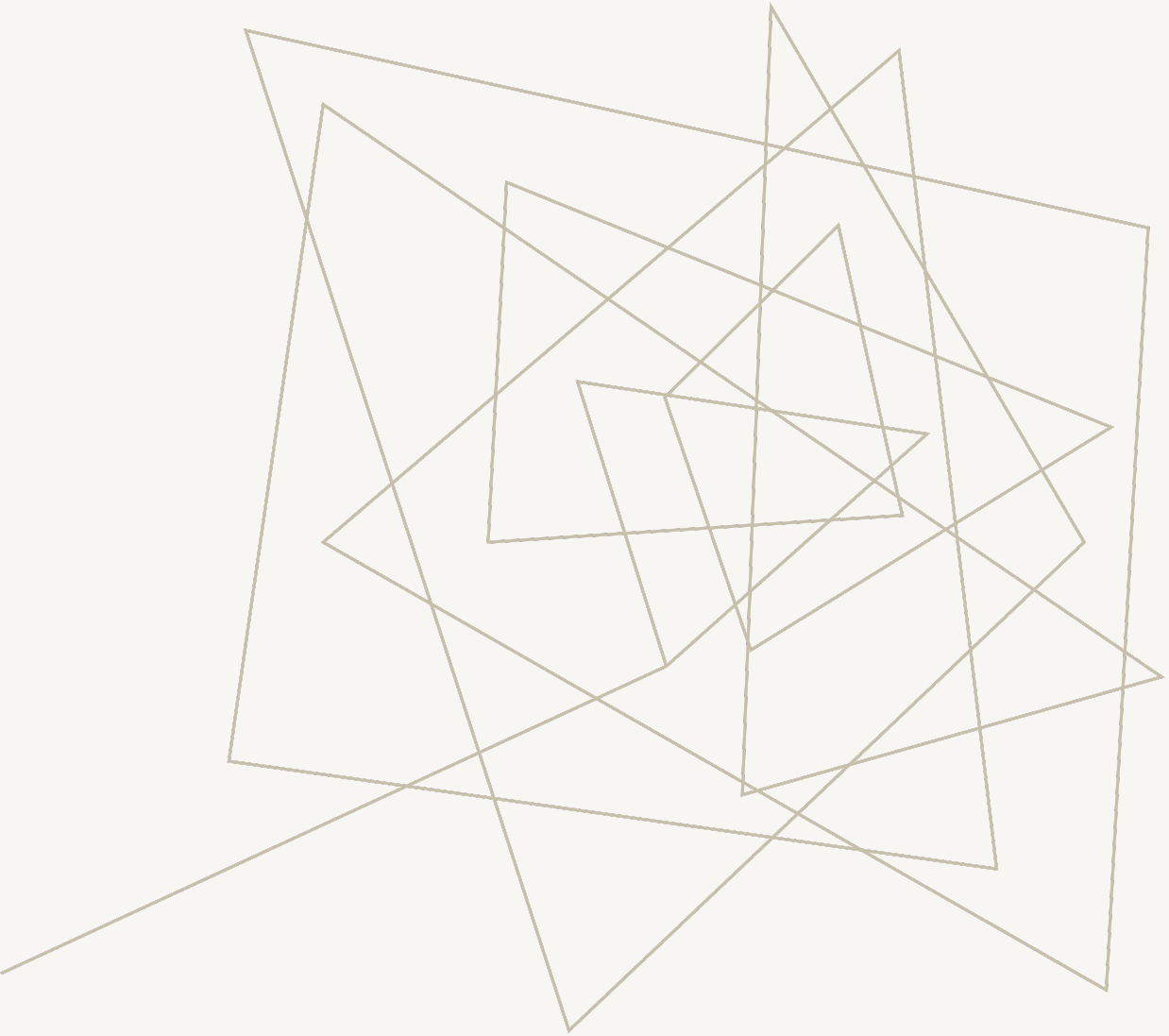
- Separar grupos
- Pode se usar um classificador linear ou não linear



INTRODUÇÃO À ANÁLISE DE DADOS COM PYTHON

MLP - Classificação

```
mlp = MLPClassifier(
    hidden_layer_sizes=(2,),
    max_iter=5000,
    learning_rate='adaptive',
    solver='adam',
    verbose=True,
    n_iter_no_change=100,
    tol=1e-4,
    learning_rate_init=.1
)
mlp.fit(X_training, y_training) # training
y_pred=mlp.predict(X_testing) # prediction
```



DESAFIO PRÁTICO

DESAFIO PRÁTICO

- Pegar a média de preço de carros manuais antes de 2020.
- Plotar o gráfico dos dados.
- Criar uma regressão polinomial.
- Gerar valores de X baseado na função e plotar os gráficos.
- Treinar uma rede MLP para regressão.
- Comparar os resultados das previsões e dos gráficos.

OBRIGADO

Github: @marciorgb

Linkedin: in/marcio-gomes-barbosa