Aula 1

Linguagem de Programação

Prof. Wellington Rodrigo Monteiro

Conversa Inicial

Bibliotecas em Python

Ir além do básico

Velocidade de desenvolvimento

Biblioteca padrão x bibliotecas externas

Leitura da documentação

Expandindo o Nosso Repertório com as Bibliotecas

Estrutura de uma biblioteca

Biblioteca (library)

Requisitos (requirements)

Módulo (module)   
Função (function)   
Tipos de dados (data types)

Bibliotecas em Python

Biblioteca padrão em Python

Bibliotecas externas

Github/GitLab

PyPI

Conda

Documentação

Python Standard Library

Tipos de dados

Números inteiros (int)

Números decimais (float)

Sequência de caracteres (str)

Listas (list)

Dicionários (dict)

Funções

Leitura (input)

Escrita (print)

Tamanho (len)

Valor absoluto (abs)

Sequência numérica (range)

Ajuda (help)

Módulos

Data e hora (datetime)

Expressões regulares (re)

Matemática (math)

Números aleatórios (random)

Armazenamento de informações (pickle)

Bibliotecas externas

Feitos por outros

“Awesome lists”

PyPI

Feitos por você

Usando módulos das bibliotecas

import [módulo]

from[módulo]

import [função(ões)]

Criando Gráficos com Python

Principais bibliotecas

Matplotlib

Seaborn

Plotly

Bokeh

Comparativo

Matplotlib

Mais antigo

Mais versátil

Mais difícil de dominar

Seaborn

Mais recente

Gráficos mais agradáveis

Mais fácil de dominar

Usa o matplotlib como base

Plotly

Integração com soluções de Business

Intelligence(BI)

Gráficos animados

Tipos de gráfico comuns

Uma variável sendo analisada

Ordenada (ex.: pelo tempo)

Gráfico de área (area)

Gráfico de linha (line)

Gráfico de caixa (boxplot)

Não ordenada

Histograma (histogram)

Gráfico de densidade (density)

Gráfico de caixa (boxplot)

Tipos de gráfico comuns

Mais de uma variável sendo analisada

Não são parecidas

Gráfico de dispersão (scatter)

São parecidas Mapa de calor (heatmap)

Gráfico de pizza (pie chart)

Resolvendo o Problema das Listas

Listas e NumPy Arrays em Python

Listas

Múltiplos valores em uma variável

Contêm vários elementos de diferentes tipos

Standard Python Library

Arrays

Múltiplos valores em uma variável

Contêm vários elementos de somente um tipo NumPy

NumPy Arrays

São mais rápidas do que as listas

Usam menos espaço em memória do que as listas

Permitem a aplicação de operações adicionais

Transposição

Concatenação e separação de arrays

Ordenação

Operações algébricas (determinante, autovalores)

Operações lógicas (where, greater/less than, isinf, isnan)

Operações estatísticas (quantile, median, std)

Trabalhando com Tabelas em Python

Pandas

Biblioteca construída sobre o NumPy

Tipos de dados

DataFrame

Series

Exemplos de fonte de dados

CSV/JSON Excel SQL

DataFrame

Resample

Agrupar dados por dia/semana/mês/ano

groupby/merge/melt/pivot\_table

head/tail loc/iloc

Extração de dados

get\_dummies

Series.dt.week/day/month/weekday

Correlações

Pearson (relacionamento linear)

Spearman (relacionamento monotônico, não necessariamente linear)

Kendall (relacionamento monotônico, mais robusto que Spearman, mais lento)

Resolvendo Problemas com uma Linha: Uso de Expressões Regulares em Python

Expressão regular

“Receita” de um padrão de busca

Conjunto de caracteres para encontrar um resultado (dar match)

Velocidade x legibilidade

[0-9] [0-9a-z] [0-9a-zA-Z] [^0-9a-z] [0-9]+[0-9]\*[0-9]?

Validação de CEP[\d]{5}-[\d]{3}

agora falando sobre bibliotecas em específico é muito legal a gente conversar um pouco sobre isso não somente essas bibliotecas mas também sobre outros assuntos em forma de analogias então é muito provável que a gente vai entregar alguns exemplos de analogias Por que realmente algo muito útil para fixar o nosso conhecimento beleza é um jeito Isso me acompanha que nos slides por favor primeiramente falando sobre uma estrutura e de uma biblioteca tá Então imagina se a biblioteca realmente como uma biblioteca física Então imagina que na Biblioteca física ela é um conjunto né o espaço físico eu posso entender que o que é uma estante um conjunto de estudantes nessa biblioteca pode ser até um prédio e dentro da sobre a Teca como você deve imaginar eu posso ter vários livros mas também não vou ter somente livros e eu posso ter bibliotecas que não tem outros recursos audiovisuais com as ideias que tem filmes posters biblioteca se tem artigos até para servir até que tem obras de arte Então você vai dizer que essa a definição de uma biblioteca Eu acredito que na cabeça não sabe mexer né sua e de mais das demais pessoas pode ter uma ideia bem parecida mas cada uma e com suas variações e também passei bibliotecas em diferentes né bibliotecas para diferentes fins uma biblioteca por exemplo que foi feita para pessoas de t.i. Vai ter muito provavelmente livros que são mais na área de exatas nem após ter livros que são de programação livros que são sobre bancos de dados eu posso ser livres que são sobre cálculos e assim por diante da mesma forma que detecta Talvez para um pessoal da culinária não vai ter quase nada de cálculo não vai ter pessoalmente nada de banco de dados e 50 livros específicos sobre Área deles e da mesma forma para os demais cursos tem um jeito isso que que é uma biblioteca dentro do Python Imagino você na hora que você pode construir um pedaço de códigos para construir um algoritmo em Python para resolver um determinado problema então seja lá um algoritmo que ele consegue mostrar um gráfico 3D animado e você gastou muito tempo e conseguiu desenvolver esse algoritmo não existe nada no mundo sobre isso você foi a primeira pessoa que desenvolveu aquilo e você entende que aquilo também pode ser utilizado em outros casos de uso pode ser até utilizado por outras pessoas Então se cria né aquele seu algoritmo você prepara ele com até outros tipos de gráficos que também você criou e queria que fosse um pacotinho fechado daquilo Esse pacote fechado como como eu falei aí no final das contas Pode sim virar uma biblioteca então você pode ver aqui no slide que eu tenho uma estrutura que ela com sua se realmente do maior nível para o menor nível Então eu tenho lá a biblioteca que é o maior nível de todos de uma angina assim que eu passei uma biblioteca que gera qualquer tipo de gráfico do mundo daí dentro dessa biblioteca que gera qualquer tipo de gráfico do mundo eu também posso ser rei requisitos uma coisa que é muito comum em paz é uma biblioteca utilizar outras bibliotecas Então existe realmente essa união entre diferentes desenvolvedor de Python de um começar realmente ao usar as bibliotecas as soluções que os outros desenvolveram Então até uma das vantagens da Gente Feliz aparição dos portugueses no cartão ele ser muito difundido hoje em dia é realmente por ter basicamente uma solução pronta para fazer tudo no dia que o dente solução pronta né mas realmente um algoritmo uma base que consegue resolver algum problema então posso ter biblioteca que vai gerar gráficos para Loteca que é de machine learning eu posso ser uma biblioteca que ela vai conseguir trabalhar com música Posso ter uma biblioteca aqui trabalho às vezes com OBS pode ser uma dor que trabalha com vídeo aí então eu tenho até a biblioteca para criar jogo Então realmente temos diferentes bibliotecas que resolve em diferentes casos e como farei uma pode até utilizar outra para ter uma de dependência da outra essas independências a gente acaba gostando aí dentro de um desses requisitos então é muito provável que quando você começar a ir trabalhar com biblioteca Se começar analisar os códigos e temos será que uma da Biblioteca de gráfico para utilizar outra de ataque de graxa um requerimento da como requisito então isso realmente é algo bem comum e que além disso essas bibliotecas elas podem ser subdivididos em módulos que eu quero parar um pouquinho e voltar naquelas analogias bibliotecas que nós estamos conversando inicialmente nesses exemplos de bibliotecas Imagine na biblioteca agora como ser um prédio não tem um prédio né Cheio de estantes cheio de subdivisões dentro dessas subdivisões dentro dessas estantes eu vou ter diferentes livros como se deve imaginar esse prédio ele tem que ter um mínimo de organização não conseguir encontrar o livro que nós precisamos é importante ter essa

Veja por algum agrupamento de grupo de interesse tem que ter essa organização aqui em paz então nós temos esses modos então na mesma forma que na naquela biblioteca física na no prédio eu posso falar assim olha o primeiro andar é somente só tem livros de terror para ser 50 estantes mas essas estantes são somente de livros de culinária no segundo andar que é livro de arquitetura no terceiro andar Então essa organização a que eu tenho e módulos da mesma forma que tem três andares nós comentamos aqui neste exemplo então eu posso ter imagina uma biblioteca de gráficos eu posso ter um módulo queria somente de gráficos 3D depois eu posso ter outro módulo que é somente gráficos 2D posso ter às vezes outro modo e vai somente fazer alguns cálculos algumas coisas assim para conseguir uma ser aqueles gráficos então esses modos Imagine como sendo essas subdivisões de uma biblioteca e dentro dos Matos nós temos funções e também podemos ser tipos de dados eu queria é só chamar um pouco a sua atenção com esse nome função se você já acabou na estudando criou outros algoritmos em Python você já deve ter criado funções Lembrando que funções Elas começam com aquele DF DF Depois tem o nome de uma função e se coloca o seu código dentro dela então quando nós estamos falando que de funções são exatamente Essas funções que você está imaginando até mesmo que você já desenvolveu Então são funções que outras pessoas criaram para resolver um determinado caso eu passei aqui tem uma função que vai mostrar um gráfico de bolinha Estou desanimado eu posso ter outra função que vai criar um gráfico de barras todo animado o outro que possa criar um gráfico de pizza então aqui vai ter diferentes funções cada uma delas é especialista né em um determinado caso de uso também a gente pode ter finalmente na bibliotecas que possuem tipos de dados a mais além daqueles que nós já trabalhamos em Python até o momento agora pensando especificamente sobre as bibliotecas em Python importante a gente vê que além temos claro né uma lotérica padrão em Python e temos várias outras que eu já dei aqui um pequeno spoiler né comentei que temos por exemplo e lotéricas que trabalham com jogos grátis que trabalham com gráficos bibliotecas que trabalham com na área de Inteligência Artificial e ciência de dados mas eu queria primeiramente uma sua atenção para que nós chamamos na biblioteca padrão em Python Então me acompanha novamente no slide essa biblioteca padrão em Python que a primeira que está listada aqui essa biblioteca padrão ela já vem entre aspas na embutida ou inclusa no Python e possui várias funções que você já deve ter trabalhado antes exemplos dois exemplos básicos e ainda lembra né aquelas funções que você digitar o seu nome mostrava depois do seu nome no final uma célula vegetal de uma soma aí são exemplos bem básicos de funções que vem dentro dessa placa padrão em Python que você não precisa instalar agora queria chamar sua atenção para esse s o item quando nós falamos aqui em jupi lotéricas externas são marcas que elas não vem entre aspas né Por padrão elas não vem já incluso no Python sem legenda só comprei um computador em uma loja Recebeu o computador na minha casa eu fui lá no site do Python é baixei e instalei não significa necessariamente que todos os botecos do mundo estão já dentro do meu computador que eu já posso usar eu não posso sujar a começar a criar gráficos em Python criar um jogo em Python é importante e na Internet baixar o que eu preciso que eu confio em ti para resolver nosso problema e com isso aí sim a gente consegue trabalhar com aquelas bibliotecas externas na então existem diferentes sites alguns recursos diferentes para isso um deles por exemplo é um lixo então descreve ou descreve na São sites de versionamento de código até uma pequena recomendação se você não criou sua conta de algum site assim né seja a Cris por quê tirante uma boa criação de um portfólio pensando em crescimento profissional importante de outras pessoas saibam como você desenvolve seus códigos como que as organização Que tipo de trabalho que você faz então até legal você criar um repositório no github criar repositório né todos os trabalhos que você por ventura cria deixa lá porque vale a pena Como falei para sua própria trajetória profissional mas independentemente disso dentro de site também existem bibliotecas que são de código aberto e que você consegue analisar enfim entender quais funcionalidades existem e com isso instalar também o seu próprio computador outra opção que também existe aqui temos o pai então o pai ele é um site aqui também é basicamente um grande repositório em Python que você consegue pegar aquele diferentes lotéricas instalar diretamente no seu computador por uma linha de comando então por exemplo foi lá no o que gera gráficos né e a super legal eu quero instalado no meu computador como que eu faço o pai ele é um site que permite entender quais versões existem daquelas lotéricas para instalar no meu computador e conseguiu utilizar livremente Então até o pai pai ele é conectado ou entre aspas ele conversa com Pipi e a gente também será em breve ele serve para instalarmos esses diferentes pacotes nessas diferentes bibliotecas em Python no nosso computador a terceira opção é o Kong ou quando há também é um gerenciador de pacotes aí ele é ele é conversa com Anaconda estranhos da gente pensa Python anaconda condá net realmente toda a temática de cobras por conta do Python e o Congo ele também é outro gerenciador de biblioteca então da mesma forma que eu posso instalar uma peteca via via pai pai eu também posso instalar uma determinada biblioteca via conta a definição se é um sem o outro Depende muito do contexto que nós estamos trabalhando tem presos por exemplo que trabalham com Pipi e com contas de empresas que trabalham somente um pipi é o pai de empresas que preferem sol quando ainda tem outras que preferem o samba na igreja de novo essa temática haja cobras então existem realmente diferentes variações e isso é importante saber que existe se você estiver trabalhando no seu próprio computador no seu próprio contexto eu direi que é um bom ponto de partida é trabalhar com pai como fazer para instalar né suas técnicas e realmente ter em uma boa gestão ponto de novo da documentação vale a pena a gente prestar atenção aqui porque no dia que recebe quando a gente entra em qualquer site em qualquer repositório não é qualquer biblioteca ele quase sempre vai ter um link Zinho para documentação e anexar documentação que a gente encontra exemplos encontramos até um guia de como instalar a Então se deve ficar pensando aqui beleza né estamos falando de paz de pipi Mas como que nós instalamos isso no computador como que eu posso instalar uma certa gráfica por exemplo nesses nessas próprias documentações existem links específicos para instalação Isso depende muito de cada caso tem por exemplo na peteca que ela pode requerer várias configurações que outras é tão somente uma linha de código então por isso que não é tão somente falar sempre Digite o comando tal e vai dar certo isso depende muito de cada dia Teca então por isso que volta no ponto de saber encontrar essas documentações né saber como se orientar e por isso que de novo trazemos este ponto aqui direto isso me acompanha aqui nos Estados mais uma vez olhando aqui nos slides pode ver que já temos alguns tipos de dados que vem por padrão no país Então esse aqui você já deve ter visto no passado né posso trabalhar com números inteiros em Python que surgiu o índio tem o neoflox para números decimais você também já deve ter trabalhado bastante com string ou STR que é uma cadeia de caracteres seu próprio nome e o endereço da cadeia de texto entra nessa stream que ela também temos discos de dados E também temos dicionários gerados dito isso é importante saber que temos outras bibliotecas que incluem mais tipos de dados do que somente estes beleza também olhando naquela biblioteca padrão né do Python você também já deve ter contato com pelo menos algumas dessas funções no passado então só Relembrando O que são funções possuem uma função para leitura de dados que seria importante ter funções para escrita né ou seja mostrar alguma coisa na tela que será print posso ter também funções aí para ler é o tamanho de uma lista de uma cadeia de caracteres que seria além é a outra vem para arredondar números tem que ser áudio também para obter o valor absoluto ou seja sem aquele aquele sinal de negativo se tiver em algum número após também criar sequências numéricas e a última que acredito que ela é bem legal e ajuda muito no dia a dia é a help então eu quero saber como que funciona a melhor print Às vezes você trabalhou é tanto com a print era só print e o nome não é o print e um valor e não sabe que outra as funcionalidades existentes na Twitch Então vale a pena por curiosidade depois digita lá help entre parentes print e ver o que que vai acontecer vai sair uma página do manual do próprio pai tô explicando que outras funcionalidades de ensino frente a isso como fazer para tirar uma vantagem muito legal no dia a dia também sobre módulos né lembra que eu falei antes aquela aquela analogia da biblioteca né da do prédio eu acho que é importante dessa analogia para ficar um pouco mais claro aqui que outros módulos existem tem uma gênia que até o momento se for preso é só uma pequena exceção daquele prédio né daquela biblioteca mas existem outros módulos dentro da sua própria biblioteca padrão do Python que podem também te ajudar por exemplo tem um módulo que é somente para data e hora mas também eu tenho animados que são somente para operações matemáticas então né quando estamos já pensando em seno cosseno e elevar o número ao quadrado ao cubo a quarta potência quem tem funções para isso eu passei também funções para gerar números aleatórios que é muito legal a no dia a dia também para aplicações de criptografia e afins a também tem um módulo sobre expressões regulares esse item de expressões regulares algo muito legal que a gente também será em breve por quê O que é no manual né um pequeno filtro para rapidamente encontrar informações em textos e também por último temos um outro módulo que é para armazenar informações então não estou falando aqui de pegar por exemplo um texto e salvar um arquivo não é pegar uma variável pegar uma lista tal como ela está e salvar realmente um arquivo no Python para carregar depois então isso também possível em Python com a biblioteca picou e Finalmente né ligando ao tópico aqui em Vila theca cisternas existem como falei várias outras bibliotecas que podem te ajudar Essas bonecas são feitas produzidos na para outros desenvolvedores você também pode criar sua própria biblioteca Então existe essa possibilidade de você também eu jogar comunidade de desenvolvedores criando mais bibliotecas e também existem realistas curadas por outros pesquisadores por outros desenvolvedores sobre quais outras bibliotecas seriam interessantes que você tivesse o contato como eu falei se tem interesse em a programação e independente tem já um conjunto de técnicas que são utilizadas no mercado de trabalho para isso e ficam dentro dessas aos analistas pegar esse tem jogar no no Google se chamar encontrar listas que são não somente do pai mas de outras linguagens de programação que possam ser úteis para você além disso a gente também pode utilizar esses módulos dentro do nosso código de duas formas A primeira é importando o modo por completa e a por isso que existe e se importe módulo como se pode ver a no slide nesse penúltimo e tem e o último item ele se refere a tão somente quando se quer utilizar algumas funções de um módulo imaginação analogia como sendo você indo a biotécnica se encontra o módulo tem uma gênia que que o modo é uma estante cheia de livros tem casos que você realmente preciso de todos os livros aquela estante vai ser um peso para descarregar mas você precisa daquilo é que ela penúltima E antepenúltima caso ele vai importar todo módulo agora se você quer somente um outro livro niketche O Peso seja tão carregado para você você pode somente escolher quais funções como a lei Nacional logia na escolher Quais livros você precisa para o seu trabalho então esse peso que eu estou falando imagina quem Python como sendo o uso de memória mesmo eu quero reduzir uso de memória no meu mel algoritmo quero que ele seja mais leve Às vezes você não precisa importar todo modo se pode tão somente utilizar algumas funções para desempenhar o seu trabalho então é para isso que serve serve na estes exemplos Beleza então eu digo esses pontos até para ficar um pouco mais claro para você que outras opções existem do ponto de vista de uso de bibliotecas até para você realmente saber que além daquela biblioteca padrão e Python existem sim outras opções que você pode utilizar para desenvolver um ótimo trabalho falando sobre bibliotecas que que é um Tec em geral vamos pensar em alguns

tempos em específico E para isso a gente começa a falar um pouco sobre gráficos

é porque nós já falamos bastante sobre isso então me acompanha aqui nos slides por gentileza sobre a criação de gráficos em Python existem algumas técnicas que podem te ajudar Essas aqui que eu solicitando não são as únicas elas são as principais então tanto se você é um pesquisador no caso pensa lá no futuro trabalhar em um mestrado doutorado ou realmente pensa em trabalhar profissionalmente com python você invariavelmente fica na com essas bibliotecas Então temos a mais forte livre temos a cyborg e a boca Então temos essas quatro novamente não significa que são as únicas Existem várias outras mas as mais conhecidas são essas e o gerente no mercado de trabalho as mais utilizadas são a cyborg e a próxima o cyborg Ele é bem simples de ser utilizado e também gera gráficos bem bonitos e o próximo também ele consegue gerar em alguns gráficos animados que são bem interessantes para você então se você também tem curiosidade sobre essas bibliotecas vale a pena né pesquisar um pouco mais sobre elas pensando assim 11 rápido comparativo entre essas biblioteca matplotlib ela é a mais antiga Então ela realmente foi criada aí é um bom tempo por outro lado ela mais versátil quando eu falo versátil aqui imagina que as pode fazer praticamente que você quiser com gráficos que adicionar título remover título a adicionar linhas a mais no seu gráfico divisórias mudar um pouco as cores mudar o tamanho da fonte tudo isso é possível mas como Tem como tudo isso é possível também mais difícil denominar imagine-se como sendo aqueles painéis de controle de aviões bem grande Sabe tem um monte de Botão um monte de opções naturalmente sem fio que se pode fazer várias coisas mas para dominar isso se realmente acaba demorando um pouquinho mais para trabalhar com essa biblioteca em cima da matplotlib foi montada a cyborg Então até lembra que eu falei sobre dependência então quando você está o cyborg já também estava no seu computador ela mais se sente ela gera gráficos que são mais agradáveis aos olhos então fala assim agradável é combinação de cores combinação de fontes é realmente gera sem gráficos que são mais bonitos de serem apresentados deve colocar na apresentação a nível gerencial e por isso ela mais fácil de dominar a Então como ela montada em cima do math plot Lib em como ela mais fácil de você dominar ela tem menos controles Então se acaba tendo aí um pouco um pouco mais de possibilidades rápidos e de criação de gráficos o Block nele ele é bem interessante porque ele possui aí algumas opções de integração com soluções de bi aí quando nós falamos biai imagina tem lá um monte de gráfico para o pessoal manitora ver em tempo real que está acontecendo a com diferentes gráficos em uma única tela Então pronto ele acaba gerando a graça os que são mais fáceis de serem incluídos nesse tipo de visualização e a tela tem outras soluções que são compatíveis com ele e também até uma das vantagens de gesso ou da vantagem de utilizar própolis é que ele também possui gráficos animados quando fala que graça Os animados não é somente um gráfico 3D que ficam girando imagine também um gráfico de linha com para exemplo a alta ou abaixa do dólar ao longo do tempo e aí você vê aquele gráfico passando com tempo de uma forma animada né pontinho pontinho sendo gerado para você então não somente pela imagem fixa então é para isso que ele também pode ser empregado também vale a pena a gente falar invariavelmente da Biotec que você está utilizando e gráficos que utilizar então Independência alce born.in se nós estamos falando aqui né do Polly você pode todos eles têm de alguma forma uma implementação de um gráfico específico Então posso acabar utilizando a um gráfico que ele é de barras e todas as teclas possuem compatibilidade com esse gráfico de barras quais são os tipos de gráficos mais comuns me acompanha aqui no slide por favor quando nós estamos falando somente de uma variável a ser analisada então imagine por exemplo Ah eu quero ver Como comentei anteriormente o comportamento do dólar né então dólar ele ele vai aumentando vai diminuindo com o tempo né esse esse próprio valor como ele é ordenado não seja o que aconteceu ontem precisa necessariamente ver depois do que aconteceu anteontem o que aconteceu hoje também tem que ver Depois desses dois dias aqui eu tenho algumas opções de gráficos eu posso utilizar o gráfico de área de linha ou gráfico de caixa esse gráfico de caixa ele acaba agrupando todos eles e dando uma boa visualização de que tipo de valor ele é muito fora do comum seja muito abaixo da Média muito acima da média então isso dá uma boa visualização

cada tipo de gráfico de nós utilizamos aqui serve para contar uma história então às vezes eu quero pensar somente na distribuição dos valores o box-plot ele pode servir mas ele não vai mostrar nada para nós ao longo do tempo eu quero mostrar algo ao longo do tempo o gráfico de linha ele pode ser útil e acompanha novamente quando nós estamos falando de dados que não são ordenados ou seja não importa a ordem deles eu posso trabalhar com histograma é o que seria um exemplo de dado não ordenado quantidade de pessoas que Possivelmente gostam de determinado sabor de sorvete às vezes essa distribuição ou a quantidade de votos que eu tive a quantidade de notas que eu tive para sorvete no ar de nado ao longo do tempo cada pessoa deu lá uma nota para um sorvete então quero ver como que ficou o agrupamento dessas notas Eu tenho um histograma Eu tenho um gráfico de densidade geográfica de caixa tem isso como eu falei é para mostrar a distribuição dessas notas e desse valores de uma forma que não leva em consideração o tempo e finalmente o que eu queria mostrar para você quando nós estamos falando de mais de uma variável então quando são mais quando são duas ou mais variáveis aqui temos o ponto que é do seu não parecidos o que que seria isso quando quantos lados eles são entre aspas parecidos É como se eu conseguisse colocá-los em uma única métrica então eu passei por exemplo a altura e o peso de diferentes pessoas às vezes altura e o peso um serial comparável entre si porque um deles não se originam quilos e outro tá mentindo em metros tem um dependendo do gráfico vai aparecer uma grande disparidade imagine comigo você só falando de altura altura pode ser basicamente não é de 1m até 2m 2m ou até acima de 2m mas sempre na escala de 1 até 2 2.22.5 já o peso e não bater alguém 12 exemplo 2 m e 2 kg em 2 m e 100 kg 2 m e 120 kg então a escala numérica totalmente diferente nesse cenário temos na outras formas da gente também mostrar esses valores na eu posso ser também o mapa de calor pode ser o gráfico de pizza até por exemplo agrupar valores que são Nery categóricos quando fala uma variável categórica Pode ser às vezes o sexo pode ser colocado que a pessoa trabalha pode ser até um tipo de sorvete Favorito Então posso combinar as diferentes categorias com números ou seja números com números Então eu tenho gráficos que também podem auxiliar nisso aí daí aqui olhando né para todos esses tipos de gráficos é basicamente mostrar para você independentemente da Biotec que você escolher quais gráficos você poderia utilizar para contar uma história Imagine que você enquanto o desenvolvedor você não se envolve exatamente algoritmos você desenvolve soluções Então se precisa garantir essas soluções que resolvam algo mostrar esses diferentes gráficos escolheu o gráfico correto é também parte do nosso trabalho enquanto desenvolvedores beleza e agora vamos falar um pouco sobre listas é muito provável que você já trabalhou com listras anteriormente dito isso me acompanha nos slides que eu acredito que esse ficará mais claro para você então lembrando listas elas são basicamente variáveis que dentro na hora que ele abre colchetes fecha colchetes e vá ser diferentes valores e dentro dela então passei uma lista que tem diferentes notas para ser uma lista que tem diferentes nomes eu passei na lista que vai ter aí diferentes datas e quentão listas aí eu passei diferentes valores em diferentes quantidades também posso ter uma lista com 10 elementos têm 50 alimentos que tem 100 elementos e tudo isso realmente é uma lista também essas listas elas podem ter aí diferentes elementos com diferentes tipos Então dentro de uma única lista eu posso ter aí quem sabe texto misturado com número misturado com data e tudo isso como eu falei faz parte de uma lista porém essas essas que você trabalhou até o momento elas fazem parte dessa biblioteca padrão do Python então basicamente nós ficamos até agora trabalhando somente com listras Porém quando nós trabalhamos com valor muito alto e de valores ou temos que fazer vários carros ao mesmo tempo isso digo de experiência né pensando em trabalho em manipular vários gigantes de informação as listas elas acabam não comportando muito bem essas informações Então a gente tem problemas de performance tem problema se pede código mesmo de ter que implementar muitas linhas de código para o que poderia ser um cálculo mais simples para resolver isso existe um pai um pai uma biblioteca também Python não lembra que o que nós falamos até agora sobre bibliotecas uma pai é uma biblioteca e ela também incluir um novo tipo de dado que a Week Away Imagine ele como sendo realmente uma lista então a gente também consegue colocar múltiplos valores é dentro dela porém somente de um Tiggo então pode ser um rei só de números 1 Reis Ode a datas no Rei só de textos e como eu falei a performance dele quando nós estamos trabalhando com muitos e muitos dados realmente dá um bom ganho no contexto de Universidade é como estamos em uma rota de aprendizagem é muito provável que você não enxergue muita diferença entre um e outro mas realmente a utilização de um pai quando nós trabalhamos com uma grande massa de dados realmente se demonstra né então fica muito mais visível Além disso O que é importante desse saber sobre esses direitos e como eles são mais rápidos que uma lista então a gente consegue criar operações matemáticas Eu quero um presente transpor quero mudar os elementos que eram originar eu quero filtrar rapidamente dentro deles imagine várias operações que para implementar na mão com listras seria muito Moroso é isso é muito complexo então no no pai nesses fez isso já fica implementado e também ajuda muito o nosso trabalho além disso eles ocupam menos espaço em memória como eu falei uma grande base de dados e isso acaba ficando mais visível e também a gente consegue utilizar esses arreios como base de outras bibliotecas então tenho pandas tem nessa quiche Lorraine tem o site pai até o Estados modernos várias outras bibliotecas que existem em Python utilizando esses Reis com base para rapidamente fazer operações são científicas operações que estão Haddad da estatística na área de Inteligência Artificial de matemática para resolver esses problemas Além disso temos também outras operações que podem ser muito úteis a com trabalho no pai então exemplos rápido SAC como fazer eu posso ordenar os elementos a Imagine você tendo uma lista padrão lá em Python se tentando ordenar na mão na criando assim um código você mesmo para tentar bioelementos um milhão de alimentos Possivelmente ele demorar muito tempo então não vai também já inclui isso a temos também um ter ações que são de cunho algébricos tecnológico esse cunho Lógico eu acredito que ele é bem útil no dia a dia para filtrar informações exemplo prático Imagine que você tem de novo aquela aquele conjunto de notas de sorvetes e você precisa encontrar se alguém de uma nota negativa porque é um erro uma nota tem que ser de 0 Até Dez Então se precisa encontrar todos os seus elementos que são ou abaixo de zero ou acima de 100 ou acima de 10 então com essas operações lógicas doce também consegue rapidamente uma onde que são as notas ou ainda né Quais são esses notas e aplicar diferentes cálculos ou encontrado em algum algum valor que não seria o número na o valor que não é o infinito né Foi um erro lá de conversão então encontrar tudo isso e gerenciar isso no código é muito fácil temos operações estatísticas isso também acaba sendo muito útil para encontrar meu celular Diz que são muito abaixo acima do que estávamos esperando pense de novo no caso do sorvetes Imagine que a escala de notas de 0 até dez e aí a gente descobriu que a média das notas é 8 o valor mínimo dessas notas foi -2 e o desvio padrão 910 você mente tem uma coisa errado nessas informações de Imagine que essas descrições estatísticas com aqueles graxos e comentamos anteriormente é toda essa essas bibliotecas adicionais ajudam bastante no nosso dia a dia para realmente encontrar possíveis problemas nos dados e com isso temos aí códigos que são mais eficazes tudo bem lembra que eu falei anteriormente sobre esses arreios que eles são muito úteis são utilizados como base de outros outras Vidas Secas então naturalmente a gente vai falar sobre tabelas também E por que que o Diego naturalmente Imagine que uma lista é um conjunto de elementos eu também posso ordenar eles como se fosse uma linha então se eu posso ter várias linhas nesse dados eu voltei naturalmente uma tabela então é por isso que estamos aqui falando também sobre tabelas me acompanha aqui no slide por favor vendo que por exemplo de tabelas nessa manipulação temos outra Biotec super legal que se chama pandas essa biblioteca é o padrão quando nós trabalhamos aí corrigir dados com ciência de dados e por que ela permite a manipulação rápida e de várias cidades que estão como fazer em formato de tabela os dados tabulares essa peteca foi construída em cima de um pai então quando se instala o panda se o seu computador você também vai instalar um painel acaba daqueles requisitos uma daquelas dependências E ele nos dois outros tipos de dados o primeiro a Data Frame e o segundo é o Siri imagina que um Data Frame é uma tabela então Data Frame vai ter aí a linhas vai ter colunas e cada uma dessas colunas é um é um desses seres não seja é um conjunto de dados então a gente também consegue fazer várias operações que poderíamos ter demorar muito tempo no Excel e fazer rapidamente em Python seja aquelas formas bem complexas que trabalho invariavelmente né várias pessoas tem aquela planilha de Excel de estimação nas várias áreas de negócio e não consegue não querem abandonar a gente também consegue a automatizar isso avidamente com o pandas olhando aqui também a Que tipo de possibilidade Se nascemos de importação de dados nós conseguimos ler informações de diferentes Fontes com o pandas então a gente pode ler onde o arquivo csv o Jason esses tipos arquivos eles geralmente são formatos de dados né então posso ter dentro dele um grande conjunto de linhas e colunas que com isso nós construímos esse Data Frame eu posso ter por exemplo eu posso me conectar a um serviço da internet de metrologia e pegar a previsão do tempo para os próximos dias dessa previsão do tempo virar injeção é um arquivo que a gente não consegue ler enquanto humanos né de uma forma fácil mas conseguimos importar isso não planas e visualizar isso realmente em uma tabela também conseguimos Ler arquivos de Excel ou até mesmo Conectar em base de dados lá naquele então aqui já é um ponto bem legal nós conseguimos diretamente conectar um banco de dados no Python por intermédio do Plantas Versus tabelas.net o traço informações e manipulado entre Python então como eu falei lei de um banco de dados executar diferentes filtros dentro do ânus criar novas colunas até criar o eixos preditivos utilizando panas então ele é uma boa base nesse sentido e como falei na sua utilização do dataframe ela é muito interessante de trabalhar no SAC com python porque como eu falei é um formato de tabela nós conseguimos manipular tabelas que são bem grandes quando eu falo grandes aqui imagina se na casa das dezenas ou centenas de milhões de linhas ou até mesmo a centenas ou milhares de colunas e com isso a gente consegue manipular aqui dentro do plantas também a gente consegue fazer alguns Alguns agrupamentos algumas operações que elas são um pouco mais complexas de serem feitas imagina Nextel ou na mão exemplos básicos aqui temos por cento ou porcento a gente consegue em 1 a partir de uma coluna que é do tipo data imagina se a previsão do tempo hora a hora na sua cidade eu consigo agrupar kilo por semana ou por mês para ter uma tabela bem menor né bem mais entre aspas achatada então a gente também consegue fazer essas operações facilmente com urbanas ou ainda né fazer um grupo aquelas tabelas dinâmicas que você já deve ter visto no céu também a gente consegue juntar e diferentes tabelas né criar nossas novas informações também é possível aqui com o plantas esse Red e o que eles são funções bem úteis tão tudo que a gente tá vendo aqui são funções o Red Taylor são outras funções interessantes para gente visualizar os dados de uma tabela tem centenas de milhares de dados mas eu não quero ver tudo aquilo quero ver uma pequena mostra o header nos permite enxergar as primeiras linhas por padrão as primeiras cinco linhas e odeio as últimas cinco linhas coloque o armlock também são parecidos eu consigo visualizar por Vince ou por linhas ao quero ver a linha 500 até 520 vai lá que nos permite eu quero ver os as 20 primeiras linhas também um Lock nós conseguimos nós conseguimos fazer isso Além disso com panos a gente tem outras funções que são específicas de manipulação na dados categóricos ou de outros tipos de dados esse gratidão por exemplo imagina só eu tenho lá de novo aquele exemplo das notas de sorvete posso ter cinco sabores diferentes de sorvete eu tenho uma tabela com todas as notas e as pessoas deram para esses diferentes tipos eu tenho uma colônia chamada tipo de sorvete eu quero agora transformar essa coluna em cinco novas colunas e chocolate sim ou não é morango sim ou não é banana sim ou não Então pegando essas diferenças crianças diferentes colunas acaba sendo muito útil no ponto de vista de ciência de dados nas de manipulação de análise de dados e o que ele faz Exatamente isso para nós e finalmente nós também a partir de uma data conseguimos extrair novas colunas e também nos ajuda nos trabalhos de análise de dados o.ic para exemplo extrai a semana daquela referente pela data odeia só o dia do mês só um mês e somente o dia da semana então essas operações elas são muito úteis até que eu falei para a previsão quero prever o comportamento de vender sorvete isso pode aumentar no final de semana então às vezes tem uma coluna específica do dia da semana pode ser muito útil para o nosso corpo e também uma outra funcionalidade super legal que o thanos tem é para as correlações por padrão a gente tem três tipos de correlação as mais comuns aberson acredito a mais comentada aí né mas ela mais simples que tem uma tem um relacionamento linear e no dia a dia nem sempre todos esses relacionamentos são lineares Ou seja eu posso ter uma curva e aqui me acompanha até na minha mão posso ter cinco comportamentos que eles não são necessariamente lineares ou seja enquanto um aumenta o outro não necessariamente aumenta mas eu passei o comportamento que é mais uma curva né então não é tão fácil assim da gente presente ou ainda ao que pode ser uma sanfona Enfim então tenho outros comportamentos que não são tão bonitos net serem visualizados mas ainda assim 41 correlação então é para isso que tem outras formas da gente criar nessas diferentes relações dessas análises eu pessoalmente gosto muito da spirulina e da Carol por que ajudam é muito no dia a dia e como fazer tantas firma com uma queda ou ela leva em consideração outros casos que não são tão lineares né como eu falei não é aquele relacionamento aquela reta bonitinha que a gente vê e sim outros relacionamentos que são mais próximos de casos reais que realmente tem muito ruído tem outros fatores externos é mais difícil da gente escrever então ainda sim tem vários casos a de correlação não necessariamente existem na localização alta via pirson e mais que pode ter de esperma ou porque tudo bem toque que eu gostaria de discutir com você visualiza aqui comigo primeiramente a sobre expressões regulares porque é interessante de conversarmos sobre isso primeiramente uma expressão regular ou que eu vou começar a falar com você no regex porque a expressão regular regular Expression e quando você pesquisar internet vai encontrar muita documentação sobre regex então é por isso eu começar a falar com esse nome O Rex ele seria uma uma receita de um padrão de busca de uma gênia sim ser estar em um artigo da Wikipédia lendo aí um um grande livro ou não é um artigo científico e tem que encontrar todas as ocorrências de alguma coisa ou ainda vamos imaginar outro cenário eu tenho uma uma lista de Um Diário Oficial ou algum documento governamental que tem várias coisas né Eu posso ser textos eu posso ter Gomes eu posso ter números e você encontrar todas as ocorrências de uma placa de carro mas não é toda placa de carro toda combinação existentes a tudo que seria aquele padrão de uma placa então a a 0000 é uma placa mas também posso ter 1000 1000 dois tão como eu tenho que diferentes padrões melhor diferentes valores mas seguindo o mesmo padrão vale mais a pena investir nos nossos esforços em tentar encontrar aquele padrão do que tentar encontrar todas as combinações existentes então o exército ele nos ajuda a economizar muito tempo e encontrar realmente tudo que seguir aquele padrão Então veja que é um pouquinho diferente de encontrar uma ocorrência e o valor que nós sabemos exatamente o que é esse em todas as variações daquilo Então até o nome que nós utilizamos para isso depois que encontramos o resultado temos um nome daquilo deu mexe na encontram-se aumente um valor aí a gente consegue visualizar entendem em qual parte foi encontrada a quilo Em qual contexto E aí sim tratar aquela informação isso para ajudar um pouco de medo a trabalhar com regex porque você começa a versão vários códigos vários letrinhas caracteres e você não sabe muito bem o que tá acontecendo lá então a princípio ele pode não ser muito legível mas ele te ajuda muito até uma velocidade a encontrar filtrar informações e naturalmente a construção de algoritmos que encontra as informações de uma forma rápida parecerem muito úteis para o seu trabalho de 16 com pente novamente nos lares para visualizar alguns exemplos esse primeiro exemplo abre colchete fecha colchete e dentro do Aterro 0-9 Imagine que estar acima é sempre um até então vamos ver isso com 10 esse primeiro item do 0-9 de 0 até 9 e o abre colchetes fecha colchetes é tudo é um valor de um caracter que esteja dentro desse padrão tem como que nasce lemos esse primeiro caso seria vamos encontrar qualquer valor esteja entre 0 até 9 em 1 0 1 2 3 4 até chegar no 9 o segundo caso ele é uma variação dele então pode ser de 0 até 9 ou qualquer letra minúscula do alfabeto inglês tem acento na então do ABC até chegar lá no na letra z minúscula Agora se eu quero encontrar qualquer digito ou letra estando no maiúsculo minúsculo seriado 0 até 9 que as terceiro Exemplo né A até Z minúsculo e ainda a até Z maiúsculo então aqui a gente pega né todo esse alfabeto encontrou um caracter Esteja dentro dessa desse dessa receita Você já consegue já deu mexe né E já encontrou essa informação para você naturalmente não vai incluir aí espaço ponto vírgula então toda isso que não se encaixam nesse padrão não será retornado para você esse parte sempre é um pouco mais interessante Então é esse acento circunflexo indica que tudo que não está nesse nesse padrão que vem logo a seguir Ou seja você está interessado em tudo que não é 0 até 9 e do a minúsculo até o Z minúsculo então pode ser a maiúsculo pode ser Vírgula pode ser ponto pode ser espaço tudo isto entra nesse padrão E aí dá certo temos ainda outras variações esse quinto item o maizinho ele significa que é um ou mais caracteres juntos então se eu tenho para exemplo número 1000000 e 1 seguido de 100 zeros todos eles estão juntos vai dar mexe aí porque é um conjunto de dígitos o antepenúltimo item tem um asterisco ele é uma variação disso e a 0 ou mais itens disso então aí eu posso encontrar se eu tenho por exemplo a 50 dígitos tem uma vírgula e mais 30 dígitos eu mais queria um ou mais o asterisco seria 0 ou mais e por outro lado esse penúltimo Exemplo né que tem o ponto de interrogação o ponto de interrogação e significa basicamente 101 então eu posso encontrar aquilo ou posso não encontrar aquilo isso é muito útil por exemplo em dígitos em que eu tenho em alguns casos eu posso ter separador de milhar que seria aquele ponto né ou dependendo do idioma pode ser uma vírgula e também pode ser valores que não tem esse separador então para pegar todos os carros que a gente consegue deixar aquela vírgula naquele ponto como opcional e ainda se entregar todo o valor então é por isso né que servem essas receitas e por último aqui no slide né a gente tem um exemplo de validação de CEP temos pode ver que ele é um pouco mais complexo como que nós interpretamos isso aquele barra invertida de é dígito Então esse bairro invertida de é a mesma coisa daqueles 0 até 9 que você viu no primeiro item é exatamente a mesma coisa só outra forma de escrever aqueles cinco entre Chaves significa que são cinco então basicamente temos que encontrar uma sequência de 0 até 9 repetido cinco vezes que é justamente aqueles cinco primeiros dígitos de um CEP Depois temos que encontrar hífen e depois do hífen temos que encontrar mais três dígitos como você deve imaginar o CEP de todo o Brasil segue Exatamente esse padrão então a gente consegue aqui vai conhecer o último exemplo que você está vendo na sua tela visualizar todos os apps que significa padrão seja que não tem ponto né alguma coisa que siga fora mais se for 5 dígitos sem espaço sem nada seguidos de um traço e mais três dígitos esse rack cereja vai encontrar todas as ocorrências que seguem esse padrão aí para você beleza

**CONVERSA INICIAL**

Olá! Esperamos que esteja tudo bem com você. Uma das frases normalmente comentadas por professores, estudantes e praticantes de TI em qualquer lugar do mundo é a seguinte: “quem domina uma linguagem de programação consegue facilmente dominar outras”. Essa afirmação, naturalmente, não é uma regra: existem linguagens e linguagens de programação. Umas são mais simples, e outras são muito mais complexas. C++ é uma linguagem diferente do Perl, que é diferente do Python, que é diferente do PHP, que é diferente do C#, por exemplo. Por outro lado, poderíamos argumentar que essa frase pode ser compreendida mais ou menos dessa forma: “quando você sabe os conceitos básicos de uma linguagem de programação, como variáveis, estruturas de seleção e/ou repetição, bibliotecas, e sabe ler a documentação, você tem menos dificuldade para entender esses mesmos conceitos em novas linguagens de programação porque agora você já conhece o conceito e já colocou em prática o que aprendeu pelo menos uma vez no passado”.

E é para isso que começamos com Python: é rápido começarmos a criar um algoritmo e, além disso, não há a obrigatoriedade de instalar algo no computador para começar a testar: o Google Colab[[1]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn1" \o ") é uma ótima porta de entrada para isso. Há uma ampla documentação disponível e uma comunidade forte. Além disso, já é utilizada por várias empresas para o desenvolvimento de algoritmos.

Quando aprendemos Python (ou qualquer outra linguagem), geralmente somos apresentados aos conceitos básicos para darmos os primeiros passos. Isso inclui, por exemplo, a entrada e saída de dados (como digitar o seu nome e mostrá-lo logo em seguida com o *input*e *print*), as operações com variáveis (como multiplicar dois valores que um usuário digitou), estruturas de seleção (com o *if, elif*e *else*), estruturas de repetição (como o *for, while*e *while True*), listas e funções. Mas isso é só o começo: serve para você se**ambientar à base** de uma linguagem de programação em específico.

E é muito provável que alguém (ou você mesmo!) tenha feito a seguinte pergunta: “como consigo fazer um site com isso que aprendi? Como consigo mostrar um gráfico? Como consigo tirar uma foto com a minha webcam? Como crio um jogo? Como crio um sistema para gerenciar uma loja? Como crio um sistema para armazenar o usuário e senha de um conjunto de usuários em Python?”.

Note que estes exemplos vão além do que vimos até o momento: é necessário, portanto, que saibamos **como****colocá-los em prática**. E é por esse motivo que estamos aqui. Vamos lá?

**TEMA 1 – EXPANDINDO O NOSSO REPERTÓRIO COM AS BIBLIOTECAS**

Vamos sair um pouco do universo de TI e fazermos juntos um pequeno exercício de pensamento: vamos supor que a torneira da sua cozinha começou a vazar água pela parede. E, para ajudar, você nunca trocou uma torneira na vida e não sabe nada de hidráulica. E agora, o que você faz? Temos aqui algumas alternativas:

1. Você compra uma nova torneira e tenta você mesmo fazer a troca, sem pedir ajuda a ninguém. Nisso, descobre que o problema era mais simples do que pensava e acabou gastando dinheiro à toa.
2. Você mesmo resolve tirar a torneira antiga, sem pedir ajuda a ninguém. Com isso, acaba estourando parte da parede e do encanamento, deixando o problema ainda pior. Agora, precisará chamar um profissional e gastar muito mais dinheiro do que imaginava.
3. Você vai ao YouTube e procura por tutoriais de troca de torneiras que foram bem avaliados. Você tenta reproduzir com cuidado o passo a passo e o adapta a suas próprias necessidades. Não fica perfeito no começo, mas você estará mais bem preparado para as próximas vezes.
4. 4. Contrata alguém para fazer o trabalho por você.

**1.1 PARA QUE SERVEM AS BIBLIOTECAS?**

E se, ao invés de falarmos de torneiras, imaginássemos que você conseguiu um estágio para uma equipe de tecnologia e lhe deram uma tarefa para criar um gráfico em Python logo de cara? Temos também algumas alternativas:

1. Você tenta criar um código para gerar o gráfico usando somente os *if/elif/else*e *for/while/while True*, listas e funções. Depois de ficar toda a semana batendo a cabeça, você percebe que ainda não conseguiu criar gráfico algum.
2. Você tenta criar um código para gerar o gráfico usando os *if/elif/else*e *for/while/while True*, listas e funções. Ao pesquisar na internet, você descobre que existem algumas formas para gerar isso. Os exemplos que você pegou funcionam, mas você não consegue integrar ao código que já possui. Logo, você fica sem saber como adaptar o problema.
3. Você pesquisa na internet para ver como é possível fazer isso em Python. Você descobre que existem **várias**formas e com várias combinações e ajustes possíveis.
4. Você diz que é impossível, e é isso aí.

Essa última alternativa, felizmente, não é uma opção: é possível sim! Para responder a isso, vamos pensar em partes. Você já deve ter se deparado no passado com as **funções**em Python. Elas possuem um objetivo muito legal: o reaproveitamento de código. Com as funções, conseguimos colocar dentro de uma caixa fechada uma lógica que precisa ser **reaproveitada**. Observemos o código a seguir, que reajusta o salário de diferentes cargos se eles estiverem acima de um salário-base.

salario_base = 3000
salario_analista = 3500
salario_estagiario = 1500
salario_lider = 9000

# reajuste salarial
reajuste = 0.035
if salario_analista > salario_base:
  salario_analista *= reajuste
  print(f'Novo salario: {salario_analista}')

if salario_estagiario > salario_base:
  salario_estagiario *= reajuste
  print(f'Novo salario: {salario_estagiario}')

if salario_lider > salario_base:
  salario_lider *= reajuste
  print(f'Novo salario: {salario_lider}')



A lógica utilizada para o reajuste se repete várias vezes, não é mesmo? Logo, podemos **refatorar**o código (ou seja, melhorá-lo sem alterar o resultado esperado) para usar funções (no caso, a função *reajustar\_salario*).

salario_base = 3000
salario_analista = 3500
salario_estagiario = 1500
salario_lider = 9000

# reajuste salarial
reajuste = 0.035
def reajustar_salario(salario, valor_base, valor_reajuste):
  if salario > valor_base:
    salario *= valor_reajuste
    print(f'Novo salario: {salario}')

  return salario

salario_analista = reajustar_salario(salario_analista, salario_base, reajuste)
salario_estagiario = reajustar_salario(salario_estagiario, salario_base, reajuste)
salario_lider = reajustar_salario(salario_lider, salario_base, reajuste)


E se adicionássemos mais 20 cargos? E se precisássemos mudar a lógica para aplicar o reajuste? Com o uso das funções, não é preciso muito esforço: conseguimos mudar a lógica em um lugar só (no caso, dentro da função *reajustar\_salario*) e os ajustes funcionarão para todo o código que vier a chamar essa função. Não precisamos também copiar e colar a mesma lógica de **dentro**da função para usarmos em múltiplos lugares: basta chamar a função em si mais uma vez (ou quantas vezes forem necessárias).

No passado, algumas pessoas desenvolveram funções só para converter uma *string*para um valor inteiro (a função *int***[[2]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn2" \o ")**); para converter uma *string*para um valor decimal (a função *float***[[3]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn3" \o ")**); para mostrar algum texto no terminal ou em um notebook de código (a função *print***[[4]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn4" \o ")**); para ler algum dado que um usuário venha a digitar (a função *input***[[5]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn5" \o ")**), e assim por diante. Note que, no final das contas, trabalhamos até o momento com uma **coleção** de funções que várias outras pessoas desenvolveram e que podemos reaproveitar em vários algoritmos. Essas coleções (ou agrupamentos) possuem outro nome: *módulos*, que, por sua vez, módulos são agrupados em **bibliotecas**(ou *libraries*).

**1.2 BIBLIOTECAS EM PYTHON**

Em Python, existem bibliotecas especializadas para inúmeras aplicações: para mostrar gráficos (como é o caso do problema que comentamos anteriormente); para manipular fotos; para manipular vídeos e músicas; para criar jogos; para comunicar com bancos de dados; para criar algoritmos de inteligência artificial, entre outras.

As bibliotecas possuem algumas vantagens, como:

1. Você consegue rapidamente adicionar funcionalidades no seu código que já foram testadas e desenvolvidas por várias outras pessoas.
2. Você consegue testar diferentes implementações que melhor se adaptem às suas necessidades: em Python, por exemplo, existem dezenas de bibliotecas que tratam somente da geração de gráficos. Uma biblioteca que é preferida para outras pessoas pode não necessariamente ser a sua favorita, e vice-versa.
3. Você consegue instalar e desinstalar rapidamente as bibliotecas de acordo com as suas necessidades.

O Python já possui uma biblioteca padrão[[6]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn6" \o ") e funções como a *input, print, float*e *int* fazem parte dela. As bibliotecas também podem incluir tipos de dados além das listas, matrizes, *strings*, datas e números. Veremos mais sobre isso no decorrer desta aula.

**Saiba mais**

Um bom conselho é explorar a documentação das bibliotecas para conhecer suas possibilidades de uso.

Lembra que comentamos que as bibliotecas são divididas em módulos? A biblioteca padrão do Python possui um módulo especializado[[7]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn7" \o ") somente em manipulação de arquivos compactados do tipo zip; outro[[8]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn8" \o ") especializado somente em listar todos os arquivos ou pastas em determinado local; outro[[9]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn9" \o ") módulo especializado em operações matemáticas; outro[[10]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn10" \o ") módulo especializado em operações estatísticas; e assim por diante. Saber ler essas documentações e encontrar a melhor biblioteca, módulo e função para o que você precisa pode fazer toda a diferença no seu trabalho.

Existem centenas de milhares de bibliotecas disponíveis, que, geralmente são instaladas com o comando *pip*. É possível sabermos quais são as mais utilizadas com base em rankings facilmente encontrados na internet. Uma busca por “pypi download stats” no Google retorna vários desses rankings, por exemplo. Algumas dessas servem para gerar gráficos como o **matplotlib, seaborn**e o **plotly**.

**TEMA 2 – CRIANDO GRÁFICOS COM PYTHON**

Uma das formas de encarar o desenvolvimento de software é pensar que fazemos este trabalho para **resolver problemas**: seja um aplicativo de internet banking, um software que gerencia uma fábrica ou um algoritmo de inteligência artificial que prevê problemas nas vendas. Logo, tão importante quanto desenvolvermos um **código bom** é **mostrar que estamos resolvendo um problema.**

Naturalmente, essa definição muda muito de caso para caso: um desenvolvedor de back-end ou um administrador de banco de dados raramente vão se deparar com um cenário no qual terão de gerar algum gráfico em Python. Por outro lado, um engenheiro de dados, um cientista de dados ou um analista de dados precisarão fazer isso **muito**. Ou, ainda, pessoas que estejam desenvolvendo pesquisas de natureza quantitativa (o que inclui um mestrado ou doutorado, por exemplo). É claro que existe o Excel (ou qualquer software de planilhas) para criar alguns gráficos rapidamente, mas ele não gera todos os tipos de gráficos de que poderíamos necessitar. Além disso, existem casos em que precisamos nos conectar diretamente a bases de dados ou manipular um grande volume de dados, que o Excel simplesmente não suportaria.

**2.1 BIBLIOTECAS MAIS CONHECIDAS**

Em Python, três bibliotecas se destacam: o matplotlib, o seaborn e o plotly. Existem várias outras, como o bokeh, yellowbrick e Altair, mas essas três primeiras são as mais conhecidas:

1. O matplotlib[[11]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn11" \o ") foi criado em 2003 e se tornou a biblioteca de scatter plot mais usada em Python. No entanto, ela é bem complexa: imagine um painel de um grande avião, cheio de botões e configurações. Muitas são as possibilidades, mas também há uma curva de aprendizagem considerável.
2. O seaborn[[12]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn12" \o ") foi criado quase dez anos depois. Ele usa o matplotlib como base, mas simplifica várias das operações. Ele é bem mais simples de usar e pode gerar gráficos mais bonitos.
3. O plotly[[13]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn13" \o ") (e em específico o *plotly express*) pode ser simples como o seaborn, mas traz uma diferença: a interatividade. Com o plotly é possível criar animações e interações nos gráficos (como passar o mouse e ver mais informações, por exemplo).

**2.2 EXEMPLOS**

Vamos supor que queiramos criar um gráfico contendo a visualização de diferentes notas e preços de cafés. Temos essas notas e preços armazenados em duas diferentes listas:

preco = [5.29, 6.49, 7.49, 7.99, 8.49, 9.99, 11.99]
nota = [5.5, 7.0, 7.5, 8.5, 9.5, 9.0, 9.5]



Observe o código utilizado para criar os gráficos utilizando os mesmos dados, mas com bibliotecas diferentes.

import plotly.express as px

fig = px.scatter(x=preco, y=nota,
                 title='Comparação entre preço e nota de cafés',
                 labels={"x": "Preço", "y": "Nota"})
fig.show()



Figura 1 – Comparação de preço e nota de cafés (1)



Fonte: Monteiro, 2021.

import matplotlib.pyplot as plt

plt.scatter(x=preco, y=nota)
plt.title('Comparação entre preço e nota de cafés')
plt.xlabel('Preço')
plt.ylabel('Nota')



Figura 2 – Comparação de preço e nota de cafés (2)

Gráfico, Gráfico de dispersão

Descrição gerada automaticamente

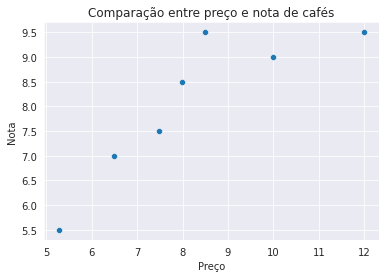
Fonte: Monteiro, 2021.

import seaborn as sns
import matplotlib.pyplot as plt

sns.set_style('darkgrid')
sns.scatterplot(x=preco, y=nota)
plt.title('Comparação entre preço e nota de cafés')
plt.xlabel('Preço')
plt.ylabel('Nota')



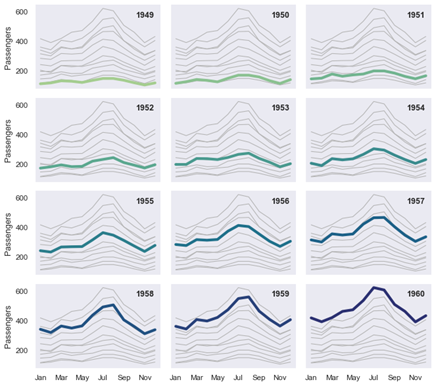
Figura 3 – Comparação de preço e nota de cafés (3)



Fonte: Monteiro, 2021.

Esse tipo de gráfico (dispersão, ou *scatter plot*) é um dos mais simples. Por outro lado, existem outras visualizações mais interessantes como esta, do seaborn:

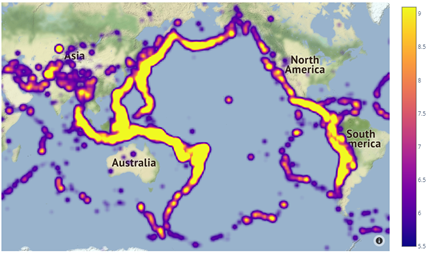
Figura 4 – Seaborn



Fonte: Seaborn, [S.d.].

Ou esta, do Plotly:

Figura 5 – Plotly



Fonte: Plotly Technologies, Inc., 2015.

Naturalmente, para criar gráficos, precisamos de dados. O breve exemplo que criamos anteriormente lida com duas listas: uma chamada *preco* e outra chamada *nota*. Mas será mesmo que o uso de duas listas seria a melhor alternativa? Seria prático permanecermos utilizando listas mesmo se tivéssemos alguns milhares ou milhões de notas? Se você já tentou manipular listas com um tamanho tão alto, deve ter se deparado com problemas de usabilidade e de performance. Felizmente, há uma saída: a biblioteca NumPy.

**TEMA 3 – RESOLVENDO O PROBLEMA DAS LISTAS**

Várias bibliotecas usadas para inteligência artificial (incluindo redes neurais), visualização, análise de mapas, bioinformática, astronomia e demais áreas utilizam NumPy[[14]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn14" \o "). E isso acontece porque essa biblioteca foi pensada para resolver problemas grandes e complexos com o menor esforço possível por parte do desenvolvedor. Para deixar isso mais claro, sugerimos que voltar ao exemplo do café.

preco = [5.29, 6.49, 7.49, 7.99, 8.49, 9.99, 11.99]
nota = [5.5, 7.0, 7.5, 8.5, 9.5, 9.0, 9.5]



Cada uma dessas listas possui sete valores e são listas em Python. Agora, imagine fazer as seguintes operações nelas:

1. Multiplicar todos os elementos de uma lista;
2. Encontrar a média e desvio-padrão dos valores de uma lista;
3. Descobrir se algum dos elementos da lista é um número infinito;
4. Obter o valor absoluto de todos os elementos da lista;
5. Obter todos os elementos que sejam somente acima de 9;
6. Remover todos os valores em duplicata;
7. Inverter a ordem de todos os valores.

É bem provável que consigamos fazer rapidamente e com uma só linha de código algumas dessas operações, mas não todas. O NumPy disponibiliza uma estrutura de dados similar às listas: o *ndarray***[[15]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn15" \o ")**. O uso do *ndarray*em relação às listas possui algumas vantagens, como:

1. **Funcionalidades**: já existem funções prontas para manipular rapidamente todos os elementos de um *ndarray*. Essas aplicações incluem operações algébricas, estatísticas, manipulações de dados e operações matemáticas. Na prática, essas operações são úteis para pesquisadores a nível de mestrado e doutorado; estudantes em cursos de engenharia e demais cursos aplicados na área de exatas; estatísticos e economistas; desenvolvedores de aplicações back-end que necessitam manipular uma grande massa de dados; cientistas e analistas de dados, entre outros. Logo, o *ndarray* é mais prático do que as listas do Python;
2. **Performance**: quando precisamos manipular uma grande massa de dados (na casa das centenas de milhares de entradas ou mais), o *ndarray* geralmente possui uma implementação mais otimizada. Logo, o *ndarray*tende a ser mais rápido do que as listas do Python;
3. **Consumo de espaço**: além disso, o *ndarray*é também otimizado para grandes conjuntos de dados. Uma prova disso é a de que o uso do NumPy foi essencial para gerar a primeira imagem de um buraco negro conhecida pela humanidade[[16]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn16" \o "). Logo, o *ndarray*consome menos espaço em memória do que as listas do Python;
4. **Reaproveitamento do conhecimento**: toda a base de conhecimento que temos sobre listas é usada para trabalharmos com *ndarray*.

Note como as duas listas anteriores seriam implementadas em *ndarray*:

import numpy as np

preco = np.array([5.29, 6.49, 7.49, 7.99, 8.49, 9.99, 11.99])
nota = np.array([5.5, 7.0, 7.5, 8.5, 9.5, 9.0, 9.5])


E é isso! Agora, e se precisássemos ler muito mais dados? E se precisássemos ler um arquivo CSV ou uma planilha do Excel? E se quiséssemos manipular os dados no Python como se fosse o Excel, o que faríamos? É para isso que serve o **pandas**.

**TEMA 4 – TRABALHANDO COM TABELAS EM PYTHON**

A manipulação de dados em formato tabular (isto é, em tabelas) é facilitada pela biblioteca pandas. Em conjunto com ela, outras bibliotecas de cunho estatístico e gráfico podem ser combinadas para analisar os dados, manipular os dados, gerar correlações e relatórios de estatística descritiva.

**4.1 PANDAS**

O pandas[[17]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn17" \o ") é uma biblioteca que foi criada utilizando o NumPy como base. Da mesma forma que uma faca é uma ferramenta que todo cozinheiro possui e um alicate é uma ferramenta que todo eletricista possui, o pandas é uma biblioteca[[18]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn18" \o ") que todo desenvolvedor que atua com dados em Python conhece e trabalha.

O pandas introduz um tipo de dados muito poderoso chamado *DataFrame***[[19]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn19" \o ")**. Para falar do DataFrame, vamos voltar ao exemplo do café. Lembre-se que tínhamos duas listas, *preco*e *nota*:

preco = [5.29, 6.49, 7.49, 7.99, 8.49, 9.99, 11.99]
nota = [5.5, 7.0, 7.5, 8.5, 9.5, 9.0, 9.5]


Nunca deixamos isso explícito, mas **é possível entender**que o primeiro valor do preço corresponde ao primeiro valor da nota; o segundo valor do preço corresponde ao segundo valor da nota; e assim sucessivamente. Agora, e se eu quisesse ordenar de forma decrescente o preço, alterando também a ordem da nota? Da forma que está, isso não seria possível. E se quisesse descobrir rapidamente o preço correspondente à mediana das notas? Também não seria tão simples. Por outro lado, em Python também podemos ter matrizes (isto é, listas de listas).

Então, poderíamos organizar da seguinte forma:

precos_notas = [[5.29, 6.49, 7.49, 7.99, 8.49, 9.99, 11.99],
                [5.5, 7.0, 7.5, 8.5, 9.5, 9.0, 9.5]]



Agora temos somente uma variável, chamada *precos\_notas*. Mas e se quiséssemos incluir a data da coleta das notas? Ou, ainda, se quiséssemos aplicar uma operação para *todas*as notas (um pequeno reajuste de 0.1 pontos, talvez)? Ou, quem sabe, se quiséssemos **dar um nome**para esses dados, como seria?

É para responder a essas questões que existem os DataFrames. Com uma linha (na maioria das vezes), conseguimos ler planilhas e conjuntos de dados inteiros para uma variável, e vice-versa. Quer ler uma tabela em SQL e converter para um DataFrame? É possível. Quer ler uma planilha do Excel? Dá sim! Um JSON, quem sabe? É claro! As possibilidades são múltiplas. Vamos ver como ficariam os mesmos dados dos preços e das notas, mas em DataFrame:

import pandas as pd

precos_notas = pd.DataFrame([[5.29, 5.5], [6.49, 7.0], [7.49, 7.5], [7.99, 8.5], [8.49, 9.5], [9.99, 9.0], [11.99, 9.5]],
                            columns=['Preço', 'Nota'])


Tabela 1 – Preço e nota



E se quiséssemos criar uma data fictícia para a coleta dessas notas?

precos_notas['DataColeta'] = pd.date_range(start='2022-01-01', freq='M',
                                           periods=len(precos_notas))


A função *date\_range***[[20]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn20" \o ")** cria uma sequência de datas. Podemos informar uma data de início e de fim ou, ainda, uma frequência entre as datas. No caso, preferimos escolher somente uma data de início (2022-01-01) e uma frequência mensal, pegando sempre o último dia de cada mês (com o parâmetro ‘M’). A quantidade de datas foi definida pela quantidade de linhas disponíveis no dataframe (pelo *len(precos\_notas)*).

Tabela 2 – Preço, nota e data



E se quiséssemos criar colunas para extrair somente o ano, ou somente o dia, ou somente o dia da semana da data de coleta? Ou, ainda, a proporção da nota em relação ao preço? Vamos lá:

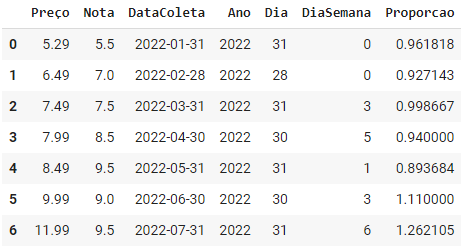
precos_notas['Ano'] = precos_notas['DataColeta'].dt.year
precos_notas['Dia'] = precos_notas['DataColeta'].dt.day
precos_notas['DiaSemana'] = precos_notas['DataColeta'].dt.weekday
precos_notas['Proporcao'] = precos_notas['Preço']/precos_notas['Nota']



Agora, vamos pensar um pouco: já imaginou como seria fazer tudo isso sem o pandas, somente manipulando as listas em Python? Há ainda uma função no pandas para nos fornecer dados estatísticos sobre esse DataFrame: o *describe***[[21]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn21" \o ")**. Observe:

precos_notas.describe()

Tabela 3 – Preço, nota, data e proporção



Aqui, temos dados como a quantidade de valores para cada coluna (*count*), a média (*mean*), o desvio-padrão (*std*), os valores mínimo (*min*) e máximo (*max*), a mediana (50%) e os quartis inferior (25%) e superior (50%). Além disso, veja que na própria documentação[[22]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn22" \o ") existem ainda múltiplos exemplos para operações, como agrupamento, deslocamento de colunas, ordenação, preenchimento de valores nulos, remoção de linhas repetidas, arredondamento, correlação, entre outros. Falando em correlação, vamos testar uma coisa.

**4.2 ESTUDANDO CORRELAÇÃO COM DIFERENTES BIBLIOTECAS**

Quando falamos em **correlação**, queremos, na prática, entender o quanto uma variável afeta outra. Por exemplo, será que, toda vez que o preço aumenta em 10%, a nota também aumenta (ou diminui) na mesma proporção? Com uma única linha, o pandas consegue calcular isso[[23]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn23" \o "). Veja só:

precos_notas.corr(method='spearman')



Tabela 4 – Correlação

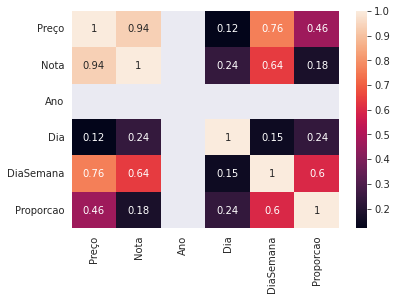


Quanto mais próximo de 1, maior a correlação diretamente proporcional (ou seja, quando um aumenta, o outro aumenta também). Quanto mais próximo de -1, maior a correlação inversamente proporcional (ou seja, quando um aumenta, o outro diminui). Aparentemente, o preço tem uma correlação muito alta com a nota, mas parece que o dia da semana também teria uma correlação interessante. Ficou difícil de enxergar vários números? Vamos ao seaborn[[24]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn24" \o "):

sns.heatmap(precos_notas.corr(method='spearman'), annot=True)



Figura 6 – Seaborn



Perceba que, aos poucos, vamos também **misturando**bibliotecas. A ideia aqui é conhecermos essas diferentes bibliotecas para saber que possibilidades elas nos trazem. E, então, saberemos como aproveitar o melhor de cada uma para atender às nossas necessidades. É como se fôssemos um grande chef de cozinha que mistura diferentes técnicas que aprendeu ao longo da vida para construir um belo prato. Ele não ficará restrito aos instrumentos e temperos que estão à sua frente, mas pensará além, para integrar diferentes técnicas, utensílios e temperos para alcançar algo novo, inovador e que atenda às suas necessidades. Aqui, a lógica é a mesma (ainda que não seja tão saborosa). É por isso que apresentamos diferentes bibliotecas e sua documentação; é importante que você conheça o que existe disponível para o seu uso, que consiga ter a autonomia de encontrar a informação de que precisa e o desejo de ir além, de descobrir coisas novas e experimentar diferentes funções e bibliotecas.

**TEMA 5 – RESOLVENDO PROBLEMAS COM UMA LINHA: USO DE EXPRESSÕES REGULARES EM PYTHON**

Agora que conversamos sobre **outras**bibliotecas, vamos retornar à biblioteca-padrão do Python. E, para isso, vamos conversar sobre uma ferramenta poderosa: as **expressões regulares**.

Para falarmos de expressões regulares (*regular expressions,*ou *regex*), vamos, primeiramente, pensar no seguinte desafio: vamos supor que a empresa na qual você trabalha precisa processar um arquivo contendo o equivalente a 500 páginas de texto, e lhe deram a tarefa de selecionar **somente** as placas de veículos.

Agora, o que seria uma placa válida? Até alguns anos atrás, uma placa válida possuía três letras e quatro dígitos. Logo, as seguintes placas seriam combinações válidas:

1. ABC1234.
2. ABC-1234.

Note que o segundo exemplo possui um hífen: existem pessoas que se referem às placas com o hífen e outras, sem. Agora, as seguintes combinações não são placas válidas:

1. 12345678.
2. ABC12.
3. 1234ABC.
4. A-B-C-1234.
5. ABCDE1234.
6. A1234BC.

Note que essas combinações não respeitam o **padrão**das placas que comentamos anteriormente: três letras seguidas de quatro dígitos. Mas, recentemente, essa regra mudou: também passaram a ser válidas placas cujo quinto caractere é uma letra (conhecido como *padrão Mercosul*), como:

1. ABC1A34.
2. ABC-1A34.

Logo, note que temos quatro **padrões**a que precisaríamos atender, pelo menos:

1. Três primeiros caracteres sendo letras e os quatro caracteres seguintes sendo dígitos (ABC1234).
2. Três primeiros caracteres sendo letras, seguidos de um hífen e, em seguida, os próximos quatro caracteres sendo dígitos (ABC-1234).
3. Três primeiros caracteres sendo letras; o quarto caractere sendo um dígito; o quinto sendo uma letra; e o sexto e sétimo sendo dígitos (ABC1A34);
4. Três primeiros caracteres sendo letras, seguidas de um hífen; o quarto caractere sendo um dígito; o quinto sendo uma letra; e o sexto e sétimo sendo dígitos (ABC-1A34).

Observe que realçamos a palavra ***padrão*** anteriormente: não queremos somente detectar a placa “ABC1234”, mas qualquer outra placa que siga esse exemplo (ZZZ9999, UNI0000, AAA0800 etc.). Isso dá mais de 170 milhões de combinações. Se considerarmos as placas do padrão Mercosul, temos quase 500 milhões de combinações a mais para encontrar.

E para colocar tudo isso em um algoritmo? Já imaginou ter que digitar todas as combinações possíveis? Ainda que tentássemos usar um loop (como o *for*) ou algo do gênero, teríamos que percorrer todas as alternativas possíveis, o que demandaria um considerável esforço nosso e do computador, para dar conta de todo o esforço.

É para isso que serve o regex: criamos uma string contendo **somente**o padrão que desejamos detectar (e, às vezes, o que **não queremos**detectar). Existem listas bem extensas sobre as possibilidades do regex, da Mozilla[[25]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn25" \o "), da Microsoft[[26]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn26" \o ") ou, ainda, do próprio Python[[27]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn27" \o "). Observe alguns exemplos:

Quadro 1 – Regex

|  |  |  |  |
| --- | --- | --- | --- |
| Regex | O que faz? | Exemplo | O que seleciona? |
| \* | Seleciona o elemento anterior nenhuma ou mais vezes | Caf\* | “Ca” em “Caça”  “Caf” em “Café”  “Caffff” em “Caffffé” |
| + | Seleciona o elemento anterior uma ou mais vezes | Caf+ | Nada em “Caça”  “Caf” em “Café”  “Caffff” em “Caffffé” |
| ? | Seleciona o elemento anterior nenhuma ou uma vez | Caf? | “Ca” em “Caça”  “Caf” em “Café”  “Caf” em “Caffffé” |
| {n} | Seleciona o elemento anterior n vezes | Caf{2} | Nada em “Caça”  “Caf” em “Café”  “Caff” em “Caffffé” |
| {n,m} | Seleciona o elemento anterior de n até m vezes | Caf{1,2} | Nada em “Caça”  “Caf” em “Café”  “Cafff” em “Caffffé” |
| [qualquer\_caractere] | Seleciona qualquer caractere que esteja em parênteses. “A-Z” selecionará todos os caracteres de A (maiúsculo) até Z (maiúsculo), por exemplo. | [a-z]+ | As duas letras “a” em “Caça”  “af” em “Café”  “affff” em “Caffffé” |

Imaginemos a frase “*The quick brown Fox jumps over the lazy dog. 1 or 1000 foxes, actually*.”. Veja, no quadro a seguir, quais padrões cada regex detecta na frase. Cada mudança de cor representa uma sequência que foi encontrada.

Quadro 2 – Utilização de regex

|  |  |
| --- | --- |
| Regex | O que seleciona? |
| [0-9] | The quick brown Fox jumps over the lazy dog. 1 or 1000 foxes, actually. |
| [0-9]+ | The quick brown Fox jumps over the lazy dog. 1 or 1000 foxes, actually. |
| [\d]+ | The quick brown Fox jumps over the lazy dog. 1 or 1000 foxes, actually. |
| [a-z]+ | The quick brown Fox jumps over the lazy dog. 1 or 1000 foxes, actually. |
| [A-Z]+ | The quick brown Fox jumps over the lazy dog. 1 or 1000 foxes, actually. |
| [A-Z0-9]+ | The quick brown Fox jumps over the lazy dog. 1 or 1000 foxes, actually. |
| [A-Za-z]+ | The quick brown Fox jumps over the lazy dog. 1 or 1000 foxes, actually. |
| [\w]+ | The quick brown Fox jumps over the lazy dog. 1 or 1000 foxes, actually. |
| [A-Za-z0-9]+ | The quick brown Fox jumps over the lazy dog. 1 or 1000 foxes, actually. |
| [A-Za-z0-9]+ | The quick brown Fox jumps over the lazy dog. 1 or 1000 foxes, actually. |
| [\w\d]+ | The quick brown Fox jumps over the lazy dog. 1 or 1000 foxes, actually. |
| \s | The quick brown Fox jumps over the lazy dog. 1 or 1000 foxes, actually. |
| [\w\d\s]+ | The quick brown Fox jumps over the lazy dog. 1 or 1000 foxes, actually. |
| [^\.]+ | The quick brown Fox jumps over the lazy dog. 1 or 1000 foxes, actually. |
| \w[^\.]+ | The quick brown Fox jumps over the lazy dog. 1 or 1000 foxes, actually. |

E aquele caso das placas de automóveis? Observemos a frase “3 placas devem ser selecionadas: UNI0100, NTE-0A00 e RRR9876. A JJ2070 é inválida.”.

Quadro 3 – Placas de automóveis

|  |  |
| --- | --- |
| Regex | O que seleciona? |
| [A-Za-z0-9]{7} | 3 placas devem ser selecionadas: UNI0100, NTE-0a00 e RRR9876. A JJ2070 é inválida. |
| [A-Za-z]{3}[0-9]+ | 3 placas devem ser selecionadas: UNI0100, NTE-0a00 e RRR9876. A JJ2070 é inválida. |
| [A-Za-z]{3}[0-9]+ | 3 placas devem ser selecionadas: UNI0100, NTE-0a00 e RRR9876. A JJ2070 é inválida. |
| [A-Za-z]{3}\d{4} | 3 placas devem ser selecionadas: UNI0100, NTE-0a00 e RRR9876. A JJ2070 é inválida. |
| [A-Za-z]{3}\-?\d[\dA-Za-z]\d{2} | 3 placas devem ser selecionadas: UNI0100, NTE-0a00 e RRR9876. A JJ2070 é inválida. |
| [A-Za-z]{3}\-?\d[A-Ja-j\d]\d{2} | 3 placas devem ser selecionadas: UNI0100, NTE-0a00 e RRR9876. A JJ2070 é inválida. |

Logo, o uso de regex é bem interessante para encontrar rapidamente múltiplos resultados dentro de um conjunto grande de textos. Ele possui algumas aplicações úteis, como:

1. Busca em textos por palavras-chave ou padrões.
2. Validação de dados (o CEP que o usuário digitou é realmente um CEP válido? O e-mail digitado realmente é um email?).
3. Web scraping (como a extração de somente os títulos das notícias em um portal de notícias).
4. Filtros em integrações (como a obtenção de somente os dados que precisamos a partir de uma integração construída com outro servidor ou com um banco de dados).

Em Python, o módulo a cargo do regex é o módulo re[[28]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn28" \o "). O código a seguir é um exemplo de implementação do regex de busca de placas que vimos anteriormente.

import re

# texto que usaremos na busca
texto_busca = '3 placas devem ser selecionadas: UNI0100, NTE-0a00 e RRR9876. A JJ2070 é inválida.'

# regex a ser usado
texto_regex = '[A-Za-z]{3}\-?\d[A-Ja-j\d]\d{2}'

# procurando
resultados = re.findall(texto_regex, texto_busca)
resultados



Caso tenha interesse, sugerimos que dê uma olhada nos links do módulo *re*do Python, bem como nos links sobre regex sobre os quais já comentamos. Saber usar regex pode fazer toda a diferença para resolver rapidamente problemas que seriam complexos no seu dia a dia. Caso também queira testar diferentes tipos de regex, existem sites que podem ajudar você a ver como funciona o regex de forma visual. Alguns exemplos incluem o regex101[[29]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn29" \o "), o RegExr[[30]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn30" \o ") e o Debuggex[[31]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a1&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnFnAdEzSBTSKCdVw1YpGirM=&ne=False" \l "_ftn31" \o ").

**Saiba mais**

Geralmente testamos um regex em algum desses sites para ver se a lógica funciona. Somente após garantirmos que o regex está selecionando o que gostaríamos (e não selecionando o que não gostaríamos) é que colocamos a lógica no código.

**FINALIZANDO**

Observe que a intenção desta aula foi mostrar a você a relevância do uso de outras bibliotecas em Python: na realidade, este é um dos motivos pelos quais o Python acaba tendo uma preferência por várias empresas, profissionais e estudantes. Isso naturalmente não significa que o Python é a **melhor**linguagem do mundo, cada linguagem de programação possui os seus prós e os seus contras. Em nosso contexto, a oportunidade de conhecermos diferentes bibliotecas é muito útil para sabermos como podemos **rapidamente**aplicar a linguagem de programação em novos desafios, algoritmos e soluções de mercado.

Nesse sentido, observe que comentamos sobre o uso de **expressões regulares**,**gráficos** e dados tabulares com o **NumPy**e**pandas**. Também comentamos sobre outras bibliotecas que podem igualmente ser úteis para o seu conhecimento. Independentemente da biblioteca, observe que também disponibilizamos links para você ter acesso à documentação em sua versão mais atualizada. Saber ler e compreender a documentação é certamente algo que lhe dará vantagem competitiva no mercado de trabalho. Logo, entenda as discussões que tivemos até o momento como direcionadores. Apontamos os diferentes pontos de partida que são do seu interesse, e não um passo a passo – até porque esse passo a passo muda a cada desafio no trabalho. Assim, é importante que você saiba como raciocinar para executar, e não como executar um passo a passo. Novamente: leia, explore e teste as documentações.

Aula 2

Linguagem de Programação

Inteligência artificial

Desmistificar conceitos

IA != mágica

IA não é a sucessora da Estatística

IA em Python

Definindo inteligência artificial e aprendizagem de máquina

Entendendo o mundo de Advanced Analytics

Advanced Analytics

Data Science

Statistics

Business Intelligence

Artificial Intelligence

GOFAI

Machine Learning

Artificial General Intelligence

Trabalhando com problemas de ciência de dados

Machine Learning

Aprendizagem supervisionada

Regressão

Prever a nota

Prever quantos trabalhos serão entregues nos próximos dias

Classificação Prever se será aprovado/reprovado

Aprendizagem não supervisionada

Seleção de atributos

Selecionar as informações mais relevantes

Redução da dimensionalidade

Agrupar informações

Aprendizagem por reforço

Otimização

Gaming

Metodologias

CRISP-DM / SEMMA / KDD / Agile / TDSP

Passo a passo

Definição do problema

O que quer fazer/resolver com ML?

Qual a dor da área de negócio?

Definição dos dados

Temos os dados para resolver a dor?

Essa base continuará disponível no futuro?

Preparação dos dados

Os dados possuem a qualidade necessária?

Que outras informações podemos extrair?

Desenvolvimento dos modelos

Que modelos/técnicas poderiam resolver o nosso problema?

Avaliação dos modelos

Os modelos são “bons”?

Usamos uma base de dados confiável para testar?

Disponibilização dos modelos

Como vão usar no dia a dia?

MLOps

Trabalhando com dados

Termos básicos

Dataset

Tabular / Texto / Imagem

Atributos (features/colunas)

Instâncias (rows/linhas)

Alvo (target/label/class/rótulo)

Análise de dados

Dados ausentes e zeros

Anomalias Linhas duplicadas

Dados categóricos

Dados numéricos

Dados de data e hora

Correlações espúrias

Transformação de dados

Tratamento de dados nulos

Bibliotecas

Featuretools

Pandas-profiling

Trabalhando com modelos

Algoritmos de ML

Aprendizagem supervisionada

Árvores de decisão (decision trees)

Máquinas de vetor de suporte (support vector machines, ou SVM)

Algoritmos gaussianos

Algoritmos bayesianos

Modelos lineares

Redes neurais

K-vizinhos mais próximos (K-nearest neighbors, ou KNN)

Comitês de algoritmos (ensembles)

Aprendizagem não supervisionada K-médias (K-means)

OPTICS

DBSCAN

PCA

Bibliotecas

Scikit-learn

TensorFlow

Keras

PyTorch

LightGBM

XGBoost

Statsmodels

Prophet

Divisão do dataset

Train/test split

Cross-validation (CV)

Rolling origin cross-validation (ROCV)

Trabalhando com resultados

Métricas

Classificação

Matriz de confusão Curva ROC (receiver operating characteristic)/área sob a curva (area under the curve/AUC)

Precision/recall

Regressão

MAE

MSE

RMSE

R2

MAPE

Seleção de atributos

Método do cotovelo (elbow method)

Olá Nesta aula vamos falar sobre o sol que eu te ajudo muito interesse pelo menos meu interesse espero que você goste que é inteligência artificial eu trabalho para oficialmente uma inteligência artificial e gostaria de mostrar um pouco sobre esse tema para você primeira coisa vale a pena se ficarmos alguns conceitos de Inteligência Artificial aquela coisa simples somente Inteligência Artificial É Mágica né se aperta o botão e aí isso já tem algoritmos que preveem alguma coisa né como se fosse uma Skynet alguma coisa assim e realmente não é o caso então eu te mostrar como que funciona a inteligência artificial na prática Que tipo de aplicações mas temos Além disso uma inteligência artificial Ela não é uma sucessora da estatística eu digo isso porque principalmente aves novas empresas novas desenvolvedores acabam deixando de lado um pouco tóxico da estatística como que somente o futuro fosse Inteligência Artificial e não é o caso temos muitas soluções que hoje não muito bem e que acabam e africanos soluções que são da área da estatística Censo é importante você não deixar de lado realmente Saara e acabar não entendendo né Inteligência Artificial se vê o futuro e que realmente tudo que nós acabamos aprendendo porventura em curso de matemática e estatística poderiam ser deixado de lado Justamente não é o caso e por último da ver como que funciona os sistemas de inteligência artificial em Python Tudo bem então dito isso vamos ver um pouco mais sobre esses conceitos

então agora para começarmos a falar sobre inteligência artificial importante como se fosse uma analogia imagina um jogo vamos pegar a bola tocar na bola no chão entendeu um pouco sobre o que está acontecendo né O que temos realmente como entendimento desse mundo da Inteligência Artificial então dito isso me acompanha nos slides quando nós falamos aqui sobre inteligência artificial eu queria voltar um passo e falar sobre Advanced Analytics Ou posso falar que como há nesse mundo aí nós temos na verdade diferentes conceitos né ou contextos quando trabalhamos nesse mundo então dentro desse mundo de Advance da Ana Nery nós temos de ir a ciência ou ciência de dados e dentro desse mundo de ciência de dados Aí se nós podemos ser diferentes técnicas ou Ares realmente do conhecimento em que nós aplicamos esses conceitos essas diferenças técnicas então primeiro em ciência de dados nós temos aí a área da estatística naturalmente a área da estatística envolve aí conhecimentos em matemática a improbabilidade né Aí é realmente uma área com pós fizer mais Hardcore nesse nesse contexto das exatas aí também nós temos o contexto não é o conhecimento de Business intelligence é o Bial falando assim bom português no mundo dos negócios praticamente Biá e sinônimo de dashboard então imagine aqueles gráficos tempo do t-cross ou poderíamos Conectar em outros momentos são gráficos de pizza relatórios números que as pessoas acabam monitorando no dia a dia bem Aquele visual aquele colorido Então isso acaba entrando muito no mundo do diário e por último mas também temos aí o mundo da Inteligência Artificial Note que esses três temas eles se conversam então aí eu posso ter muito bem soluções de ciência de dados que às vezes pega balão desce daí então eu tenho uma solução todo trabalho que nós fizemos para entender em cima dos dados o que que hoje tem na cerveja por exemplo tendências de vendas se tem algum problema em uma linha de produção em uma fábrica as temos por exemplo o problema de vendas Como é o mesmo comentei antes um problema na área da saúde com pacientes com problemas de manutenção preditiva Então temos aqui em diferentes cenários que podem ser aplicada em praticamente qualquer empresa e isso pode no final das contas combinaram em um algoritmo preditivo mas sim entender como as coisas são hoje na então entendeu aqui como que é o histórico a Que tipo de tendências nós temos que tipo de aprendizados nascemos e acabou para ir no mundo do diário às vezes nós também podemos ter soluções que elas são um pouco mais complexa e também pode acabar no mundo dessa crítica exemplo assim bem simples olha toda vez que chega numa sexta-feira às suas vendas aumentem 57 10% e é sempre aquela regra ou ainda né Nós em cima de uma base menor de dados podemos ter modelos econométricos na existe uma área que eu diria que estaria dentro da estatística que chama econometria em que a gente tem em modelos que realmente tenta entender o comportamento na área da economia né então todos esses movimentos econômicos tudo isso existe uma área que já estuda isso e como eu falei entra na área da estatística quando nós falamos de uma base gerados que ela é muito grande e acaba sendo um pouco complexos a parte de reconhecimento de padrões ela não é tão tão simples que pode ser explicado em uma equação a gente começa a se aproximar no mundo da Inteligência Artificial então Ando só de votar nos slides pense o seguinte quando falamos de ciência de dados não necessariamente Vamos acabar com algoritmo preditivo não não necessariamente nós vamos prever alguma coisa Não não necessariamente construiremos um algoritmo de Inteligência Artificial então não existe essa combinação e de diferentes técnicas não são somente essas né Essas aí são as mais as mais difundidas em que a nós poderemos sair acabar trabalhando no mundo de Analytics beleza entendi isso isso agora me acompanha nos sites mais uma vez veja que no mundo de Inteligência Artificial nós temos três subdivisões e isso é o que eu gostaria que fosse Claro para você não sei se você se lembra ou até mesmo se você era nascido nessa época no final aí da década de 90 foi de uma forma famosa aí apareceu em vários noticiários uma inteligência artificial que naquela época conseguiu ganhar de um grande mestre de xadrez não sei se você joga xadrez ou não mas no contexto aqui de xadrez temos que basicamente Inter Quais são as combinações que existem no futuro e diferentes jogadas tudo que o seu oponente para jogar YouTube tipos e também pode jogar Inter como que isso vai chegar em uma solução mas aqui essa inteligência artificial que foi

não tinha lá um computador muito potente para aquela época que testava todas as combinações possíveis escolher a melhor hoje em dia isso não é tão entendido como uma inteligência artificial Por que é um algoritmo que nós chamamos de força bruta esse gol ai ele é uma sigla que significa World fashion a boa e velha Inteligência Artificial e aqui na verdade né no nosso contexto Ou seja hoje em dia não vale tanto a pena nós trabalhar mais com isso porque como eu falei é Força Bruta né ele vai testar diferentes combinações ao infinito e hoje como você deve imaginar tem uns tantas combinações tantos fatores no dia a dia e para mudar completamente aí uma resposta então por isso que hoje em dia não faria tanto sentido essas técnicas logo eu gostaria de deixar claro aqui hoje em dia nas empresas quando nós falamos de inteligência artificial é praticamente um sinônimo de machine learning ou aprendizado de máquina são basicamente algoritmos que reconhecem padrões para tentar entender o que que vai acontecer no futuro a partir desses padrões no histórico sempre Toda vez que chove e também não existem guarda-chuvas à venda ou qualquer coisa do gênero ou a temperatura cai a quantidade de vendas de sorvetes também caem naturalmente no futuro ele vai tentar prever a mesma coisa nunca vai ter às vezes é necessariamente um número fixo que ele vai prever no futuro e vai tentar reconhecer padrões vai tentar generalizar os padrões para ir no futuro sim temos aí uma boa previsão é bem complexo isso dentro de machine learning existem diferentes técnicas e algumas mais do mundo da biologia alguns mais próximos do mundo da probabilidade outros que são mais próximos a realmente todo mundo da matemática e cabe a nós realmente escolher qual que é a melhor técnica aquele exemplo que seja né dos filmes Jarvis Skynet e outras né super inteligências como se você pedisse um comando ele já sabe o que você faz e serve para todos os cenários seria essa inteligência artificial geral esse último item que consegue ver nos slides essa super inteligência seja para o bem ou para o mal ainda não existe hoje basicamente toda agência que hoje Se nós queremos as vendas queremos prever a probabilidade de um funcionário sair ou se um funcionário Ele Pode Ele deve receber um aumento apresenta-se uma máquina dentro da empresa pode quebrar ou não para cada um desses cenários teremos um algoritmo de machine learning Pode ser que no futuro com essa inteligência artificial geral teremos um único algarismo que foi tudo isso mas esse não é o caso então como eu falei hoje temos pequenos algoritmos Ou nem tão pequenos assim algoritmos especialistas para entender cada um desses casos e para realmente prover é previsões beleza depois de ser apanhado é sobre todo o histórico de Inteligência Artificial dança do Analytics ciência de dados espera até que tenha sido um pouco mais claro para você gostaria de falar um pouco mais comigo trabalho realmente desse mundo desses dados Beleza tem dito isso por favor me acompanha nos slides quando nós falamos aqui do trabalho né de um problema de sensibilidade aqui quando eu falo um problema imagine algo que nós precisamos escrever então Aqueles exemplos que eu dei eu quero prevê uma venda de sorvetes ao longo do tempo é um problema eu quero prever a se uma pessoa ela vai ela pode pedir demissão ou não é outro problema eu quero ver se um cartão de crédito ele pode ser concedido ou não também é outro problema então problema é sempre algo que nós precisamos para ver se você já mexeu em um Excel ou alguma coisa do tipo imagina então que eu tenho várias colunas naquela planilha eu tenho falei várias colunas vai aparecer linhas tem muita informação lá e como você quisesse prever uma dessas colunas então quando nós começamos a falar de previsão de tentar entender alguma coisa as colunas então a gente começa aí se aproximar desse mundo aí de uma Shirley Beleza então assim dessas divisões nós temos primeiro aprendizagem supervisionada e aprendizagem não supervisionada quando nós falamos de aprendizagem supervisionada é quando basicamente temos uma base histórica em cima dessa base histórica nós queremos prever o resultado então imagine lá um histórico de jogo de basquete nós temos equipe seja a equipe que estava em casa na equipe adversária histórico dos jogadores como foram as últimas partidas e no final quem ganhou quem perdeu então olhando para todos histórico é se nós queremos prever por exemplo quem ganhou o quem perdeu e aí seria um grupo né ou é xvy é verdadeiro ou é falso ou é isso é aquilo quando nós estamos falando de categorias é o problema de classificação agora se nós queremos escrever um valor e quando fala o valor pode ser quantos pontos foram feitos naquela partida ou saindo aqueles isento do basquete na e pensando no seu próprio caso em cima do seu histórico eu quero prever a sua nota né Quando que você vai tirar de média final dessa própria disciplina CD aí um exemplo de regressão então quero ver um valor aqui quando fala o valor é uma escala numérica pode ser de 0 a 100 pode ser de 10 a 1000 pode ser and - 1 até mais um não importa então quando estamos falando de prever valores numéricos é um problema de regressão agora se eu quero prever algo que não é uma escala numérica Ou seja sim ou não verdadeiro ou falso e ABC quero descobrir o sabor de sorvete Favorito de uma pessoa então aqui é classificação então é importante primeiro deixar Clara essa grande diferença que temos entre regressão e classificação ambos E baixa de aprendizagem supervisionada vale a pena deixar muito claro aqui que para criarmos algoritmos de Inteligência Artificial e machine learning em aprendizagem supervisionada é importantíssimo é imprescindível termos uma base histórica então eu quero prever o sabor de sorvete Favorito de alguém eu preciso ter uma base histórica com todos os sabores favoritos de sorvete de uma grande quantidade de outras pessoas porque sem isso como que é algoritmo vai entender os padrões é como é que ele vai tentar se achar para gerar novas previsões então por isso que existe essa necessidade ou ainda pensando lá na regressão né quero prever uma nota quantos quilos para tirar no final dessa disciplina para chegar nessa televisão eu tenho que ter uma base Histórica de todas as outras pessoas e até a que passaram também por uma determinada disciplina para entender informações poderiam ajudar o país padrões nessas informações poderiam ajudar numa determinada a previsão de claro que para prevenir sindados futuros Eu também preciso ter Aquelas mesmas informações exemplo prático se eu quero ver uma nota em cima de fatores como idade quanto tempo quantas horas por semana uma pessoa estuda aquela disciplina se ela tem às vezes um conhecimento prático de mercado ou não se ela trabalho não porque é as horas que a pessoa está trabalhando Ela poderia não estar estudando então isso pode acabar afetando estão olhando todas essas informações e isso naturalmente podem influenciar em uma televisão agora imagina só chegou uma nova pessoa eu quero gerar uma uma predição da nota para essa pessoa se eu nunca votei informação das horas trabalhadas eu não posso consertar aquela informação para treinar modelo então tem que remover isso então acho que para gerar novas paredes Vamos trabalhar no futuro eu quero prever a água da ação do dólar do dia de amanhã pela presença cotação do dólar eu tenho que ter uma base histórica da cotação do dólar às vezes eu preciso todas as informações que ajudem naquela tomada de decisão porque se não tiver aquelas informações Muito provavelmente eu não preciso nem considerá-las para um treinamento beleza Além disso que nós temos aqui além dessa aprendizagem supervisionada que eu comentei temos aprendizagem Não supervisionada essa aprendizagem não supervisionada tenha que uma virada de chave então quando falamos de aprendizagem não supervisionada são casos em que eu não preciso necessariamente informar qual coluna eu quero te ver eu não quero mais escrever a nota não quero mais perder o sabor de sorvete eu não quero prever alguma coisa eu posso fazer tratamentos em cima da nossa base histórica imagina só eu quero ver como fazer uma nota a nota se eu tenho muitas informações que podem ajudar a prever isso informações que não está realmente ajudam por exemplo a cor favorita da pessoa é quantas calopsitas não é uma pessoa tem em casa eu posso ter às vezes humorista Favorito de uma determinada pessoa não necessariamente essas informações ajudam na tomada de decisão então também temos informações temos algoritmos e eles podem ajudar a reduzir essa base de dados para realmente selecionarmos apenas os dados que importam então tem o algoritmo de seleção de atributos foram falou tributos são as características traduz selecionar somente as colunas e as características que importam para nós então existe a origem de seleção de atributos que fazem isso E além disso tem algarismos de redução da dimensionalidade Então imagina só eu tenho aqui como diferença de seleção de atributos redução da dimensionalidade imagina o seguinte E para isso vamos voltar naquele exemplo daquele exemplo das notas às vezes eu posso ser uma informação de quantidade de horas que a pessoa se dedica aos estudos na segunda-feira quantidade de horas que a pessoa cirúrgica Santos na terça-feira na quarta-feira já pegou a ideia quando estou falando de seleção de atributos eu vou escolher qual dessas colunas importa para nós às vezes é somente a quantidade de óleo que a pessoa se dedica no sábado e no domingo quando eu falo redução de dimensionalidade agrupamento eu posso falar sim estou pegando Todas aquelas informações quantidade de horas dedicadas de segunda terça quarta quinta e sexta eu junto novamente junto todas elas em uma coluna só no caso poderia ser quantidade de horas dedicadas ao estudo durante a semana e a mesma coisa de domingo quantidade de horas dedicadas no final de semana Então veja que eu não perco muitas informações estado somente aglutininas né somente município elas dependendo da técnica ficado aqui ou outro pode ser mais mas é desejável pelas pessoas que vão consumir aqueles algoritmos e dependendo do Poder de explicação as vezes que mais utilizam aquele algoritmo precisa saber quantas horas a pessoa se dedicou no sábado de uma forma separada do Domingo então Muito provavelmente esse algoritmo de redução de dimensionalidade não seria muito útil e sim um de seleção de atributos Então já existem e se caso sair e por último mas não menos importante temos também os algoritmos de aprendizagem por reforço temos aqui por exemplo utilização eu tenho uma quantidade de matéria prima eu tenho um conjunto de materiais de produtos que eu posso construir uma madeira com cadeiras mesas móveis em geral e eu preciso maximizar o meu lucro qualquer combinação melhor que eu posso fazer com que eu tenho irmão de matéria-prima para atingir esse objetivo é um algoritmo de otimização qual que é o caminho mais rápido daqui de onde eu estou nesse momento até por exemplo a casa de um amigo então aqui também é o outro algoritmo de otimização quero a rota que seja mais curta e mais rápido ao mesmo tempo então por isso que aí nós temos outros algoritmos e finalmente temos aplicações no mundo dos jogos é também para encontrar rotas ou por exemplo para desenvolver né modelos de Inteligência Artificial robôs que possam jogar em conjunto contigo então aqui também teremos outra aplicação de aprendizagem por reforço Ah também importante saber que existem algumas metodologias quando falamos metodologias e formas de trabalho mesmo é muito provável que em outras oportunidades você veja principalmente metodologia de trabalho de software de projeto nem software utilizando wall-e mas não sabem que são essas metodologias quando falamos de ciência de dados existem outras formas crisp-dm mas também existem outras metodologias como você no Cadê meu próprio Agir é metodologia ágil e o PSP como que funciona essas metodologias de uma forma geral tá aqui primeiro eu preciso definir o problema o problema como nós falamos anteriormente a saber oque O que que a gente precisa a trabalhar com Ashley lane para quê que nós estamos trabalhando em machine learning na O que que a gente precisa resolver lembre mais uma vez que quando nós falamos de desenvolver software desenvolver algoritmos de machine learning nós não trabalhamos para desenvolver algoritmos mas sim para resolver algum problema né para desenvolver uma solução então é justamente importante entendermos Qual que é a dor qualquer necessidade da rede negócio até realmente para entender o porquê que nós estaríamos dedicando nosso tempo nosso esforço nosso conhecimento para desenvolver em um algoritmo depois de entender esse problema importante definirmos os dados ou seja oque querem prever de algum uma forma quais serão as vendas dia nas próximas duas semanas beleza querem fazer essa previsão dia a dia temos hoje uma base sólida que é também de agir Se hoje eu tenho uma base teórica que somente tem os dados mês a mês já não serve eu não posso resolver esse problema com a base histórica ou agência da principal Helena ou pega outra base de dados ao quero prever sabor de sorvete Favorito de todas as pessoas que estão na região Tá mas tem uma base histórica um sabor de sorvete Favorito de pessoas se não tiver não tem como começar né não tem como inventar dados para resolver algum problema então por isso é importante entender que dados nós temos hoje se esse dados eles servem né E também vão continuar sendo ter no futuro para nós se temos essa base de dados disponível aí que começa também sou trabalho esse processo de preparação de dados facilmente aí demora 80% do tempo porque realmente entender a base de dados Popular ela trabalhar com ela criado novas colunas pega por exemplo informação da data a partir de uma coluna que tem uma data eu posso criar uma coluna que se chama ano outra que é mês dia dia da semana Qual que é a semana do ano informações ou até mesmo se é feriado ou não então a gente começa a criar novas colunas e ajudam a resolver o problema então o trabalho de cientista de dados é também e tentar trabalhar em cima dessa base entender quais informações melhor auxiliam modelo a reconhecesse padrões então por isso que entra nesse trabalho de preparação dos dados aí naturalmente né quando nós estamos falando do dia a dia de um trabalho quase nunca mais gerados ela vem perfeito Então vai ter dados nos teriam dado que as pessoas digitarão errado contém dados que são assim muito fora do comum a cotação do dólar estava em 55.25.66 terminado em 6 milhões alguém digitou errado aquilo aquele valor tá certo mesmo então tem que ter por uma preocupação em arrumar sua base de dados e deixar ela bem bonitinha para ir sim nós conseguimos desenvolver os modelos em importantes porque daí existe um termo em esses dados chamado garbage in garbage out ou seja se você alimenta lixo você também vai ter lixo como retorno desse modelo de Inteligência Artificial então por isso que existe essa preocupação em preparar uma base de dados e garantir que essa base de dados aí se ela vai ser a muito bem preparada para não termos esses lixos né que eu comentei que realmente é esse o termo que acabam utilizando no trabalho então por último nós desenvolvemos a esse modelos quando falo de desenvolvimento de modelos imagina só é um problema de aprendizagem supervisionada é um caso de aprendizagem não supervisionada é classificação é regressão dentro de classificação na Existem várias bibliotecas vários algoritmos que pode podem ajudar a resolver esse problema então aqui realmente é um passo de entender qual seria a melhor técnica o algoritmo para resolver esse cenário aqui vale um parente se quando falamos de Inteligência Artificial Python hoje em dia é linguagem padrão para esse tipo de desenvolvimento não é a única na existem aí outras linguagens que nos auxiliam nesse aumente o padrão é o Python e existem a bibliotecas como SAC Light e várias outras que são utilizadas para esse desenvolvimento depois que você treinar o modelo você avalia eles dá para garantir esse modelo se aumente estão funcionando corretamente essa gente tem uma base de dados confiáveis para testar até porque quando treinamos esses modelos Ok Google algoritmo de teve acesso uma base de dados mas para validar a gente tem que estar com novos dados precisamos garantir que o algoritmo realmente vai funcionar bem com novos casos tem por isso que precisamos ter uma base de treinamento que o algoritmo só ler validar para garantir que é algoritmo esteja funcionando adequadamente daí depois disso se avaliação foi tudo ok né as métricas elas estão de acordo com o esperado que nós podemos disponibilizar para que as pessoas utilizem no dia a dia até existe um termo chamado Heliópolis é como se fosse baseado no de votos que é realmente uma metodologia é uma forma de trabalhar descongestão com desenvolvimento de modelos quando operamos eles no dia a dia finalmente eu gostaria de mostrar para você essa imagem Então esse é um exemplo do crisp-dm Então os pais de ver primeira vez tem muitas flechinhas ela tem muito vai vem Isso realmente é muito comum de um trabalho de ciência de dados deve-se que lá no começo eu comentei que precisamos definir o problema que é bem essa fase do bizness Lander Tenere a entender as dores e depois entendeu se temos realmente dados para para resolver aquilo se não temos dados precisamos montar um passo então da mesma forma dele por ele chama estou lá preparando uma base de dados vou modelar criei uma dela preditivo não ficou muito legal eu volto um passa arruma de novo os dados Vou treinar deu errado de novo volto Passa então tem realmente muito vai e volta nesse processo e é por isso que ele realmente não se encaixam muito bem no trabalho de ciência de dados em uma metodologia Waterfall bem aquela padrão né que é trabalhada aí por várias décadas ou até mesmo em uma metodologia ágil Eu já vi muitos casos em que por exemplo eu posso gastar muito tempo no Rio de Janeiro terno Anders l e que eu tô falando semanas meses e depois o modo nem demora horas ou vice-versa eu posso ficar pouquíssimo tempo de Daniel stern eu já tem uma base de dados mas eu preciso ficar muito tempo lá no módulo até achar um melhor resultado então por isso que realmente é um trabalho de descoberta e desenvolvimento até realmente encontrar a melhor solução beleza Observe que muita preocupação que nós tivemos até o momento foi realmente sobre os lados chegar em que bases de dados ela realmente ela é roubou suficiente para trabalharmos com criação de algoritmos uma boa análise de dados Um Bom desempenho em um trabalho de ciências de dados então é importante também esclarecemos alguns termos que são bem comuns aqui na sua natureza de um trabalho de ciências gerados Beleza então dito isso por favor me acompanha nos slides alguns temas básicos que eu gostaria que ficasse uma história para você alguns desses termos eles estão em inglês e também português mas assim não estou tentando não aportuguesar neta dos termos mas realmente entender quais são alguns dos nomes em inglês que ajudam muito encontrado aumentação internet a saber como resolver um problema até mesmo encontrar novas bibliotecas Então realmente é no na intenção de garantir que você consiga ter uma autonomia aí em um trabalho em relação aos dados tá primeiro ponto quando falamos de base de dados essa base de dados ou de 7 ele é um conjunto de dados é uma base histórica aí que nós temos de informação essa base histórica ela pode ser de alguns tipos os mais comuns são uma base que é tabular então quando eu falo tabular imagina uma tabela em uma planilha de excel um conjunto de dados em que basicamente temos valores a temos aí uma sequência de informações para poder trabalhar essa sequência de informações ela quase sempre ela é dividida entre colunas ou atributos ou fixo da então basicamente são eles se referem a mesma coisa então são características diferentes que existem dentro daquela base de dados aqui imagina de novo aquele exemplo de uma previsão de notas um exemplo de atributo seria a idade de uma pessoa outro exemplo de atributo seria quanto tempo que aquela pessoa se dedica aos estudos outro exemplo de atributo seria a cidade onde ela mora Então temos aqui alguns exemplos de atributos agora que seria uma Instância uma Instância ou Dinha são exemplos nação amostras que fazem parte dessa base de dados uma pessoa com todas as seis exemplos das notas é uma Instância na ocupa uma linha ela totalmente diferente e Independente de outra pessoa que também está naquela base de dados então se eu tenho nessa base de dados das notas 5000 Estância significa que são 5.000 pessoas que fazem parte daquela base de dados ou pegando por exemplo uma base histórica da cotação do dólar diária se eu tenho nessa cotação aí nessa base de dados 365 instâncias significa aí nós temos basicamente 365 dias 365 amostras então aqui é importante né só deixar isso claro para você e Finalmente né eu tenho principalmente em aprendizagem supervisionada um alvo em alguma coisa que eu preciso prever então Mas qual que é o atributo que eu preciso prever eu quero prever a nota Então essa nota eu quero prever o nosso tarde o Label Class né E aí a gente vai ter a diferentes nomenclaturas e syllable e o que eles são mais específicos ao mundo da classificação mas é a coluna que nós queremos escrever então é importante nessa saber qual que é esse rótulo ou fica a gente precisa encontrar com o uso de aprendizagem supervisionada se estivermos utilizando isso mas assim ele acerte que é tabular ele sempre terá e atributos e instâncias também nós podemos fazer bases de dados que são do tipo texto Então documentos mesmo né imagina artigos para serem Diário Oficial é o passeio diferentes documentos de fato que incluem texto em parágrafos e assim diferentes idiomas e também podemos trabalhar com inteligência artificial eu quero por exemplo a partir de novos textos na reconhecer todas as ocorrências que tem o nome de uma empresa ocorrências que tem lugares Eu preciso de uma base sólida para fazer o treinamento então também temos aí bases que são do tipo texto e também temos bases de dados do tipo imagem então posso ter por exemplo há várias fotos de lugares e com isso a tentar reconhecer o tipo de lugar em novos locais vamos supor por exemplo que você gosta de muito de pássaros Você pode tirar várias fotos de três tipos de Pássaros Pássaros diferentes e com isso com novas fotos de ter um algoritmo e consegue prever Tem um aqui já seria um caso de classificação prever qual seria o tipo do pássaro de novas imagens que você acaba fornecendo Impressionismo Então aí é um exemplo importante é que os poderia aplicar outro caso que ele é bem bem estudado reconhecimento de dígitos manuscritos então várias pessoas escreveram na com algoritmos que a partir desses números escritos reconhecem Qual que é o dígito 9 com 16 O que é 15 que a 15 de 18 14 19 e aí com isso para novos dígitos inscritos ele também consegue reconhecer o que seria aquilo que foi escrito então só para você saber né que existem aí alguns tipos diferentes de datas certas não são somente esses nem eu posso ter por exemplo som eu passei outros tipos que também possam ser trabalhados aqui em vigência artificial Além disso quando nós falamos de análise de dados eu posso ser muito bem dados que são ausentes e zeros Aqui tem uma diferença tá muitas vezes eu posso ter dados que tem vários zeros e alguns desses dados em São zeros e outros dados alguém lá colocou como valor padrão 0 Imagina assim o seguinte exemplo eu tenho uma base histórica da quantidade de clientes que entraram em uma loja ou Supermercado essa base histórica da quantidade de pessoas que entrarem Supermercado eu tenho histórico de 5 anos esse supermercado a gente sabe que ele fica aberto de domingo ou domingo ou seja não fecha quase nenhum dia a não ser por exemplo um dia de oferta Depois tem um jeito de oferta no dia de fechamento alguma coisa assim de balanço que aí sim aquele Supermercado fecha quando você analisa a base de dados você vê que tem muito mais zeros Ou seja quando ninguém acabou entrando naquele Supermercado do que deveria então aqui pode ser muito um caso de existir andados em que realmente ninguém entrou naquela loja E aí sim era 10 legítimo e também dados em que ano passado não tinha informação uma piada daquele dia alguém preencheu zero por padrão então importante saber diferenciar que porque realmente podemos ter esses dois diferentes casos tão 10 não necessariamente é 10 ainda que isso possa soar um pouco confuso Além disso nós podemos ter numa lista lembra aquele exemplos que eu odeio a cotação do dólar na do 55.26 e do nada 6 milhões 7 milhões essa anomalia pode ser realmente o valor que era muito assim muito abaixo do comum mas é um valor legítimo ou realmente era um valor aqui Alguém esqueceu de colocar uma vírgula alguém digitou um pouco mais do que deveria e por isso tá aquele valor é bem acima do limite então isso também a gente pode tratar se necessário for também podemos ser linhas duplicadas Então imagina de novo aquela base histórica das notas eu posso ter às vezes um aluno que tá a 5 10 vezes a mesma base eu consigo remover suas colônias o picadas porque senão vou acabar induzindo algoritmo que aprenda muitas vezes em cima desses casos repetidos Então posso acabar ensinando errado e não é isso que a gente quer em algoritmo de machine learning queremos mostrar diferentes casos legítimos entre si para que ele consiga entender corretamente esses padrões então por isso que é nós queremos remover as espinhas oq Caldas até porque um treinamento algoritmo que vai dar o mesmo peso para diferentes fins então não quero acabar dando um peso a mais não é para um caso só porque ele está repetido na nossa base de dados e finalmente aqui nascemos dados categóricos então uma categoria pode ser um sim ou não pode ser uma cidade pode ser um estado posso ter que diferentes informações que são na prática categorias que eu preciso tratar até porque grande parte desses algoritmos ele somente trabalham com números então uma cidade Curitiba São Paulo Rio de Janeiro para ser essas diferenças e daí se eu não posso na grande maioria dos casos ter cidades em texto eu preciso converter São Paulo vira código zero Rio de Janeiro ver a código 1 Curitiba a vida código 2 e assim por diante até realmente consegui vamos tratar adequadamente esse essas categorias não somente existe essa técnica que eu falei nada de converter para código 01 e 2 eu posso criar nossas colunas eu posso atribuir um peso na tomada de decisão dessas dessas diferentes categorias mais importante você saber né que existem diferentes formas de tratar mas esse dados categóricos beleza Além disso não Eu posso ser diferentes ainda tipos de dados além daqueles dados numéricos que são mais comuns aparecer data e hora e aí se eu posso destrinchar nesses dados entre dia mês ano semestre semana do ano e dias da semana a mesma coisa com h eu também posso Preciso né tomar cuidado com correlações que são escuras ou seja toda vez que é exemplos práticos exemplos práticos econômicos na existência de sites que dão algumas desses exemplos que seriam Pedros até que seria assim a quantidade de filmes que determinado autor de quantidade de mortes por afogamento com queijo no Estados Unidos não tem nada a ver uma coisa com a outra mas ainda assim se você vai colocar um gráfico as duas coisas elas possuem uma correlação altíssima Então não é isso a gente quer também que é algoritmo acabe aprendendo sem querer né que uma coisa acaba influenciando outra então por isso que também eu trabalho de análise de dados é garante que essas relações elas na prática faz sentido Aí sim para gerarmos boas flexões também tem um pássaro que pode ser importante de transformação de dados como falei exemplo prático né dados de data eu tenho lá no mês de junho eu posso criar novas colunas eu posso mudar escala delas fizeram sem vamos dar uma escala de 0 até o Uber tem alguma chance funciona o melhor nessa escala então não existem aí passos para isso e também tratamento de dados nulos ou seja tem uma coluna da quantidade de horas que alguém estuda na quarta-feira e somente 70% das pessoas responderam aquela informação eu posso pegar aquelas 30 por aquele 30% restante e utilizar algoritmos para preencher com valores em cima de padrões Então olha todo mundo que mora em São Paulo na média na quarta-feira estuda 2 horas tenho tenho que uma pessoa que não informou nada e ela também é de São Paulo Então vou atribuído as horas para ela eu posso fazer isso eu também colocar um valor constante lá para o padrão tem não respondeu é uma hora é 025 isso também depende de cada problema então eu posso utilizar uma técnica que ela mais avançada nesse sentido de entender padrões e aí sim preencher a esses dados não usou o que é um valor constante e acabou também da importante saber que existem algumas bibliotecas para automatizar esse trabalho para nós tanto de análise como age a criação de realmente atributos Então tem um filho Porto é uma lotérica em Python né ela nos ajuda bastante nesse trabalho de criar novos atributos não é de tentar relacionar entre eles e aí sim criar novos entender nossa base de dados um pouco mais encorpada né criando realmente novos atributos e também tem um panda super Fire super Fire ele vai pegar um gato ferido plantas e fazer várias análises é correlação entender dados números onde que são estados nos Quantos são que valores extremos paster fica categoria O que que não é então isso pode também nos auxilia bastante no nosso trabalho Beleza então aqui não foi uma passada geral sobre análise de dados eu espero que todos os tópicos possam dar mais caros para você lembra que eu comentei que existem diferentes técnicas para realmente criar em cima dela os primitivos é esse que nós veremos um pouco mais a partir de agora vamos lá no slides então quando eu te falamos de algoritmos de machine learning lembre-se que que é uma subdivisão né Tem uma aprendizagem supervisionada e eu poderia ser classificação regressão e dentro dela e nós temos também suas subdivisões a que são alguns tipos de técnicas para criar um algoritmo de classificação ou criar um algoritmo de regressão árvores de decisão essas árvores de decisão é imagina se eu tenho uma raiz a imagine lá o próprio caso das notas de uma nota essa pessoa posta um Norte a uma base de tudo que começa assim essa pessoa ela estuda mais de 10 horas na semana sim ou não se for não ela vai em uma rota se for Sim ela vai outra aí vem uma segunda pergunta ela respondeu que sim beleza então estuda mais de 10 horas ela trabalha olha hoje em dia a trabalha mais de 20 horas ou menos de 20 horas se for sim vai para o lado se for não vai para o outro e vai fazendo vagas em várias perguntas ou decisões e por isso que tem esse nome já diz de decisão até chegar ao resultado final a beleza como você estuda mais de 3 horas por semana como você mora em uma capital como você trabalha mais de 20 horas por semana e vários outros fatores a sua nota final nós entendemos que será o 8 ou 9 então existem nessas decisões que conseguem chegar a um determinado valor também temos máquinas de Vetor de suporte USB M são as máquinas são basicamente vetores.na então imagine diferentes linhas que são cortadas em um gráfico até realmente de qualquer divisão entre diferentes atributos e instâncias até chegar ao resultado final Então pega lá uma máquina de vetores de suporte Qual que é a divisão a fronteira olhando para nossa base histórica que vai determinar se uma pessoa vai gostar de sorvete de morango ou não vai gostar de sorvete de morango então essas máquinas elas ajudam a nesse sentido então é outro tipo de técnica temos também algarismos gaussianos temos algoritmos de 15 anos que aí sim era um mundo mais da probabilidade na estatística temos modelos lineares e praticamente ter uma equação e para chegar em um determinado caso temos as redes neurais redes neurais essas que são em tópicos super quente tem um pessoal fala de depilar me fala de tensorflow fala Esquece essas redes neurais dela são muito mais complexas do que uma delinear por exemplo mas imagina se diz orais como sendo um canhão para matar uma formiga um canhão às vezes é desnecessário para matar uma formiga e um caminhão ele pode errar formiga então por isso que a sociedade neurais elas são como eu falei técnicas muito complexas mas não imagina que elas são a solução para tudo eu posso utilizar esse modelo linear comentei uma svm alguma outra coisa para chegar em um melhor resultado tem Léo KNN também que ele vai por proximidade a ver essa pessoa eu quero determinar nota dela eu vi que um perfil bem similar ela outras aí eu encontrei outras cinco pessoas e essas outras cinco pessoas elas tiraram seria uma nota muito boa então naturalmente eu também vou atribuir vou prever uma nota muito boa para essa nova pessoa para esse novo caso que eu estou analisando então cai nele ele funciona dessa forma aqui também e também por último temos comitês de algoritmos então Imagine que eu posso juntar esses diferentes técnicas até chegar realmente aí tem um melhor resultado É como se eu pegasse o melhor de cada um desses algoritmos juntasse todos eles para a se enxergarem uma decisão eu diria que no dia a dia vale a pena testar diferentes técnicas Isso depende muito de cada caso depende muito de cada problema e não vai ter uma resposta simples então eu não posso afirmar que que por exemplo olha para esse tipo de caso de professor deixo você tem que usar o carne que vai ser o melhor ou para prever a nota de alguém se tem que usar redes neurais Isso depende muito de cada caso até às vezes para prever anota aí de alguém de um lugar Pode ser um algoritmo uma técnica muito diferente do mesmo problema que é prever nota e outro lugar ou não é o sabor de sorvete assim por diante então existem aqui diferentes técnicas que nós precisamos invariavelmente de Sá até chegar e o melhor caso exemplo prático do que aconteceu não e ciência de dados mas na ciência como todo criação de vacinas dão para criar a vacina foi necessário né testar diferentes técnicas até realmente encontrar o que resolvesse O problema não é somente uma técnica que resolveu Aquilo é cada um realizou uma forma diferente para chegar uma solução então aqui é a mesma coisa as vezes você pode ter um estado muito bom uma rede neural e o e não há nenhum problema nisso também para aprendizagem não supervisionada existem outras técnicas eu passei o caminhos para ser o o pix Posso ter o pessoal de esquerda e todas elas e mexendo tentar agrupar as informações até eu falei encontrar clusters perfis grupos Entre esses diferentes casos quando eu falo casos aqui em Gene que eu posso agrupar tanto atributos quanto instâncias eu posso pegar dessa base de alunos e tentar descobrir quais são os perfis de alunos que existem para descobrir perfil eu tenho que ter uma base que me ajude a entender Qual que é o tipo de perfil se nesse perfil se nessa base de dados eu tenho uma quantidade de calopsitas que os alunos têm a quantidade de cachorro se as pessoas têm a quantidade de lagartos né que as pessoas têm outros animais eu vou descobrir perfil de preferência de animais sendo batendo nada a ver com nós aqui agora se eu tenho a informações somente denota quantidade de horas estudadas assim por diante eu vou ter Perfil completo diferença porque esses perfis foram calculados em cima dos dados que nós temos em nossa base de dados então não acha que que existe nos virarmos e superfícies é importante entender o que nós estamos provendo de informações para esses algoritmos também gostaria de mostrar para você algumas bibliotecas saber né daqui caminhos existem a mais conhecida de todas as mais utilizadas de todos é o seguinte lá então ele esse já tem várias aquelas técnicas que nós vimos até agora tem o também tensorflow que horas que o cartório essas três aqui na tensorflow que era o pai do Norte os três eles trabalham com redes neurais então para rodar Eliseu invariavelmente preciso de um computador um pouco é mais poderoso é para dar conta desse algoritmos Mas também tenho outros algoritmos bem mais leves né outras bibliotecas mais leves light.br morte de buscar Eles trabalham com várias árvores de decisão se criasse centenas delas ao mesmo tempo e aí se enxergasse uma decisão eu tenho status móveis que é mais rápido da estatística Mas também eu posso utilizar ele para criação de algoritmos eu tenho uma técnica madeira Teca muito boa para prever comportamentos ao longo do tempo ele é do próprio Facebook então ele foi desenvolvido até para prever o comportamento das pessoas navegando no sites e isso também pode ser utilizado para problemas de séries temporais ou seja que apresenta um comportamento ao longo do tempo é um caso de agressão que também nós podemos utilizar aqui no caso com o profeta por último só um ponto que é importante para você saber quando nós vamos tentar um algoritmo em portante dessa divisão entre uma base de dados que ela é de Treinamento ou seja separar 70% 75 80% da base de dados para o algoritmo na para entender que foi utilizado para ir reconhecer os padrões e o restante a 20% e 30% restante validar o algoritmo é um quando falamos de divisão dos dados 7 é para isso o site chover nele já tem um método ele tem uma função que ele trata disso entendo já faz essa divisão para nós se informar você é uma data certa para ele outra forma da gente também trabalhar com validação cruzada então ele consegue pegar aquele j7 subdividido em pequenas partes e em quanto ele faz o treinamento com quase todos os países eles deixam muito fora e valida com ela signo aquela mesma subdivisão ele escolhe outro pedacinho para validação E treina com todo o resto e assim por diante até cobrir todo 7 e assim nós também conseguimos validar o nosso a nossa base de dados prata Lemos de séries temporais é um pouquinho diferente esse rolo em Origin ele faz essa divisão ao longo do tempo tenho como se fizer assim eu vou prever a cotação do dólar até o ano e vamos supor que estamos em dezembro de um determinado ano eu posso fazer uma validação cruzada que vai de Janeiro até Fevereiro desculpa Janeiro Até abril e aí eu válido com Maio depois fevereiro até maio válido com Juninho Março até Junho válido com Júlia e assim por diante até chegar em dezembro então aqui também o outro exemplo de validação cruzada mas mas mais especificamente para problemas de séries temporais Beleza então enfim o que são os tópicos relacionados mais trabalho quando ele acessa um técnicas que acreditam que possam ser muito úteis para você o seguinte cenário você é um segurança de aeroporto então a sua missão é basicamente garantir que as pessoas com boa intenção passei por você e as pessoas mal-intencionadas que sejam filtrados por você então você não quer que as pessoas mais tensionadas passei não é por você e continue suas faltas dentro do aeroporto agora imagine o seguinte proporcionalmente quantas pessoas mal-intencionadas estão dentro de ser o porto Talvez seja um por cento e meio por cento até - Vamos pensar a quantidade de pessoas eu posso ter às vezes em um determinado momento 999 pessoas que são bem intencionadas e uma pessoa ela não tem umas intenções muito legais você quer filtrar somente a única pessoa que se intencional mas se ela passasse e causar um grande estrago para todo mundo então você não quer que isso aconteça o que é aquela pessoa como você deve imaginar uma tarefa muito difícil até por isso ele 01 quartos acaba selecionando muito mais pessoas proporcionalmente né você já temos aí a falso positivo sendo selecionados mas justamente para tentar garantir que aquela pessoa que realmente ela não é muito bem intencionada que ela acabe também sendo filtrada agora imagine o seguinte cenário dessas desse total de 1000 pessoas novamente 999 são bem intencionados de uma delas não é você acaba se relaxando um pouco vai almoçar tomar um café e deixa as mil passarem acaba acontecendo um grande incidente por conta daquela pessoa mãe tensionada que você deixou passar e aí quando eu falo para você fala assim não olha o meu processo aqui meu procedimento tem uma curasse de 99.9 por cento dos casos deixou é realmente passar 999 pessoas bem-intencionadas mais daquela uma pessoa você deixou passar significa que seu processo está legal você foi almoçar você deixou passar todo mundo tá legal isso muito provavelmente Costa É Não então por isso a gente precisa tomar muito cuidado com o termo acurácia me acompanha aqui nos slides por favor quando nós falamos de acurácia Existem algumas métricas nesse sentido então passei médicas que são de classificação ou direção justamente para garantir que aquela uma pessoa não é aqueles casos bem específicos também sejam capturados pelas métricas e não falar que tem 99% de álcool traz exemplos clássicos de métricas que a gente pode utilizar que justamente para filtrar os casos temos a matriz de confusão é uma matriz de confusão um gráfico que ajuda bastante para entender esses casos e justamente entender como é que ele está se comportando para diferentes oportunidades temos também essa curva roc o área né sobre essa mesma curva roc e também ajuda nessa mesma intenção e por último o PSG ou Recall que também é oferecem esse mesmo cenário seja entendeu com bem que o Coritiba está funcionando tanto para casos bem intencionadas ou mal-intencionadas nesse exemplo do aeroporto também para regressão temos outras médicas que elas podem ajudar geralmente na em casos mais simples falando é quadrado ou R2 + R2 ele não é tão legal e em casos no dia a dia né de trabalho Principalmente nesse contexto caótico comportamento de mercado de pessoas em que a gente tem padrões bem diferentes de comportamento por outro lado existem outras métricas que podem ser mais úteis por exemplo o My Oh m s e r m c e o meio e cada um deles se existe uma equação que relaciona essas diferenças técnicas e cada um tem seus prós e os contras um por exemplo ele vai ser mais sensível a valores a erros muito altos outro vai ser sensível e os percentuais muito altos Então depende muito de qual que é o seu caso qual que é o seu problema para escolher qual que é melhor técnica que poderia ser utilizada e Finalmente nós temos pela seleção de atributos o método do cotovelo ele basicamente permite entender quais atributos Qual que é o limite de tribos que a gente pode selecionar Eu tenho 20 atributos tem 130 ao todo Qual que é o limite é preciso selecionar 5 6 7 Então esse método de nos ajuda nesse sentido especialmente utilizado em uma das técnicas de seleção de atributos registração de atributos e seria o endereço então rapidamente aquilo mostrei para você né algumas dessas métricas que podem ser utilizados no dia a dia no mercado de trabalho para garantir que este fato não vai ser uma accuracy de 99 Quantos por cento não e sim vai ter em uma porcentagem de um valor e seja muito mais próximo do que você necessita

**CONVERSA INICIAL**

Olá! Anteriormente, comentamos sobre o uso das bibliotecas e, ainda, a importância de lermos a documentação das bibliotecas. Aqui, vale reforçar: o objetivo é o de mostrar *o***caminho**para você, mas é impossível mostrar tudo que existe pela velocidade de mudança. Apostamos que, nas próximas 24h, pelo menos algumas **centenas**de bibliotecas em Python mudarão alguma coisa. Elas poderão ter novas funcionalidades, mudanças em funcionalidades já existentes, correções na documentação, novos bugs, bugs resolvidos, entre outros. Por outro lado, a**base** não muda: a forma de ler, procurar e interpretar a documentação que aprendemos aqui lhe ajudarão independentemente das possíveis mudanças no futuro.

Bom, e por que relembramos isso agora? Uma das áreas da ciência em maior crescimento e velocidade é a da Inteligência Artificial. E, naturalmente, existem bibliotecas em Python para Inteligência Artificial (IA) – de fato, o Python já é entendido atualmente como a linguagem padrão para o desenvolvimento de algoritmos de IA. Logo, que tal criarmos uma IA juntos?

**TEMA 1 – DEFININDO INTELIGÊNCIA ARTIFICIAL E APRENDIZAGEM DE MÁQUINA**

Quando nos aprofundamos um pouco mais no mundo de IA é comum nos confundirmos com vários termos como: *inteligência artificial, machine learning, aprendizagem de máquina, deep learning, ciência de dados, advanced analytics, redes neurais,*entre outros. Agora, qual é a diferença entre eles? O que é o quê? Vamos pensar nesses exemplos utilizando analogias.

1. Imagine uma **praça de alimentação** de um shopping: existem vários locais para comer e atendendo diversos gostos, como pizzas, massas, comidas típicas e vegetarianas, sorvetes, sanduíches e outros.
2. Agora, imagine que dentro dessa praça de alimentação existe um **restaurante** muito famoso. Pelo fato de ser um restaurante, você pode imaginar que diversos pratos são servidos, e para atender a todos os gostos.
3. Ainda que esse restaurante sirva vários pratos ele é mais conhecido pelos seus **pratos italianos**.
4. E, na cidade, “pratos italianos” é praticamente um sinônimo para **massas**.
5. Quando falamos em massas, uma **macarronada**bem-feita já é digna de muitos elogios e satisfaz o desejo de quase todos os clientes.
6. Por outro lado, quase sempre a **lasanha**vem à mente também. O problema, contudo, é que ela exige mais trabalho para fazer (e limpar) em comparação com a macarronada.
7. Só que existem vários tipos de lasanhas. Lasanhas mais complexas podem **ter mais camadas** de massa e recheio, e existem recheios mais complexos do que outros.



Crédito: Artisticco/ Shutterstock.

Dito isso, vamos explicar as analogias. A praça de alimentação é análoga ao universo de **Advanced Analytics**, uma área que engloba não somente IA, mas também a visualização de dados. Já ouviu falar em *Business Intelligence*, ou BI? Dentro das empresas, BI é praticamente sinônimo de relatórios e painéis visuais (*dashboards*) contendo vários gráficos mostrando o desempenho de indicadores importantes para a área de negócio. Esses indicadores podem ser, por exemplo, o histórico de vendas; a quantidade de problemas nas fábricas; o volume de produção registrado até o momento; entre outros. Note que, nesses exemplos, quase sempre falamos **do passado e do presente**. Esses dashboards, por outro lado, também podem ser combinados com outros algoritmos de IA e outras técnicas para prever o futuro. É como uma praça de alimentação de um shopping no qual podemos comprar as refeições de mais de um lugar e comer em um lugar só.

Já o restaurante é análogo ao mundo da **Ciência de Dados**. Veja que Ciência de Dados fica **dentro**do mundo de Advanced Analytics. Quando trabalhamos com Ciência de Dados não trabalhamos **somente**com IA, mas também usamos a Estatística e os nossos próprios conhecimentos em TI.

Os pratos italianos são análogos à **IA**: é um tópico interessante e conhecido no mundo todo. É também amplo e cheio de possibilidades. Para alguns, pode parecer simples. Para outros, pode parecer algo bem refinado e que exige um bom conhecimento para ser feito com qualidade.

Por outro lado, se você perguntar a uma pessoa qual é a primeira coisa que vem à cabeça dela é provável que tenha respostas bem diferentes. Para alguém novo, isso pode ser sinônimo de lasanha. Para uma pessoa com mais experiência de vida, por outro lado, pode ser sinônimo de um tiramisu artesanal. E, para você, pode ser um gelato. Observe que diferentes públicos podem possuir interpretações completamente diferentes sobre o que é um prato italiano. E, ainda, diferentes públicos possuem diferentes **expectativas**sobre o que seria um bom prato italiano. A mesma coisa é com IA: pessoas que não são da área podem possuir como expectativa um algoritmo superpoderoso e que aprende em um toque de mágica como resolver problemas. Imaginam uma IA próxima daquelas superinteligências que vemos nos filmes – como um assistente ou um robô que conversasse conosco e que resolvesse qualquer problema de qualquer área. Podem, até mesmo, imaginar algo próximo ao J.A.R.V.I.S. do Homem de Ferro (ou, por outro lado, a Skynet do Exterminador do Futuro). Ou, em outro extremo, podem entender que a IA hoje em dia é somente um termo novo para algo mais antigo: Estatística (e/ou Matemática).

Já as massas são análogas ao **machine learning**(ML ou **aprendizagem de máquina**). Apesar dos exemplos anteriores (lasanha, tiramisu e gelato), se entrevistássemos hoje um grande conjunto de pessoas e perguntássemos a eles por um exemplo de “prato italiano servido em restaurante”, certamente a grande maioria das respostas seria massas. E aí vai uma informação importante: **hoje, no contexto da maioria das empresas, ML e IA são sinônimos**.

O que se entende por IA foi mudando ao longo do tempo: até os anos 1990 IA poderia ser sinônimo de força bruta (o Deep Blue, famoso na década de 1990 por vencer o campeão mundial de xadrez Garry Kasparov, usava força bruta para ganhar: isto é, ele avaliava milhões de combinações futuras de jogadas e escolhia a melhor) e, ainda, de **sistemas especialistas** contendo várias regras construídas à mão: um sistema responsável pela segurança dos equipamentos e maquinário de uma usina hidrelétrica entra nessa regra, por exemplo. Por outro lado, esses algoritmos podem ser complexos para serem construídos e geridos – principalmente com o exponencial aumento de regras e necessidade de ajustes nessas regras ao longo do tempo. Já o que veio nos anos 2000 é o ML: algoritmos capazes de reconhecer padrões e aprender, de forma generalizada, a partir de uma base de dados histórica. É esse o tipo de algoritmo utilizado atualmente e entendido como “IA”. Já as superinteligências dos filmes (ainda) não são realidade, mas podem ser em um futuro próximo. Note que Ciência de Dados inclui IA, mas não é somente isto: também inclui Estatística, conhecimento sobre a área na qual se aplicaria o problema (como Vendas, Logística, RH, Jurídico e outros), e de computação.

**1.1 *MACHINE LEARNING***

Continuando: da mesma forma que a macarronada é um tipo de massa, existem técnicas e bibliotecas mais simples em ML, como o **scikit-learn**. Essa é uma das bibliotecas (lembra da nossa última conversa sobre bibliotecas?) mais conhecidas em Python e inclui técnicas simples como uma árvore de decisão ou um algoritmo de agrupamento de dados.

Por outro lado, a lasanha é outro tipo de massa mais complexa. Da mesma forma, também existem algoritmos mais complexos como as **redes neurais** – algoritmos que tiveram suas origens em trabalhos que tentaram traduzir a estrutura dos neurônios de um cérebro em algoritmos e circuitos. Esse tipo de algoritmo possui várias funções de ativação (como neurônios) agrupados em **camadas**. Existem, ainda, diversas conexões entre essas camadas. Sabemos que esse tipo de algoritmo é tratado como se fosse o melhor de todos, mas aqui vale um ponto de atenção: entenda as redes neurais como um “canhão para matar uma formiga”: às vezes, o problema a ser resolvido é tão simples que não precisaria de uma rede neural. Por outro lado, se mesmo assim você quiser usar uma rede neural, você pode acabar tendo um resultado pior do que um algoritmo mais simples – em outras palavras, o canhão também pode “errar” a formiga.

Existem várias arquiteturas de redes neurais como o multi-layer perceptron (MLP), recurrent neural network (RNN), feed-forward (FF), entre outros. Na prática, existem dezenas de arquiteturas: servem tanto para trabalhar com imagens, textos, sons, prever valores, grupos, entre outros: da mesma forma que também existem diferentes tipos de lasanhas. E, ainda, existem lasanhas com várias e várias camadas – da mesma forma, existem redes neurais com várias e várias camadas: essas redes se chamam *deep learning* – ou aprendizagem profunda. *A escolha da melhor arquitetura, biblioteca ou tipo de algoritmo depende muito de cada caso. Existem cientistas de dados que possuem algoritmos e técnicas de “estimação”, enquanto outros cientistas gostam de testar diferentes técnicas para cada problema. A escolha da melhor técnica que se aplica a cada caso é um exemplo de trabalho de Ciência de Dados.*

**TEMA 2 – TRABALHANDO COM PROBLEMAS DE CIÊNCIA DE DADOS**

Um projeto de Ciência de Dados sempre utiliza alguns passos – o nome dos passos e a ordem podem mudar ligeiramente dependendo de cada metodologia. Por outro lado, a execução possui alguns passos típicos. São eles:

1. a definição do problema;
2. a definição dos dados;
3. a preparação dos dados;
4. o desenvolvimento dos modelos;
5. a avaliação dos modelos; e
6. a disponibilização dos modelos.

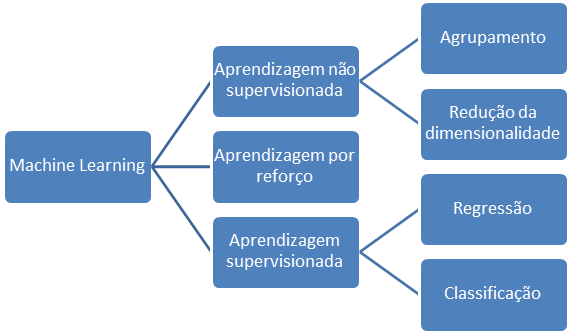
Vamos com calma por cada um desses itens. Primeiro, falaremos aqui sob um olhar de **projeto**, e depois sobre **execução**.

**2.1 DEFINIÇÃO DO PROBLEMA**

Esse passo trata, em sua essência, de responder: “o que você quer fazer com ML?”. Ou, ainda, em que ML pode te ajudar a resolver algum problema. Pensemos do ponto de vista de TI, e não sejamos vagos: “quero diminuir os gastos da empresa” é algo vago. Existem tantas formas de se diminuir os gastos e várias delas não envolvem ML – o consumo consciente de água por parte das pessoas já ajudaria isso, e nada teria a ver com ML.

Agora, um exemplo que poderia envolver ML é: “quero saber quais são os clientes que podem ser inadimplentes no futuro próximo”. Nesse caso, temos um exemplo específico do **escopo** do trabalho (os clientes), qual é a **dor** **da área de negócio** a ser endereçada (as inadimplências) e **o que se deseja** fazer (prever e apontar quem poderiam ser os futuros inadimplentes).

Ao definir o problema, poderemos saber qual é a melhor tipo de técnica a ser utilizada. ML é dividido em alguns subgrupos, conforme ilustrado pela figura a seguir.



A **aprendizagem supervisionada**seaplica quando sabemos **o que**queremos prever e em base **do quê**. Ela é dividida em dois principais tipos.

* **classificação**: quando queremos prever um “grupo”. Pode ser algo como cartão de crédito aprovado/cartão de crédito reprovado; sim/não; foto de cachorro/foto de gato; candidato pertencente ao grupo A/B/C.
* **regressão**: quando queremos prever um valor em uma “faixa numérica”. Pode ser algo como temperatura, cotação do Bitcoin, número de novos acessos no site, valor de venda da casa, nota do estudante.

Quando trabalhamos com problemas de aprendizagem supervisionada partimos da premissa de que precisamos de **uma base de dados**. Se queremos prever a temperatura de uma cidade para os próximos meses precisaremos de uma boa base histórica dela contendo, se possível, alguns anos de dados contendo informações que possuam relação com o que queremos prever (para pegarmos o comportamento dependendo da estação do ano; a relação entre temperatura, vento e chuva; entre outros); e, naturalmente, a coluna que queremos prever (no caso, a temperatura).

Caso queiramos prever se um futuro cliente terá ou não o seu pedido de cartão de crédito aprovado a partir do seu histórico, precisaremos ter também uma base de dados histórica com outros clientes, contendo dados que são relevantes à concessão (ou não) do cartão de crédito e incluindo também a coluna que queremos prever (no caso, se o cartão de crédito foi ou não aprovado). Ou, ainda, se queremos prever se uma nova imagem possui um gato ou cachorro dentro dela, precisaremos obrigatoriamente ter em mãos uma base de dados contendo várias outras fotos com cachorros ou gatos.

Já a **aprendizagem não supervisionada**trata majoritariamente da **exploração**de dados: às vezes, não sabemos exatamente **o que**queremos prever, mas precisamos manipular a base de dados, como a seguir.

* **redução da dimensionalidade**: quando precisamos selecionar quais dados realmente importam/são relevantes. Peguemos uma base de dados da meteorologia com 100 colunas diferentes: será que todaselas são importantes? Quando temos um número muito alto de dados, podemos cair naquilo que se chama de *maldição da dimensionalidade*: como várias técnicas de aprendizagem supervisionada em ML tratam de reconhecer padrões, fica bem difícil de detectar padrões com tantas colunas a serem analisadas. Logo, às vezes, é preferível selecionarmos somente aquelas que são realmente úteis para o nosso problema.
* **agrupamento** (também mencionado, às vezes, como *clusterização*): peguemos as mesmas 100 colunas que comentamos anteriormente – será que não conseguiríamos de alguma forma agrupá-las de acordo com a sua similaridade em novas colunas de um jeito que não percamos muita informação ao fazermos isso?

Por fim, a **aprendizagem por reforço** inclui técnicas que são orientadas a resolver problemas a partir de **recompensas**. Incluem-se aqui algoritmos que buscam otimizartrabalhos complexos como a distribuição de rotas logísticas ou de escalas de trabalho tentando descobrir quais pequenos ajustes levam a um melhor resultado ou, ainda, certos algoritmos usados em jogos de computador e de console.

É possível dizer que, no contexto do mercado de trabalho, grande maioria dos trabalhos de Ciência de Dados se concentram em problemas de aprendizagem supervisionada e não supervisionada.

**2.2 DEFINIÇÃO DOS DADOS**

Um bom projeto de ciência de dados costuma ter também o apoio de quem utilizará essa solução e que entende do problema. Nesse caso, quem entende do problema são as pessoas que lidam com as inadimplências hoje. É imprescindível que elas caminhem junto para resolver o problema. Pensemos aqui: quais fatores impactam? A visão do cientista de dados e do analista da área de negócio que o apoiará são igualmente importantes:

* o **analista da área de negócio**dará várias informações que fazem parte já da **experiência**dele: ele provavelmente pode citar itens como o período do ano (já que podem ter épocas do ano em que a inadimplência é maior); o score de crédito; a data de fundação do comércio; entre outros. Essas informações são valiosas para você por dois motivos.
  + ganho de tempo: você “aproveita” a experiência dessa pessoa ao descobrir com antecedência quais informações **podem ser**importantes. Às vezes, a intuição da pessoa pode estar errada em alguns pontos, mas estará certa em outros: logo, sempre haverá algo bom a se aproveitar.
  + confiança: é o analista (e/ou a equipe na qual essa pessoa faz parte) quem fará uso do algoritmo. Como quase nada na vida é 100% previsível, é improvável que o algoritmo acerte 100% das vezes. Por outro lado, se essa pessoa nunca participou no processo de construção do algoritmo e não sabe quais fatores foram levados em conta, no primeiro momento em que o algoritmo der um resultado diferente do esperado haverá uma desconfiança. E, como você deve imaginar, quando alguém não confia em algo é bem difícil de mudarmos a sua opinião.
* o **cientista de dados** ajudará a trazer um conhecimento fora do contexto do dia a dia do analista: é parte do trabalho do cientista de dados pesquisar em bases científicas quais foram os principais achados sobre um determinado problema: no caso da inadimplência, existem vários estudos apontando quais fatores culminam na inadimplência – pode ser que alguns desses fatores não sejam do conhecimento do analista e poderão ser úteis para o seu trabalho. Logo, um trabalho de ciência de dados vai além de criar somente um algoritmo – é também o de ajudar os humanos em seu trabalho. Além disso, ele poderá trazer a sua experiência em projetos anteriores para fazer perguntas que possam ajudar no desenvolvimento do algoritmo: será que a tendênciado score nos últimos 12 meses é também relevante? Um score baixo pode ser uma coisa, mas e se o score estiver melhorando? Ou, ainda, e se o score é alto, mas está caindo vertiginosamente? Será que isso não é um sinal preocupante? Será que a inflação ao longo do tempo também não poderia ser relevante? Ou a cidade? Cidades menores e em regiões rurais podem ter uma dependência maior com a época da colheita de principais safras, ou cidades mais turísticas podem receber movimentar mais dinheiro nos finais de ano ou feriados estendidos. Veja que é um pensamento “fora da caixa” que pode ser bem-vindo durante o desenvolvimento do projeto.

Com o apoio dessas pessoas, é possível definir **a base de dados**a ser utilizada. Isso ocorre principalmente porque algoritmos de ML não “descobrem” sozinhos novos dados: é importante que um humano informe ao algoritmo que conjunto de dados deve ser utilizado. Esse conjunto pode vir de um banco de dados da própria empresa ou, ainda, de fontes de dados externas como sites governamentais, empresas de consultoria especializadas ou sites de acesso público. A preocupação aqui é a de garantir que a base de dados faça sentido no futuro: ou seja, que todas as informações a serem usadas estejam disponíveis também ao testarmos novos dados futuros.

Peguemos novamente o caso da inadimplência: não adianta treinar um algoritmo para prever casos futuros de inadimplência usando como apoio uma base de dados governamental que foi desativada no mês passado. Novos casos jamais poderão utilizar essa mesma base e, assim, você tem um algoritmo que não funcionará corretamente no futuro.

Algoritmos de ML são capazes de aprender sobre um conjunto de dados predefinido e, a partir dele, cria-se uma **generalização**para que possa funcionar bem com casos futuros. Imaginemos o próprio caso dos nossos estudos: você não quer só criar algoritmos com os conjuntos de dados trabalhados com o que nós aprendemos, não é? Na realidade, você quer aprender diferentes tipos e aplicações de algoritmos, saber o que funciona e o que não funciona para que você possa **aprender com os exemplos** e saber aplicar em **novos cenários**no futuro.

É como se você estivesse aprendendo a dirigir um carro na autoescola pela primeira vez: você não quer aprender a dirigir somente o Volkswagen Up 2016 1.0 branco, placa ABC-1234 da “Autoescola Centro” e somente no bairro “Ribeira”, mas você quer aprender diferentes situações de tempo, ruas, trânsito e outros carros para que você saiba o que fazer com outros carros, outros horários e outras ruas, não é? Por outro lado, para que você saiba o que fazer, você precisa ser exposto a diferentes cenários enquanto está aprendendo. Como você vai saber como estacionar o seu próprio carro se nunca teve contato com isso na vida? Como irá reagir se nunca você nem viu uma aula ou vídeo ensinando isso?

**2.3 PREPARAÇÃO DOS DADOS**

Com os algoritmos de ML é a mesma coisa: precisamos expor a eles as situações nas quais desejamos que ele aprenda para que ele saiba como reagir no futuro. Se vamos criar um algoritmo que vá prever a temperatura nas próximas horas é importante que forneçamos uma base para que o algoritmo aprenda. Agora, que base de dados é essa? Vamos imaginar uma planilha de Excel. Vamos pensar na sua própria cidade: qual é a granularidade? Queremos prever por hora? Por dia? Por semana? Por mês? Bom, comentamos anteriormente que seria por hora – logo, teremos **no mínimo** duas informações, a data com a hora em uma coluna e a temperatura em outra. Vamos usar os dados do Instituto Nacional de Meteorologia (INMET) para a cidade de Curitiba como base:

|  |  |
| --- | --- |
| **DataHora** | **Temperatura** |
| 2021-01-01 00:00 | 20.5 |
| 2021-01-01 01:00 | 19.9 |
| 2021-01-01 02:00 | 19.7 |
| 2021-01-01 03:00 | 19.6 |
| 2021-01-01 04:00 | 19.9 |
| 2021-01-01 05:00 | 18 |
| 2021-01-01 06:00 | 18 |
| 2021-01-01 07:00 | 17.7 |
| 2021-01-01 08:00 | 18.3 |
| 2021-01-01 09:00 | 18.9 |
| 2021-01-01 10:00 | 20.3 |
| 2021-01-01 11:00 | 21 |
| 2021-01-01 12:00 | 21 |
| 2021-01-01 13:00 | 20 |
| 2021-01-01 14:00 | 22.8 |

Veja que essa pequena amostra somente possui dados de algumas horas do dia 1º de janeiro de 2021. Vamos imaginar que queiramos prever a temperatura das 15h do dia 1º de janeiro de 2021 (em que, a partir de agora, sempre deixaremos no padrão *ano-mês-dia hora:minuto*, também convencionado nas documentações das bibliotecas como *yyyy-mm-dd hh:mm*– logo, essa data será escrita como 2021-01-01 15:00). Bom, nós, humanos, esperamos que a temperatura fique por volta dos 22 graus. Sabemos disso porque nas últimas horas a temperatura se manteve nessa faixa e porque é uma tarde de verão. Quer dizer, pode ser que abaixe um pouco (talvez uns 19 graus) ou aumente (talvez 25 graus), talvez? Veja que, para chegar a essa decisão, temos todo um conjunto de conhecimentos prévios: sabemos que a temperatura geralmente não muda muito hora a hora; sabemos que a temperatura à tarde geralmenteé maior do que a temperatura de madrugada; sabemos que uma mudança de 2 ou 3 graus é aceitável, mas 20 ou 30 graus de uma hora para outra não. Veja, também, que esse conhecimento depende de onde vivemos: se você cresceu em uma cidade na qual a temperatura não muda muito de uma hora para outra, você provavelmente chegará à decisão diferente de outra pessoa que está acostumada a grandes variações na temperatura (como é o próprio caso de Curitiba).

Agora, como um algoritmo vai aprender todos esses conceitos? Um algoritmo não sabe o que é “hora”, “sol” ou “Curitiba”. É como se tudo fosse grego para ele – algo incompreensível, algo assim:

|  |  |
| --- | --- |
| **ΗμερομηνίαΩρα** | **Θερμοκρασία** |
| ??? | 20.5 |
| ??? | 19.9 |
| ??? | 19.7 |
| ??? | 19.6 |
| ??? | 19.9 |
| ??? | 18 |
| ??? | 18 |
| ??? | 17.7 |
| ??? | 18.3 |
| ??? | 18.9 |
| ??? | 20.3 |
| ??? | 21 |
| ??? | 21 |
| ??? | 20 |
| ??? | 22.8 |

Mas, e aí? Como resolvemos isso? Uma alternativa é incluirmos as informações que são importantes para que o algoritmo tome uma decisão. O que influencia muitoa temperatura em Curitiba? Será que a quantidade de liquidificadores ligados ao mesmo tempo em São Paulo influencia? Ou seria a umidade, ou a incidência do sol (isto é, se o céu está limpo ou nublado, por exemplo)? O fato de estar chovendo ou não, talvez? Ou o vento? Ou a quantidade de potes de margarina vendidos no maior supermercado da cidade?

Veja que existem informações mais úteis do que outras, e algumas delas realmente mais atrapalham do que ajudam. Não sei se percebeu, mas às 14h a temperatura aumentou quase 3 graus. Se você fosse apostar a temperatura às 15h olhando somente o histórico da temperatura, o que você apostaria? Agora observe como fica a tabela quando incluímos a informação da radiação solar:

|  |  |  |
| --- | --- | --- |
| **DataHora** | **Temperatura** | **RadiacaoSolar** |
| 2021-01-01 00:00 | 20.5 |  |
| 2021-01-01 01:00 | 19.9 |  |
| 2021-01-01 02:00 | 19.7 |  |
| 2021-01-01 03:00 | 19.6 |  |
| 2021-01-01 04:00 | 19.9 |  |
| 2021-01-01 05:00 | 18 |  |
| 2021-01-01 06:00 | 18 |  |
| 2021-01-01 07:00 | 17.7 |  |
| 2021-01-01 08:00 | 18.3 | 33.3 |
| 2021-01-01 09:00 | 18.9 | 268.2 |
| 2021-01-01 10:00 | 20.3 | 583.9 |
| 2021-01-01 11:00 | 21 | 848.6 |
| 2021-01-01 12:00 | 21 | 572.5 |
| 2021-01-01 13:00 | 20 | 621.6 |
| 2021-01-01 14:00 | 22.8 | 2513.2 |

A radiação solar aumentou muito de 13h para 14h: aparentemente o céu ficou com menos nuvens nessa hora. E agora? Você mudaria o valor da sua “aposta” na temperatura às 15h? Veja o que realmente aconteceu:

|  |  |  |
| --- | --- | --- |
| **DataHora** | **Temperatura** | **RadiacaoSolar** |
| 2021-01-01 00:00 | 20.5 |  |
| 2021-01-01 01:00 | 19.9 |  |
| 2021-01-01 02:00 | 19.7 |  |
| 2021-01-01 03:00 | 19.6 |  |
| 2021-01-01 04:00 | 19.9 |  |
| 2021-01-01 05:00 | 18 |  |
| 2021-01-01 06:00 | 18 |  |
| 2021-01-01 07:00 | 17.7 |  |
| 2021-01-01 08:00 | 18.3 | 33.3 |
| 2021-01-01 09:00 | 18.9 | 268.2 |
| 2021-01-01 10:00 | 20.3 | 583.9 |
| 2021-01-01 11:00 | 21 | 848.6 |
| 2021-01-01 12:00 | 21 | 572.5 |
| 2021-01-01 13:00 | 20 | 621.6 |
| 2021-01-01 14:00 | 22.8 | 2513.2 |
| 2021-01-01 15:00 | 23.2 | 2337.6 |

A radiação solar diminuiu um pouco, mas permaneceu em um patamar alto. A temperatura aumentou um pouco também, pulando de 22.8 para 23.2 graus. É nesse sentido que um cientista de dados poderia trabalhar: entendendo e mapeando quais variáveis poderiam ajudar na tomada de decisão. Nesse caso, poderíamos incluir a umidade, vento e pressão atmosférica (mas não os liquidificadores ou os potes de margarina, que nada possuem relação com o nosso caso).

**2.4 DESENVOLVIMENTO DOS MODELOS**

O passo de gerar uma base de dados confiável (o que também inclui limpara base de dados de possíveis erros – se alguém errou um ponto a temperatura de algum dia pode ser apontada como “213” em vez de “21.3” graus) e o passo de desenvolver os modelos pode demandar até 80% de um projeto. É um trabalho que envolve muitos testes e, naturalmente, muito “vai e volta”.

Para os casos de **aprendizagem supervisionada**, retornemos novamente ao caso das inadimplências: após termos uma base de dados em mãos poderemos realizar o treinamento de um modelo preditivo. Para isso, é importante dividirmos a nossa base de dados original em pelo menos duas partes.

* A **base de treino** é uma amostra obtida a partir da base original (entre 60% e 80% da base na maioria das vezes) e é essa base que utilizamos para *treinar*um algoritmo – isto é, é ela que utilizamos para que um algoritmo reconheça e “aprenda” os padrões que estão dentro daquela base e, se possível, algumas exceções.
* A **base de testes** é o restante da base que não foi utilizada pela base de treinamento. Ela é usada para avaliarum algoritmo. Esse passo é importante porque representa como ele funcionaria na vida real e nos dá uma forma de compararum valor previsto com um valor que de fato aconteceu já que nós (humanos) sabemos o resultado que o algoritmo deveria prever, mas ele (o algoritmo) não. Afinal de contas, é fácil o algoritmo ter uma taxa de acerto alta olhando somente para a base de treino, mas não é nada representativo sobre como o algoritmo iria desempenhar no dia a dia: já a métrica de acerto sobre a base de testes possui uma representatividade bem melhor nesse sentido.

Ainda nesse passo, escolheremos o algoritmo a ser utilizado. Existem várias técnicas.

* aprendizagem supervisionada:
  + modelos lineares;
  + algoritmos estatísticos;
  + redes neurais;
  + árvores de decisão;
  + máquinas de vetor de suporte (*support vector machines*, ou SVMS);
  + florestas (múltiplas) de árvores de decisão; e
  + comitês de algoritmos.
* aprendizagem não supervisionada:
  + análise fatorial;
  + análise de componentes principais (*principal component analysis*, ou PCA);
  + K-médias (*k-means*); e
  + eliminação recursiva de atributos.

**2.5 AVALIAÇÃO DOS MODELOS**

Após treinar um modelo, é importante avaliarmos se ele está funcionando como esperado (ou não). Falar de “acurácia” é um termo complicado porque existem dezenas de métricas que possuem como objetivo explicar se um modelo está funcionando como esperado – logo, a forma de se calcular essa acuracidade muda radicalmente dependendo do caso.

Para saber se os resultados estão funcionando de acordo com o esperado, as métricas mais comuns são as que se segue.

* aprendizagem supervisionada
  + regressão:
    - erro médio absoluto (*mean absolute error*, ou MAE);
    - erro quadrático médio (*mean squared error*, ou MSE);
    - erro médio absoluto percentual (*mean absolute percentage error*, ou MAPE); e
    - raiz quadrada do erro quadrático médio (*root* *mean squared error*, ou RMSE).
  + classificação:
    - matriz de confusão;
    - precisão e revocação (*precision*e *recall*); e
    - curva ROC (*receiver operating characteristic*) e a área sob essa curva (*area under curve*, ou AUC).
* aprendizagem não supervisionada
  + testes estatísticos, como os testes de hipótese.

**2.6 DISPONIBILIZAÇÃO DOS MODELOS**

Após os resultados estarem aceitáveis (e a definição do “aceitável” depende muito de cada caso: às vezes, 75% de acerto para as inadimplências pode ser algo muito bom, mas os mesmos 75% para um algoritmo da área médica pode ser completamente inaceitável), o modelo pode ser disponibilizado para o uso diário.

Essa disponibilização (*deployment*) do modelo trata de deixá-lo disponível para um uso futuro. Isso inclui a sua integração com uma arquitetura técnica em uso por uma empresa e a forma que os dados serão alimentados ao algoritmo e seus resultados propriamente utilizados. É nesse ponto que se determina quem visualizará os valores e de qual forma.

**TEMA 3 – TRABALHANDO COM DADOS**

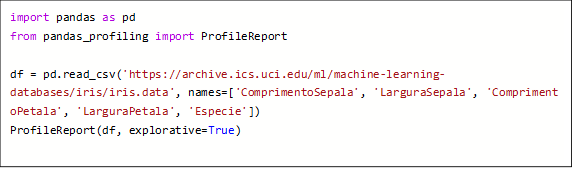
Agora que conversamos sobre o passo a passo de um projeto típico de Ciência de Dados, poderemos falar sobre o trabalho em si. Existem trabalhos nessa área que são executados em várias linguagens de programação, como Scala, R, Java e Python. Do ponto de vista corporativo, o Python recentemente se tornou a mais utilizada dentre essas.

As bases de dados (*datasets*) trabalhadas em ML possuem alguns tipos – os principais são os que se seguem.

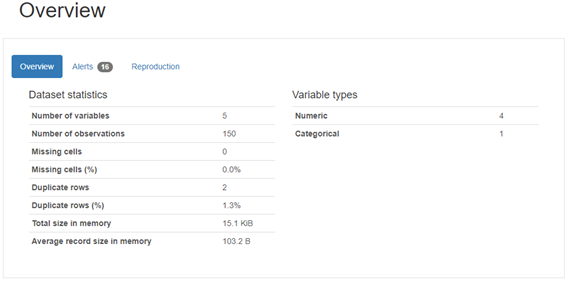
1. Dados **tabulares**: são aqueles que se parecem com planilhas de Excel, e são os mais comuns. São caracterizados por **atributos** (ou *features*): características (colunas) que fazem parte da base; e **instâncias**, isto é, indivíduos (linhas) que fazem parte da base de dados.
   1. Para problemas de aprendizagem supervisionada podemos ter um **label** ou **target** – um atributo que é aquele que queremos prever.
   2. Os dados podem ser numéricos (como valores referentes à temperatura, idade, preços e afins); categóricos (como aprovado/reprovado; as categorias da Carteira Nacional de Habilitação; tipos sanguíneos; entre outros); dados temporais (que possuem relação com data e hora, por exemplo); e textos (como esta descrição que você está lendo agora).
2. Bases de **imagens**: são fotos ou outras imagens as quais queremos detectar o seu contexto (por exemplo: ao tirarmos uma foto de dentro de um comércio estamos interessados em saber se ele é uma loja de autopeças ou um supermercado “ao olhar o todo”, mas não que estejamos interessados em reconhecer cada item das gôndolas) ou seus objetos (ou seja, quais produtosestão dentro das gôndolas sendo mostradas na foto).
3. Bases de **texto**: são conjuntos de texto nos quais queremos entender o seu contexto; encontrar entidades (pessoas ou lugares); detectar similaridades entre documentos; traduções; entre outros.

Independentemente da base sendo trabalhada é importante relembrar de um conceito chamado *garbage in, garbage out*(entra lixo, sai lixo): algoritmos de ML aprendem em cima de uma base em que os humanos fornecem. Logo, é importante que nós (humanos) repassemos uma base confiável e previamente curada por nós. Ainda que a ideia possa soar interessante, por enquanto, os algoritmos ainda não conseguem adivinhar o que nós pensamos e, ainda, procurar novos dados da internet sem supervisão alguma.

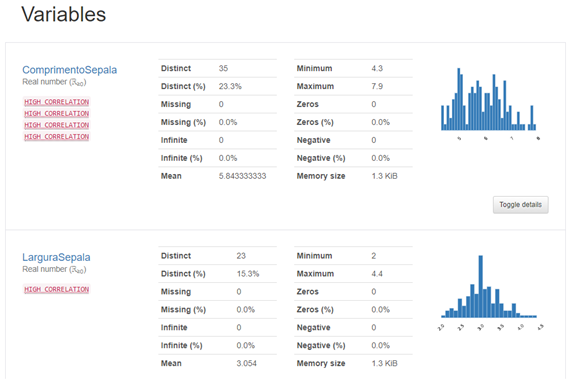
Nesse passo, nós podemos realizar uma **análise de dados exploratória** para ver como está o relacionamento entre os dados. Uma das bases de dados mais simples é a *Iris*, a qual possui dados sobre o comprimento e a largura das sépalas e pétalas de três espécies de flor do gênero Iris. O código a seguir combina duas bibliotecas para realizar a análise exploratória: o pandasspan >, responsável por gerar o DataFrame a partir dos dados de um arquivo CSV, e o span >pandas-profilingspan >, o qual nos gera um poderoso relatório sobre um DataFrame.



O relatório resultante nos fornece várias informações, como: o número de atributos (*variables*) e instâncias (*observations*), bem como os dados nulos e o espaço em memória gasto:



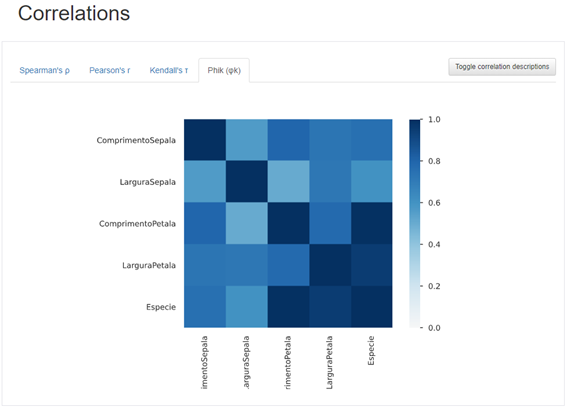
A distribuição dos dados entre as variáveis:



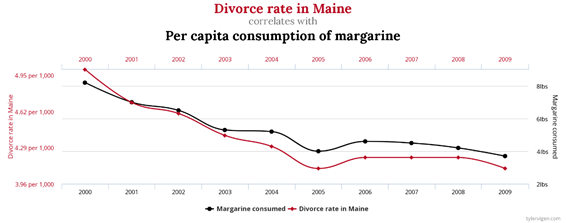
Dados duplicados:



As correlações com diferentes técnicas:



Entre outros. Isso é útil para detectarmos possíveis relacionamentos importantes entre as variáveis e que podem ser úteis para validar a base antes de avançarmos com os modelos. Sobre as correlações, vale também comentar sobre as “correlações espúrias”: não é porque temos uma alta correlação entre duas variáveis que isso significa alguma coisa útil. Um exemplo clássico é a alta correlação entre a porcentagem de divórcios no estado do Maine, nos EUA, com o consumo per capita de margarina:



Fonte: Vigen, [S.d.].

Observe que ambos possuem uma correlação altíssima: quando um aumenta, o outro também aumenta; quando um abaixa, o outro também abaixa. Por outro lado, um não influencia no outro de forma alguma. Tome o mesmo cuidado ao trabalhar com bases de dados, pois certamente não queremos detectar correlações espúrias nela!

Nesse momento, também podemos criar atributos adicionais ou manipular os que já temos. Os tamanhos máximo e mínimo de todas as dimensões da sépala e pétala são diferentes, e existem algoritmos (como as próprias redes neurais) que se beneficiam se todos estiverem na mesma escala. Logo, o código a seguir já poderia endereçar isso usando a MinMaxScaler:

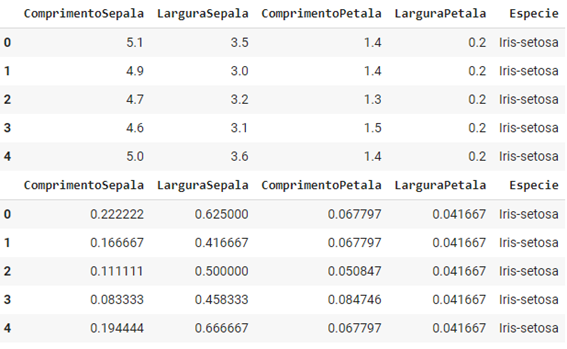
from sklearn.preprocessing import MinMaxScaler

display(df.head())

# colocando todas as colunas na mesma escala numérica
df_scaled = df.copy()
df_scaled[['ComprimentoSepala', 'LarguraSepala', 'ComprimentoPetala', 'LarguraPetala']] = MinMaxScaler().fit_transform(df_scaled[['ComprimentoSepala', 'LarguraSepala', 'ComprimentoPetala', 'LarguraPetala']])
display(df_scaled.head())



Agora, veja como eram as primeiras cinco linhas da tabela **antes**de colocarmos todos na mesma escala e como ficou depois:



Existem ainda outras técnicas, como:

* a transformação de categorias em múltiplas colunas numéricas;
* a extração de números como a semana, o dia da semana ou o mês a partir de uma coluna de data e hora; e
* o preenchimento de valores nulos com zeros ou outros valores.

**TEMA 4 – TRABALHANDO COM MODELOS**

Antes de qualquer coisa, vale um lembrete: **um modelo de Ciência de Dados não é necessariamente um modelo de ML!**Dependendo do caso, podemos ter modelos de áreas como Estatística e Econometria, por exemplo. Por outro lado, ML possui várias aplicações úteis no dia a dia.

As bibliotecas mais conhecidas em Python para a criação de modelos de ML são as que se seguem.

* Scikit-learn
  + O scikit-learn é potencialmente a biblioteca de ML mais versátil. Ela possui implementações de técnicas de aprendizagem supervisionada e não supervisionada.
  + Seus modelos vão desde a regressão linear até modelos de redes neurais, passando por SVMs, comitês de algoritmos, árvores de decisão e outros.
  + Possui técnicas de pré-processamento de dados e métricas de acuracidade dos algoritmos.
* Tensorflow, Keras e PyTorch
  + Os três são bibliotecas para o desenvolvimento de algoritmos de redes neurais (e *deep learning*). Pela complexidade desse tipo de algoritmo, recomenda-se que sejam usados computadores com mais poder de processamento.
  + O Tensorflow também possui um playground disponível para se testar diferentes configurações de redes com problemas simples e, assim, ver como o processo de treinamento de uma rede neural funciona na prática.
* LightGBM e XGBoost
  + Ambas as bibliotecas trabalham com *gradient boosting*, uma técnica de ML a qual é composta por várias árvores de decisão. Nessa técnica, as árvores vão sendo adicionadas no modelo para se obter um bom resultado de forma que uma possa reduzir o erro das anteriores.
* Statsmodels
  + O statsmodels é uma biblioteca com foco estatístico, e não exatamente em ML. De qualquer modo, ela permanece sendo relevante ao incluir funções para séries temporais (como o ARIMA) e outras técnicas de estatística descritiva.

Vejamos um exemplo do uso do scikit-learn. Testaremos dois modelos diferentes (RandomForest e SVC – um tipo de SVM específico para problemas de classificação) para a base de dados Iris, a qual comentamos anteriormente. Na prática, seria algo como: “aprenda os padrões entre os tamanhos da pétala e sépala de uma flor que determinam qual é a espécie dela. Validaremos em seguida com ‘novos’ exemplares de flor”.

Para melhor ilustrar como um algoritmo aprendeu utilizaremos também a biblioteca mlxtend junto com o matplotlib, que já vimos antes. Dito isso, observe o seguinte código. Note também os comentários (sempre iniciados com um “#”) para que saiba o que cada seção do código está fazendo. Nesse caso, estamos considerando somente o comprimento (e não a largura) da pétala e da sépala para tomar uma decisão.

Observe também que introduzimos duas novas funções que não vimos antes: a OrdinalEncoder, a qual transforma somente as categorias da coluna “Especie” que hoje estão como texto (“Iris-setosa”, “Iris-virginica” e “Iris-versicolor”) para números (0, 1 e 2). Essa transformação é necessária para utilizar a função plot\_decision\_regions do mlxtend. Além disso, usamos o train\_test\_split para dividir a nossa base de dados em duas partes: uma de treinamento e outra de teste. Veja que o parâmetro “test\_size” está com o valor 0.4. A explicação sobre esse parâmetro se encontra na própria documentação do scikit-learn e indica a porcentagem de distribuição aleatória para a base de treino e teste: no caso, 40% para a base de teste e os 60% restantes para a base de treinamento.

Ainda, veja que em alguns momentos existe também o parâmetro “random\_state”. Ele também é útil ao permitir a **reprodutibilidade** dos nossos testes. Vários algoritmos inicializam de forma aleatória e, para fins de teste e confiabilidade do algoritmo, em alguns casos, é importante garantir que conseguimos ter o mesmo resultado se rodarmos o mesmo código hoje ou amanhã, ou em computadores diferentes considerando a mesma base de dados. Dessa forma, ao indicarmos um valor para o “random\_state” (que chamamos de *seed*), garantiremos que o número aleatório utilizado para gerar a lógica do algoritmo será sempre o mesmo.

Veja, também, que no *train\_test\_split*não estamos só informando o dataframe *df*, mas sim dois parâmetros: df[['ComprimentoSepala', 'ComprimentoPetala']] (isto é, os atributos que queremos considerar no treinamento do modelo) e df['Especie'] (ou seja, o atributo que desejamos prever). O resultado, então, são quatro variáveis: os atributos utilizados para treinar e testar o modelo (X\_train e X\_test, respectivamente) – isto é, as variáveis que devem ser usadas pelo modelo aprender e reconhecer padrões para prever o nosso *target* (no caso, o atributo “Especie”); e o *target*em si, também correspondente à base de treinamento e teste e exatamente na mesma ordem do X\_train e X\_test, e que convenientemente chamamos de y\_train e y\_test.

# importação de libraries
import matplotlib.pyplot as plt

from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import OrdinalEncoder
from sklearn.model_selection import train_test_split

from mlxtend.plotting import plot_decision_regions

# transformando o texto da espécie em categorias numéricas
df['Especie'] = OrdinalEncoder().fit_transform(df[['Especie']]).astype(int)

# dividindo a base de dados entre treino e teste
X_train, X_test, y_train, y_test = train_test_split(df[['ComprimentoSepala', 'ComprimentoPetala']], df['Especie'], test_size=0.4, random_state=0)

# treinando o modelo de RandomForest (RF)
modelo_rf = RandomForestClassifier(random_state=0)
modelo_rf.fit(X_train, y_train)

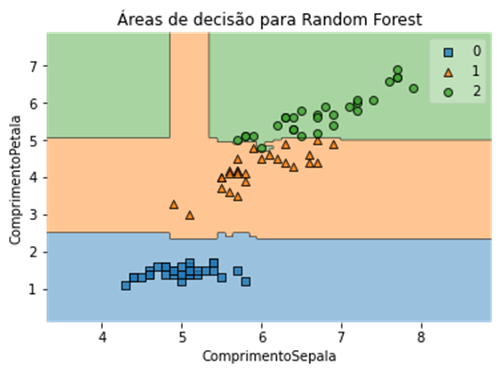
# mostrando como ele tomou decisões
fig = plot_decision_regions(X=X_train.values, y=y_train.values, clf=modelo_rf)
plt.xlabel('ComprimentoSepala')
plt.ylabel('ComprimentoPetala')
plt.title('Áreas de decisão para Random Forest')
plt.show()

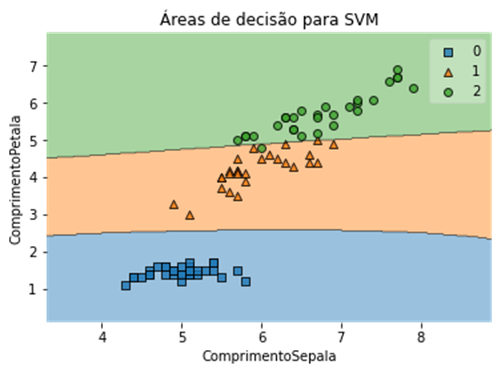
# treinando o modelo de GradientBoostingClassifier (RF)
modelo_svm = SVC(random_state=0)
modelo_svm.fit(X_train, y_train)

# mostrando como ele tomou decisões
fig = plot_decision_regions(X=X_train.values, y=y_train.values, clf=modelo_svm)
plt.xlabel('ComprimentoSepala')
plt.ylabel('ComprimentoPetala')
plt.title('Áreas de decisão para SVM')
plt.show()



Agora, veja como cada um dos modelos aprendeu:





**TEMA 5 – TRABALHANDO COM RESULTADOS**

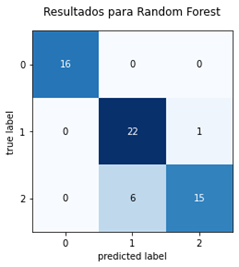
Veja que, no exemplo anterior, dividimos a base em uma de treinamento e de teste. Por outro lado, somente mostramos como o algoritmo aprendeu a partir da base de treinamento (no caso, X\_train e y\_train). Lembra quando comentamos que um dos passos necessários é a **avaliação**dos resultados? Pois bem – vejamos quais foram os resultados via matriz de confusão para as duas técnicas. Veja que agora estamos utilizando a base de testes para validar os resultados:

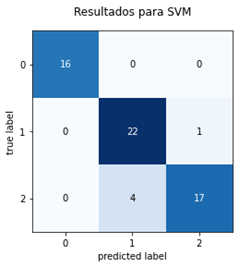
from sklearn.metrics import confusion_matrix
from mlxtend.plotting import plot_confusion_matrix

plot_confusion_matrix(confusion_matrix(y_test, modelo_rf.predict(X_test)))
plt.title('Resultados para Random Forest')

plot_confusion_matrix(confusion_matrix(y_test, modelo_svm.predict(X_test)))
plt.title('Resultados para SVM')







Veja que cada linha indica a classe real e cada coluna indica qual foi o valor predito. Queremos o máximo de acertos para todas as classes. Logo, gostaríamos de ter o máximo de valores na diagonal principal da matriz: o máximo de valores para a primeira coluna da primeira linha; da segunda coluna da segunda linha e da terceira coluna da terceira linha. Veja que ambos acertaram todos os valores da classe 0 e erraram somente 1 caso da classe 1 (no caso, houve um 1 caso em que uma flor era da classe 1 e os algoritmos incorretamente classificaram como sendo da classe 2). O Random Forest classificou corretamente 15 flores como sendo da classe 2, mas 6 flores da mesma classe foram incorretamente classificadas como sendo da classe 1. O SVM foi um pouco melhor: acertou 17 flores e errou 4 da mesma classe.

**FINALIZANDO**

Aqui, nesta nossa conversa, tivemos a oportunidade de prover a você uma breve introdução ao mundo de Ciência de Dados em Python. O termo “Ciência” pressupõe o uso da Metodologia Científica para a análise e modelagem de dados e, com isso, também requer um rigor científico igualmente importante. Por essa razão, naturalmente, há uma fundamentação teórica forte por trás de um trabalho dessa área a qual não se restringe somente a ML: também é perfeitamente possível usarmos técnicas estatísticas para resolver esse problema.

Isso posto, mostramos o que se entende por ML, IA, Ciência de Dados, Advanced Analytics e demais termos utilizando uma analogia de uma praça de alimentação. Esse tipo de comparação é importante, uma vez que os termos podem ser confusos e serem trocados entre si ainda que representem coisas completamente diferentes. Também mostramos como funciona um projeto de Ciência de Dados – tanto do ponto de vista de gestão como do ponto de vista de implementação com Python.

Também ilustramos quais são as principais bibliotecas em Python para trabalharmos com ML com exemplos e mostrando algumas possibilidades da construção de algoritmos. Veja que estamos trabalhando com bibliotecas da mesma forma que vimos anteriormente e que aos poucos vamos também expandindo o nosso “repertório” de bibliotecas como o Tensorflow, scikit-learn, LightGBM, mlxtend e outros.

Aula 3

Mudando paradigmas

Transpondo conceitos da vida real em código

Programação Orientada a Objetos (POO)

Um problema para ser resolvido pela Programação Orientada a Objetos

Programação estruturada

Método até agora:   
programação estruturada

Receita de bolo

Funções Estruturas de seleção

Estruturas de repetição

Entrada e saída de dados

Programação Orientada a Objetos

Segurança

Confiança

Sistemas corporativos

Internet Banking

E-commerce

Sistemas contábeis

Sistemas corporativos (ERPs)

Noções de Programação Orientada a Objetos

Programação Orientada a Objetos

Unified Modeling Language (UML)

Programação Orientada a Objetos (POO)

Structured Query Language (SQL)

Objetos

"Coisas" da vida real

Dois objetos não ocupam o mesmo lugar ao mesmo tempo

Estados Características

Estados

Variáveis: atributos

Comportamentos

Ações: funções (métodos)

Classes

"Manual" (blueprint) para se criarem objetos

Receita de bolo, não o bolo

Planta da casa, não a casa

Instância de classe: objeto

Exemplo

Classe "Cachorro"

Métodos >> Latir Comer Dormir

Atributos >> Cor Nome Brinquedo favorito

Objeto "Kora": instância de "Cachorro"

Objeto "Pippoca": instância de "Cachorro"

Objeto "Dexter": instância de "Cachorro"

Heranças

Herança de classes

Classes especializadas

Herdam métodos e atributos

Carro

Caminhonete

Furgão

Minivan

Bicicleta

BMX

Bicicleta de Carga

Bicicleta de Pista

Residência

Sobrado

Casa de Praia

Apartamento

Cliente de Internet Banking

Conta simples

Conta premium

Conta empresarial

Exemplo

Classe "Cachorro"

Métodos

Latir Comer Dormir

Atributos Cor Nome Brinquedo favorito

Classe "Cão Farejador"

Métodos Farejar Sentar

Atributos Tempo de Treinamento Aeroporto Raça

Interfaces

Classes podem usar interfaces

Interfaces listam os métodos que uma classe que a usa deve ter

Interfaces não implementam métodos

Classes e Objetos em Python

Criando uma classe

class [nome da classe]

Atributos Métodos self

Exemplo

class Bolo:

peso = 0

def atribuir\_peso(self, peso\_a\_atribuir):   
 self.peso = peso\_a\_atribuir

bolo\_aniversario = Bolo()   
print(bolo\_aniversario.peso)   
  
bolo\_aniversario.atribuir\_peso(2)   
print(bolo\_aniversario.peso)

Inicializando uma classe

def \_\_init\_\_

Exemplo

class Bolo:

peso = 0

def \_\_init\_\_(self, peso\_inicial=0):   
 self.peso = peso\_inicial

def atribuir\_peso(self, peso\_a\_atribuir):   
 self.peso = peso\_a\_atribuir

bolo\_aniversario = Bolo() print(bolo\_aniversario.peso)

bolo\_aniversario.atribuir\_peso(2) print(bolo\_aniversario.peso)

Polimorfismo

Implementações diferentes do mesmo método

Parâmetros diferentes

Resultados diferentes

Veículo: estacionar

Carro: estacionar

Moto: estacionar

Caminhão: estacionar

Bicicleta: estacionar  
  
Closures e Decorators  
  
Encapsulamento

Funções dentro de funções

Redução de casos de variáveis globais

Encapsulamento e isolamento da lógica

Funções que:

Ficam dentro de outras funções e

Usam variáveis atribuídas a elas

def calcular\_volume\_area\_base(area\_base):   
 constante = 3

def calcular\_volume\_piramide(altura):   
 return (area\_base \* altura)/constante   
  
return calcular\_volume\_piramide

def calcular\_volume\_area\_base(area\_base):   
 constante = 3   
 def calcular\_volume\_piramide(altura):   
 return (area\_base \* altura)/constante

return calcular\_volume\_piramide   
  
# closure   
  
piramide = calcular\_volume\_area\_base(15)   
# area base piramide(2) # altura

Decorator

Closures que podem ser inseridas no código sem alterações estruturais

Uso do "@"

@staticmethod

@classmethod

class Bolo:   
 peso = 0   
   
 def \_\_init\_\_(self, peso\_inicial=0):   
 self.peso = peso\_inicial

def atribuir\_peso(self, peso\_a\_atribuir):   
 self.peso = peso\_a\_atribuir

@staticmethod

def calcular\_porcoes\_por\_ovo(numero\_ovos):   
 return round(numero\_ovos \* 3)

Olá Nesta aula a gente vai conversar um pouquinho mais sobre um tema que é um pouco complexo mas é muito importante para desenvolvermos algoritmos que realmente resolva os problemas das pessoas que acompanha enchimento laje por gentileza a ideia aqui imagina só quando Nós criamos algarismos geralmente nós queremos resolver problemas problemas que são na vida real seja por exemplo prevê comportamento seja até criar um site serve para alguma coisa a criação de um algoritmo e quando serve para alguma coisa se a criação de um algoritmo nosso basicamente precisamos isso em alguns casos transporte conceitos da vida real em código apresenta Internet Banking Internet Banking temos Correntistas na tem conta corrente tem clientes pelos gerentes e todos esses fatores esses conceitos precisam ser transpostos em código para nos ajudar nesse sentido existe um conceito não paradigma chamada programação orientada a objetos é um pouquinho lente daquilo que nós estamos acostumados Principalmente quando começamos aprender a desenvolver códigos mas saber o que se trata a programação orientada a objetos é muito importante para realmente garantirmos que estamos conseguindo algoritmos possam resolver problemas da vida real beleza

ir para falar de programação orientada a objetos muito importante é até legal né conversar mais um pouco sobre exemplos a vida real entendermos o que que nós desenvolvemos até o momento e tipo de hábitos nós temos no mundo da programação e saber o que que é necessário para realmente transforma esses algoritmos para programação orientada a objetos então Egito isso me acompanha nos slides por favor Imagine que até agora isso se você desenvolver algoritmos mais simples Até o momento foi basicamente utilizando uma programação estruturada nesse paradigma então ele é um pouco mais simples Imagina que uma analogia como sendo uma receita de bolo receita de bolo a possui começo meio e fim possui até as veias no começo mini coração do que é necessário para fazer aquela receita Então os ingredientes que você precisa para fazer aquela receita e depois você tem aí um modo de preparo até no final se chegar nesse bolo Então até utilizamos isso em vários cursos essa analogia da receita de bolo para começarmos a falar de programação mas especificamente programação estruturada Então até com essa analogia da receita de bolo nós trabalhamos com funções estruturas de seleção repetição entrada e saída de dados O que seriam esses exemplos aqui em até traçando Paralelos em Python com essa receita de bolo entrada e saída de dados aquele input né aquele print pegando todas as colunas a desculpa todos os variáveis e nós precisamos para um algoritmo seja por exemplo o nome o salário de alguém é o peso altura eles exemplos bem simples de algoritmos que nós fazemos Enquanto estamos aprendendo programação de uma forma geral é realmente a esse seriam exemplos de entrada e saída de dados pensando na loja Gerais que tem de bolo seriam todos os ingredientes que nós precisamos para começar a fazer a receita depois disso quando temos por exemplo tomadas de decisão nessa receita de bolo a se a temperatura estiver cima de x faz tal coisa se a o bolo né começou a passar o começou na uma certa aparência faça outra coisa tudo isso são exemplos de estruturas de seleção no wi-fi wi-fi wi-fi que nós também trabalhamos em programação a preciso repetir um passo várias vezes bata massa a tela ficar homogênea na que o exemplo de classe receita de bolo ou em programação é um fórum do Uai então essas estruturas que são de reflexão e também por último temos funções que são aqueles pequenos na conjunto de códigos que são independentes entre si e gente consegue a chamar nvc utilizar se necessário for exemplos de funções que já existem no Python é o próprio ABS

sentido então lembrando tudo que nós fizemos até agora provavelmente em Cachoeirinha e programação estruturada tem um começo meio e fim sempre roda que é algoritmo como falei com começo meio e fim executequery algoritmo Terminei o trabalho e é isso aí por outro lado isso não necessariamente endereça todos os casos da vida real por exemplo quando nós falamos de programação orientada a objetos tem casos que não necessariamente tem somente um começo um meio e um fim pega não é o caso do Internet Banking quando hoje você abrir um aplicativo internet banking não necessariamente você fará exatamente a mesma operação que estará amanhã depois de amanhã hoje às pode abrir o aplicativo pedir para exemplo a segunda via de um cartão amanhã se pode fazer uma transferência depois de amanhã pode fazer uma transferência e um pagamento então não acho que são ações que nós fazemos em diferentes casos diferentes oportunidades pega também um caso de um e-comerce um site que vende produtos hoje você pode fazer uma ação que adicionar produtos uma lista de favoritos tempo de amanhã se vai escolher produtos e adicionar ou remover no carrinho daqui 5 dias você pode comprar alguma coisa então são novamente ações que você faz ao longo do tempo não necessariamente sempre a mesma operação então imagino a o caso da receita de bolo receita de bolo na sempre começamos com água temos é um começo meio e fim é sempre aquele começo meio e fim esperado aqui não Então temos ações diferentes que fazemos Aí dependendo do dia representando esse paradigma nessa forma que temos hoje de fazer qualquer coisa né Qualquer ação não necessariamente que hoje você vai fazer enquanto pessoa mesmo é a mesma coisa que você fará amanhã então sempre são ações diferentes não é sempre que são coisas diferentes acabam acontecendo também a mesma coisa se aplica em programação de um para isso eu não saio paradigma de programação orientado a objetos e também ela nos Progressos segurança e confiança porque segurança porque confiança na Imagine que comprou G orientada a objetos Nós também conseguimos estabelecer papéis então de novo lá no exemplo de Internet Banking você pode ser um cliente que vai ter determinadas ações ou vai conseguir fazer algumas coisas que não necessariamente um gerente Daquele mesmo banco vai conseguir fazer ou também você não sabe mexer conseguir fazer todas as ações que um outro tipo de cliente vai conseguir fazer então com isso a gente consegue segregar né de diferentes tipos de acessos ações e até objetivos na resultados dessas ações com esse tipo de paradigma quem também naturalmente isso traz na segurança Você não quer a partir de certas informações que você fornece entender que outras pessoas também possam mexer na sua conta né ou também a garante quando você clica lá no botão né de um carrinho de compras em um site e direto para fazer ultrassom que mudar o seu endereço ou finalizar compra quando você clica no botão carrinho de compras que ir para o carrinho de compras quando se quer ir lá finalizar sua compra só que se só consegue finalizar sua compra se você escolher alguma coisa então novamente com esse paradigma a gente consegue possibilita alguns dicionários e se para de irmã não é o único ou seja aposto não necessariamente utilizar programação orientada a objetos para resolver isso não entenda que programação orientada a objetos ela vai resolver todos os problemas do mundo ou que todos os desenvolvedores com muito tempo de experiência utilizou-se paradigma é somente uma outra forma de endereçar nos resolvermos problemas no dia a dia beleza

então Existem algumas funções que são específicas desse mundo de programação orientada a objetos que eu gostaria que você entendesse eu acho que até olhando um pouco mais sobre essas noções e conceitos se vai começar associar com até outras tecnologias outras técnicas existem aí nesse mundo de tecnologia da informação e acompanha os likes por favor uma das coisas que você pode perceber enquanto começar você vai trabalhar com programação orientada a objetos é uma similaridade comprar exemplo SQL ou seja bancos de dados no Excel a estruturados você vai perceber por exemplo banco de dados que trabalha aí conhece Kelly eu passei tabelas dentro dessas tabelas eu vou ter colunas dentro dessas colunas eu posso guardar informações não necessariamente uma linha desse banco de dados vai ter as mesmas informações da outra né eu posso identificar de uma forma única cada uma dessas informações na base de dados no banco de dados e também a mesma lógica posso acabar ficando em programação orientada a objetos ou seja eu posso também conectar um banco de dados que instalar em SQL E amarrar ele é um código Python o papel de linguagem de programação C Sharp C + + Java são exemplos mais clássicos né e trabalham com programação orientada a objetos e aí com isso eu posso representar é aqueles objetos elas informações que estão na baixa idade é aquelas linhas eu posso colocar isso dentro de uma linguagem de programação isso porque todas elas acabam aí respeitando alguns conceitos do ML é uma linguagem unificada de modelagem Como o próprio nome já diz e isso aí nos ajudam nessa representação de objetos de atores em 1 na vida real ou seja nas empresas da mesma forma aqui em SQL pensando em um banco de dados de um banco na internet banking pode ser uma tabela lá só lista os gerentes você é uma tabela que vai estar os Correntistas eu posso ter uma tabela de realizar as transações eu posso pegar essas diferentes tabelas E convertê-las aí em objetos da em classes dentro de um paradigma de programação orientada a objetos e sua ajuda te responder aquela pergunta sim tá eu estou criando banco de dados porque eu tô criando uma tabela G gerentes porque eu tô querendo uma tabela decorrentes de transações para quê que eu vou utilizar isso como que isso se amarra em programação de Fato né seja um site um aplicativo então é esse paradigma de programação orientada a objetos ajuda a endereçar essa ponte na responder essa ponte não somente se existe também a uma área que se chama arquitetura corporativa também ajuda a entender o nível um pouco maior como que diferentes sistemas como que diferentes processos de negócio acabam se amarrando então também esse paradigma auxiliar nesse sentido veja que eu falei muitas vezes a palavra programação terminar programação orientada a objetos mas nunca me aprofundei no objeto então que seriam esses objetos imaginei que objeto ele é uma coisa alguma alguma coisa da vida real é que vou deixar o termo coisa né realmente bem bem aberta por que um objeto pode ser o óculos que eu estou utilizando É pode ser uma bancada pode ser uma mesa imagina que objeto é um objeto vamos supor que você tenha exatamente o mesmo óculos que eu estou utilizando agora na verdade estamos com óculos no mesmo modelo esse óculos que eu estou usando agora não é o mesmo óculos que você está usando agora podem ser do mesmo fabricante mesmo modelo fabricado em conjunto mas esse objeto aqui ele é específico ele é único ele é próprio da mesma forma que o seu óculos se eu tenho um copo de plástico e você é um copo de plástico que Na aparência são exatamente a mesma coisa que nascem o meu copo e outra é o seu copo não são objetos diferentes em programação para manter exatamente essa mesma forma de pensar tá então são coisas da vida real Como eu falei esses dois objetos sejam óculos o copo a uma lâmpada não são a mesma coisa então ela se não ocupam o mesmo lugar ao mesmo tempo se você pegasse o seu óculos O seu cofre e colocasse junto com meu ainda se eles não compraram um mesmo lugar ou seja são de fato objeto sair Independência esses objetos vejo que eles possuem estados então óculos ele pode ter estado sujo Limpo aberto fechado um copo ele pode ter o estado às vezes Limpo sujo cheio Neo com Charles Darwin Qual que é o líquido que tem dentro desse copo então o qual que é a cor desse copo veja que cada objeto ele tem uma característica se possível parar um pouco para pensar em objetos tem mais próximo de você pode ser um teclado mouse uma lâmpada uma porta qualquer coisa quais estados quais o melhor Quais características

quando falou características são coisas que eu definir Então imagina a porta qual característica que deve ter aquela porta aquela porta pode ter um tamanho ela pode ter um material né que foi utilizado na construção dela ela pode ter uma cor imagine de novo né o óculos a qual que é o fabricante daquele óculos só que é o tamanho daquele óculos a cor Então são características que define ou coloco né Qual que é o tipo de plástico que é utilizado a qual que é o tamanho daquele copo é qual que é a cor daquele copo então são novamente características que o define ou imagine pessoas né O que que definiria o que que distinguiria uma da outra se Estamos por exemplo em um lugar comendo alguma coisa restaurante às vezes só o nome é uma característica que vai na diferencial no outro às vezes para Receita Federal vai ser muito mais do que isso vai ser no CPF data de nascimento nome dos pais então Depende muito do contexto é Quais características se nós vamos ter desses objetos e também dos Estados quando eu falo estado é algo que pode mudar Aí dependendo de uma ação Então tem um óculos óculos eu posso tirar ou colocar no rosto eu posso abrir e fechar ele uma porta também não posso abrir e fechar uma porta um copo né eu posso pegar ou colocar em cima de um local então também imagine esses estados como como aquele objeto está nesse momento e o que que pode ser feito para mudar ele então é importante dessas diferenças aqui algumas comparações que podemos ser que então tanto das características e afins dos Estados eu posso atrelar ele de certa forma com os atributos que podemos comentar no mundo de inteligência artificial na linha chilena ou também essas variáveis e também esses comportamentos alegações ou funções né E também a gente pode falar partir de agora em métodos em linguagens de programação que é representar o objeto copo eu quero beber ele tem uma ação chamada beber posso transformar essa ação em uma função em linguagem de programação quando nós começamos a do Mundo desse paradigma de programação orientada a objetos a gente deixa de ser ferimento como funções e as funções essas ações que ocorrem em cima de objetos acabamos chamados de métodos e tá importante para saber essa mudança de nome então a invés de funções a gente começa a ser isso muito mais como métodos dentro desse paradigma as classes Imagine que as classes elas são aí manual de instrução no seu grupo entrou um molde que nós poderíamos fazer para fabricar objetos de novo exemplo dos óculos o óculos a classe que apresenta o óculos seria um manual de instruções de uma fábrica como que nós construímos sair 10 óculos desse mesmo modelo a multa de um jeito faz de outro tem que ter no mínimo é uma cor tem que ter um tamanho Então são são definições que nós utilizamos para criar objetos outro exemplo que que é o dono imagina que o objeto é o bolo a classe a receita de bolo então para ter bolo Mas precisamos ter a receita de bolo para ter um objeto preciso de uma classe que defina aquele objeto não é legal essa essa comparação a imagem outro caso também é uma classe seria a planta de uma casa e já o objeto seria a casa esse Então veja de novo a classe não seja exatamente a planta não sei não que seja necessariamente a casa já construída mas sim o que vai garantir que nós possamos construir 5 10 20 casos exatamente aquela forma a física né que se pode entrar você não pode espera-se que não né que você não coma a receita de bolo mas assim que você coma o bolo produzido não espera-se que você entre que você mora dentro de uma planta de casa mas sim que você mora na casa construída tá então é importante dessa diferenciação e outra forma que nós nos referimos aqui é que toda vez que nós criamos um objeto a partir de uma classe nome disso é instanciar então eu quero criar um objeto a partir de uma classe instanciamos uma classe e aí se Nós criamos um objeto exemplos que também trago aqui imagine uma classe chamada cachorro que tipo de método é atributo nós podemos ser que é um método que tipo de ações o cachorro pode fazer todo o cachorro ele pode latir ele pode comer para ir dormir pode mexer na vários outros métodos várias outras ações poderiam ocorrer aqui e também nós podemos ter alguns atributos eu posso ter dentro de uma casa de cachorro Esperamos que todo cachorro ele tem uma cor e aparecer um nome ele para ter o brinquedo Favorito Eu também pode imaginar vários sexo e outras características que possam ajudar definir todos os cachorros que existem dependendo desse contexto então nós temos de novo a diferenciação entre métodos e atributos Isso define todos os cachorros aqui quando nós instanciamos o cachorro eu posso ter objetos tem o que que seria o objeto com H é um cachorro chamado o que é uma cachorra é totalmente diferente de outro cachorro chamado Pipoca é diferente de outro cachorro chamado Dexter Eu tenho três cachorros diferentes poderia estar correndo aqui e aí eu posso ter na realmente esses objetos instanciados eu posso financiar 1023 cachorros e cada um deles terão essas esses atributos não é esses métodos que nós vimos aqui seria cor nome brinquedo Favorito e assim por diante Outro ponto que também gostaria de comentar né que são as heranças quando falo herança herança de uma casa Face imagina se eu posso ter uma classe que ela é bem genérica Eu passei algumas especialidades dela então como você deve imaginar eu posso ter subtipos na de cachorros então posso ter às vezes um cachorro que pode ser de estimação Mas nem todo mundo entende um cachorro de estimação da mesma forma você é um cachorro que ele é utilizado para exposição outro que é utilizado como um cão de guarda outra que é utilizado para corrida para eventos e assim por diante então como eu tenho eu posso ter a diferente entre aspas subtipos nas aplicações dessa classe ou classe especializadas então o nome disso seriam herança essas heranças elas eram os métodos e atributos Mas eu também posso incluir novos métodos novos atributos que são específicos é aqueles exemplos então outro caso né Imagine que eu tenho lá uma uma classe chamada carro então se como se deve imaginar essa classe carro ela pode ter em suas seus métodos ela pode ter aí seus atributos já definidos aí dentro dessa dessa classe porém não necessariamente os mesmos métodos atributos do carro com certificados para todos os carros existentes no mundo então eu passei aí uma herança de carro chamado caminhonete caminhonete eu posso ser às vezes uma novo método que é só da caminhonete que seria por exemplo colocar uma carga é uma carga maior que é um carro comum seria comum não seria da esperado Pode ser às vezes num furgão não é m ou possibilidades atributos que são específicas do furgão e não necessariamente de qualquer carro a mesma forma da minivan Então veja que cada um desses objetivos de carros eles podem ter suas particularidades nós conseguimos ser essa mobilidade que nesse sentido caminhonete furgão mini Gun tradução carros né eles eram de carro mas ainda se eles podem estender isso eles podem ter esses atributos a mais outro exemplo aqui na próxima bicicleta toda a bicicleta ela tem nessas atributos seus netos eu também posso ser tipos especializados de bicicletas BMX né que é uma bicicleta específica para ser bicicleta de carga bicicleta de pista de velódromo sefins então cada uma dessas bicicletas elas vão atender nichos diferentes necessidades diferentes e naturalmente aí eu vou ter netos que são específicos a eles uma bicicleta de carga por exemplo espera-se que ela deve carga Mas é uma bicicleta de pista então uma bicicleta eu posso pedalar qualquer uma das três bicicletas mais carregar produtos que é preciso transportar talvez tá na bicicleta de carga tem um beijo que eu tenho aqui essas possibilidades também outros cenários é residência eu passei subtipos de residências posso ir aí na residência do tipo sobrado outra que é totalmente diferente de uma casa de praia que é totalmente diferente de um apartamento no caso da residência eu posso selar um método específico do Sobrado que seria aparar grama se tivesse um gramado para um apartamento que não é no térreo não faria muito sentido aparar grama não faria sentido ter esse mesmo mas eu também passei para qualquer tipo de residência e método chamado entrar na abrir a porta e entrar na sua residência Então veja novamente eu tenho coisas que são em comum para todos esses objetivos de residência mas eu também posso criar alguns exemplos em específico veja aqui sempre tô colocando três subtipos específicas mas você não é restrita isso os pais de teclado que vão ter 10 heranças uma duas dessas heranças E aí depende de cada cenário também nós podemos nem todo mundo seria um único tipo de cliente eu posso ter tipos diferentes de clientes Internet Banking eu passei um cliente Ele Me Ama com outra mais simples esse cliente ele vai ter algumas ações que ele pode fazer que um cliente O que é Premium e vai ter muito mais coisa ele pode ter na imensa várias ações adicionais e também esses diferentes tipos de clientes eles podem ter dados de dados que são diferenciados atributos que são específicos a eles que não são de uma conta Empresarial e vice-versa exemplo mais simples que nós podemos pensar que é o o próprio identificador então uma conta Empresarial só ela vai ter um atributo é chamado CNPJ a conta simples o prêmio que seria uma pessoa física não contei isso eles vão ter como atributo ou CPF então aqui eu botei uma particularidade de cada um desses tipos Então é só para você entender alguns cenários que temos disponíveis aqui também né com exemplos práticos de métodos atributos lembra lá do cachorro não tenho método latir comer dormir se eu tenho uma classe chamada cão farejador um cão farejador e ainda sim é um cachorro mas esse cachorro ele vai ter métodos adicionais um cão farejador ele também vai late cão farejador ele também vai comer ele também vai dormir mas Além disso ele pode fazer jar e ele pode sentar aí alguma coisa ele senta então também né Nós podemos ter atributos adicionais que são específicas somente do cão farejador o cão farejador ele também tem cor com pelejador ele também tem nome ele também tem um brinquedo Favorito que um cão farejador antes de mais nada é um cachorro mas esse cão farejador ele tem coisas a mais tem atributos a mais que só dele teremos essas informações por exemplo tempo de treinamento é um cão farejador de preciso ser treinado para mim sempre a sua função e qual era o porto ele pode tá trabalhando nessa um cão farejador que a tua aí para detectar objetos legais e também a raça então às vezes um cão farejador nós precisamos saber a raça dele para um cachorro normal né em geral as vezes essa informação é necessária então aqui é um exemplo que você deixa trazer para você para entender qual que é essa diferença entre classe você ver como que isso se aplica beleza por último gostaria de falar contigo sobre interfaces veja aqui as classes elas podem utilizar suas interfaces as interfaces elas são como se fosse assim um manual um pouco mais simples e que uma classe que ela deve ter Então até pensando lá na classe cachorro quando a gente cria uma classe chamada cachorro se tem que criar o código falando que se espera quando alguém chama o método latir que que o cachorro deve fazer quando Nós criamos uma interface a gente somente fala assim olha se algum dia alguém quiser criar uma uma classe de um animal qualquer animal que seja olha todo animal tem que dormir então você cria uma interface chamada animal lance olha todo todo e qualquer classe que vem a utilizar-se interface chamada animal Ela tem que ter pelo menos esse método aqui né seja dormir seja comer alguma coisa do gênero mas como vai ser essa forma de comer e dormir vai depender de animal para animal pode ser uma classe chamada cachorro ela vai utilizar sem interface s no cachorro ele vai comer do jeito dele que é diferente da forma que um humano pode comer diferente da forma de uma tartaruga pode comer e assim por diante então nós que essas interfaces somente fala Olha tem que ter assinatura aqui mas não o código que eu utilizo né Essa que possa utilizar essa essa classe interface então por isso que deixou bem claro esse ponto que as interfaces não implementam m a gente não coloca o código para isso beleza agora como seria um assistente mentha sões de classe em Python é porque até o momento nosso somente vimos exemplos mais mais teórico né alguns exemplos na vida real mas até agora nada de Paiva tem como que isso seria na prática e acompanha aqui no slides por favor primeira coisa da mesma forma que quando criamos uma função a gente senta que cria nós utilizamos Chile Death naquela palavra-chave Death depois o nome da função e aí os parâmetros que fazem parte dela em uma classe É bem parecido então ao invés de nós utilizarmos DF para definir uma função utilizando o classe para definir uma classe tem um cílio cresce o nome da classe que nós precisaremos né Que Nós criamos e dentro dessa classe temos os atributos os métodos e uma palavra chave que é muito legal de você conhecer essa selfie selfie basicamente garante que nós conseguimos referir na criar atributos ou métodos que modifiquem atributos na arte mostre quem é o estado realmente o que acontece dentro de um objeto né dentro de uma classe Então esse selfie nos ajudam muito nesse sentido na prática imagina então que nós estamos criando uma classe chamada bolo e nesse desafio não é o melhor nessa criação não somente temos um M então m é basicamente assim estamos querendo um bolo que a gente quer definir o peso daquele bolo só isso então novamente é um bolo de um objeto bolo que nasceram no Ceará queremos criar vários bolos e para cada um desse bolo para cada um desses bolos nós queremos definir o peso só isso olhando aqui no código q que nós temos veja que primeiro temos a definição da classe da Lacraia espaço bolo até veja que cresce né ele tem azulzinho porque realmente é uma palavra protegida do pai tá uma palavra utilizada pelo próprio pai tá on igual Davi e também tem Azul então primeira linha o Class bolo lá então estou definindo uma classe chamada bolo de noiva Imagine que a sua definição de classe e começou a ser uma receita de bolo e não bolei esse então estamos criando a nossa receita depois definimos os atributos que fazem parte daquele bolo no caso peso aqui por padrão está zero né então todo bolo por padrão nesse exemplo específico ele começa com zero e aí Nós criamos um método para ele esse único método e Claros faz criadores 320 m do cemitério foi para uma classe A que nós temos o método chamado atribuir peso dentro desse método chamado atribuir peso desde que temos dois parâmetros o primeiro ao self.exe selfie que eu comentei antes com esse selfie nós basicamente o mesmo sem entender assim quando criarmos um objeto toda vez que nós chamamos esse atribuir peso estaremos atribuindo peso daquele objeto específico se nós não colocássemos de se atribuir peso o Patolino ele entender que nós estamos tentando atribuir o peso de um objeto instanciado de um bolo e sim que estamos chamando uma função qualquer que vai ter uma entrada e vai ter uma saída e não que novamente modificar a aquele objeto específico e o segundo parâmetro é o peso atribuir informamos Qual que é o peso que precisamos atribuir E aí sim ele muda para o peso específico veja aqui nessa quarta na quinta linha temos o selfie ponto peso que significa esse selfie ponto peso o selfie ponto ele está falando sim nós queremos alterar o peso daquele objeto específico então se eu não colocar esse selfie ponto eu não conseguiria na prática mudar o peso de um bolo em específico é nós não conseguiremos realmente alterar né aquele peso e saber o que que não quer dizer então nós nunca conseguimos mudar aquele valor daquele atributo desde aqui no final nós temos aí a criação de um bolo então estamos criando um objeto é chamado o bolo aniversário e nós estamos com aquele bolo abre e fecha parentes cri criando um novo bola nesse sentido é bem na salinha que nós estamos fazendo aquele instanciamento naquela instance estamos criando uma Instância de um bolo é bem na salinha que entre aspas a mágica acontece então criamos um bolo um objeto né de bolo aqui o bolo aniversário e depois iria sugerir se possível pega esse código implemento na no seu próprio computador e veja o que que vai acontecer se nós fizermos esse print bolo aniversário ponto peso nós teremos como resultado zero porque geralmente nós não mudamos o peso até agora só queremos um bolo e o bolo por padrão quando você viu lá no começo na segunda linha propaganda e tem o peso zero então por isso que ele sai dessa forma agora se depois fizemos o bolo aniversário ponto atribuir peso 2 bolo aniversário específico se eu tivesse criado o outro 5 bolos diferentes bolo aniversário bolo festa no fim de semana né bolo qualquer outra coisa se eu fizesse o bolo aniversário quando atribuído peso 2 eu somente a tribo de Mc peso ao bolo de aniversário todos os outros bolos não seriam afetados Então não é importante entender esse conceito e se fizermos esse último print veja que se print bolo aniversário ponto peso é igual ao print anterior então os dois nação preenche o bolo aniversário ponto peso porém ao fazer esse S print ou será que aí sim teremos o peso atribuído de 2 ou seja nós conseguimos mudar o valor do Peso na que por meio desse método o outro negócio bem legal de você saber que nós conseguimos inicializar uma classe com esse Def init veja que esse nicho ele tem dois underline SO2 Andreas códigos no começo e dois andares depois do início isso porque esse nit padrão no pai o pai então ele tem essa isso específico outras linguagens de programação inicializam as classes de uma forma diferente mas a lógica é a mesma que serei inicializar uma classe se nós olharmos de novo nesse exemplo aqui vem aqui o peso ele sempre por padrão tá zero e se nós quiséssemos já criar um bolo um preso padrão a gente também consegue fazer isso como com início então é para isso que serve esse início estamos aqui modificando o código e agora sim colocando início vejo que temos agora né aí se define e temos esse peso Inicial O que que eu gostaria que você olhasse tá esse definity então na terceira e quarta linha ou na quarta e quinta linha se considerarmos uma linha em branco verde que tem lá no Def init o selfie se mantém E agora temos um parâmetro chamado peso Inicial e até temos um peso Inicial = 0 que que significa esse peso Inicial igual a zero é igual a zero ele quer dizer que nós estamos atribuindo que podemos atribuir um valor Se quiser tem um bolo abre parentes 2 fecha parentes 2 ele já iria criar por padrão o peso Inicial não é igual a 2 mas se não informar Simus nada o peso Inicial por padrão ser esse 0 então virão ao opcional né a gente pode me informar esse peso Inicial Se quisermos se não quisermos por padrão e 60 e aí com isso a gente consegue rodar todo o código depois o que que eu sugeriria para você pega esse código tanto executado seu computador né deixa como ele se comporta e vai mudando esse peso Inicial O que que acontece se aquele bolo aniversário igual bolo ali fecha parentes se naquele abre e fecha parentes você me informasse 5 que queria conhecer com código ou se você pegasse o peso Inicial = 0 nessa terceira quarta linha odacy peso Inicial igual a 5 ou remover sol igual a zero o que que iria acontecer com código então realmente é como sugestão mudando a esses diferentes valores é muito provável que só entender um pouco mais na prática Como de fato funcionam aí essa atribuição né padrão a uma classe como que nós conseguimos instanciar ela e também costomizar alguns dos seus atributos iniciais beleza um outro toque também é bem interessante de conhecermos um pouco mais sobre polimorfismo na palavra um pouco mais complexa mas somente o conceito dela é bem simples para isso me acompanha nos slides por favor o polimorfismo é basicamente assim nós podemos ser um método Ou seja eu não falo um vírgula imagina se um nome padrão geométrico lá no exemplo daquelas cachorro não para de latir ou ainda não exemplo do bolo poderia ser assar ou próprio atribuir peso eu posso ter então um método com esse nome né que eu dei alguns exemplos implementado algumas vezes diferentes tem um passeio na verdade três quatro funções com o mesmo nome eu posso ter cinco funções diferentes chamadas exatamente assar bolo eu posso ter por exemplo 10 m diferentes da que é um exagero é chamado o cachorro latir então eu posso ter realmente diferentes implementações e seja repetições de um único nome de método mais dentro de cada uma dessas implementações melhor dentro de cada uma dessas funções e passei um código diferente então a premissa que básica do polimorfismo nesse cenário é basicamente a ciência pode ser como fazer 35 implementações diferentes de assar bolo mas o que vai diferenciar cada uma dessas funções diferentes cada um desses métodos diferentes serão os parâmetros então por exemplo eu posso ir lá um assar bolo que vai ter a temperatura e vai ter o tempo e a temperatura ambiente porque às vezes tá mudando a temperatura ambiente Ou posso também tem uma ação diferente para assar ou eu posso ter por exemplo E se eu tivesse disponível chip da Travessa que eu estou utilizando ou não é a pressão atmosférica que pode alterar um pouco ou ainda não é o tipo do forno que estou utilizando Então dependendo desses parâmetros se nós informamos nós também podemos ser implementações diferentes formas por exemplo Nessas horas sai um bolo quanto mais informações você possui em suas mãos pode ter as suas diferentes a posso assar um bolo Tá mas qual que é o tipo do bolo Qual que é o tamanho é quanto tempo que eu tenho Qual que é o forno que tem disponível tem muito gasto nem vou usar um forno a gás e elétrico Então eu tenho que ações diferentes e compare meu filho nós conseguimos implementar isso né considerando como fazer esses parâmetros de entrada diferentes também com esses parâmetros de entrada diferença para ter resultados diferentes ou soltar um pouquinho lá no exemplo do cachorro às vezes um cachorro de uma forma genérica tem o método latir M Launcher de um cachorro sem uma implementação eu vou ter uma determinada determinado resultado agora o cão farejador latir dele Pode ser que seja Beta mente diferente Ou ainda né imagina o Internet Banking pedir atendimento de um cliente e se pedir atendimento de um cliente comum pode ser completamente diferente do atendimento de um cliente por e-mail de um cliente Empresarial Então veja aqui estou falando a mesma coisa que pedir atendimento mas a forma que o atendimento ocorrerá ou a forma de atendimento será conduzido Depende muito do cliente então a lógica que é mesmo veja que estou falando de novo em uma ação que na teoria Seria a mesma mas a forma que acontece muda muito dependendo da classe ou dos parentes que nós temos aí disponíveis exemplo aqui naquele domingo cereja trazer imagine de novo veículo se eu falar para você estaciona o veículo na sua cabeça às vezes você vai ter em sua mente ou o carro que você postou disponível ou não é o carro de um vizinho vai ter assim o exemplo mais comum que você tem aí no seu no seu Imaginário Às vezes o que eu imagino para estacionar seja algo também mais próximo da minha realidade mas olhando aqui esses exemplos de slide vejo aqui que são casos que na prática mudam a forma que a gente vai estacionar um carro ou local que a gente consegue estacionar o carro é muito diferente no local tipo consegui estacionar uma moto ou até a forma né que nós vamos estacionar uma moto é bem diferente de um caminhão ou de uma bicicleta e assim por diante até porque uma bicicleta dependendo do caso nós podemos estacionar dentro de uma casa ou na terra colocar em um apartamento já é um caminhão isso não seria muito fácil ou até mesmo seria possível na prática Então veja aqui de novo estamos falando sempre da mesma são sempre do mesmo método estacionar agora a forma que nós vamos estacionar tanto lá da classe veículo ou estacionar o veículo pode ser uma forma muito genérica conforme nós vamos aqui olhando as diferentes implementações lógica completamente diferente então é isso que eu gostaria de deixar claro para você enquanto nós estamos falando de polimorfismo nesse conceito de programação orientada a objetos e agora eu gostaria de falar contigo sobre outras vezes temos que ele se conversam e acredito que também passa um ser bem outra no seu desenvolvimento de bons algoritmo utilizando o paradigma de programação orientado a objetos seria aqui no caso os códigos e os de corais e acompanhe por favor nos lares quando nós falamos aqui em loja de corretor importante primeiro falarmos sobre encapsulamento Então imagina né que quando estamos falando de encapsulamento eu posso trabalhar em Python e claro com outras linguagens de programação com funções dentro de funções então eu posso não pode parecer um pouco confuso Caso seja não tem implementado ao desse tipo mas eu posso ser realmente funções que dentro dela eu consigo como eu falei de uma forma encapsulada ter outras funções então eu posso ter na até conjuntos de código que eles são realmente encadeados isso nos dá duas vantagens primeiro redução de casos de variáveis globais quando fala uma variável Global às vezes para se conseguir transportar na variáveis ou informações de uma função para outra você acaba tendo que utilizar uma variável Global algo que assim outros outras funções outros objetos outras classes podem atrair contato e isso para não ser muito legal do ponto de vista de segurança é do ponto de vista de otimização no seu código então por isso que não é muito legal nós utilizarmos as variáveis globais A não ser que tenhamos um bom motivo para isso e finalmente temos aí eu caso de encapsulamento né começo comentei isolamento Da Lógica Ou seja quando nós estamos falando aqui do paradigma de programação orientado a objetos é importante ter nosso código bem é compartimentalizado ou seja bem ser ligado né bem dirigido bem organizado em diferentes blocos até pensando nessa organização da seriação no Gene que toda que acontece dentro de uma caixa toda lógica que se aplica dentro daquela classe como realmente fosse uma caixinha é específico é aquela classe e aos objetos que são influenciados por ela a mesma coisa também né em outros em outras funções outras implementações que possamos fazer aí nesse sentido então como é importante termos desse isolamento Da Lógica pensando até em manutenção uso no dia a dia importante termos isso de uma forma bem independente para garantir energia nessas implementações uma não acaba afetando a outra então às vezes você acaba tendo um código muito complexo isso não é muito discutido dependendo do tópico de programação orientada a objetos que algoritmos construídos nesse paradigma acabam tendo aí um grande conjunto de código então você acaba tendo é o chão de uma rede né realmente uma grande quantidade de linhas a mais e você não criaria se utilizasse outros paradigmas isso sentença os prós e os contras um dos motivos pelos quais criamos muitas linhas de código a mais realmente para garantir esse isolamento nessa Independência entre diferentes classes e objetos então isso naturalmente pode nos ajudar por lá do trabalho né que temos um volume juninas de código muito maior prejuízo entrar mas também ajuda porque se precisarmos alterar alguma coisa diminuir um pouco o risco de quebrar mas na outra sessões aí do código que nem acabavam nem chegamos aí prestado atenção ou conhecendo no dia a dia Então nesse sentido é por isso que falando de encapsulamento para falarmos de Cláudio Lins essas coisas aí são funções Então primeiramente cor igual função elas são aquelas que ficam dentro de outras funções e usam aquelas variáveis atribuídos a elas Então nada melhor do que mostrar um exemplo veja aqui eu tenho uma função dentro de outra função eu preciso a partir de uma área da base eu quero calcular é o volume que seria especificamente de uma pirâmide veja que eu tenho um Death silence calcular volume pirâmide é que está aí em volta por esse um retângulo que tem aí uns cantos arredondados mas nesse caso aí que foi Saiu uma linha laranja e essa função interna ela está dentro de uma outra função chamada calcular volume e área base Então essa função interna essa parte que está realçada nesse retângulo que não é bem um retângulo essa função interna é um exemplo de como é isso eu vejo aqui altura né que está em cinza quando nós utilizamos calcular volume pirâmide nós passamos a altura para ela e aí sim ela consegue fazer o seu dever do cálculo tá então é isso é que nós entendemos como um exemplo de implementação de Cláudio até se possível também utilizar esse código entendo né o Como que é o comportamento dessa utilização de Cláudio E como que você implementaria sem utilizar isso né Como que você resolver esse mesmo problema sem utilizar Cláudio e tipo de dificuldade hosteria né Que tipos de variáveis você teria que criar e não estariam dentro de uma função seria uma variável global e prós e contras histeria aí é uma espécie código Não é isso você deseja trazer para você nesse sentido com o uso de Cláudio até também expandido nesse caso veja aqui é usado em amarelo temos a constante essa constante ela é reaproveitada veja que no10 calcular volume pirâmide só temos um parâmetro que se chama altura e Justamente esse né que estava realçado em cinza não temos um seguro do parâmetro chamado constante Ele não tá aí mas mesmo assim nós utilizamos essa variável ou constante Pode parecer paradoxal não é uma variável chamada constante mas ela acaba sendo utilizada dentro dessa função calcular volume pirâmide e ela foi definida pela função entre aspas pai né então por isso que temos aí essa implementação então deixa aqui também utilizamos um exemplo né calcular volume área base 15 o que seria o 15 o 15 é justamente essa área da base Então essa área da base era transmitida para calcular volume pirâmide E aí sim nós temos o resultado aquele mitani calcular volume pirâmide nada mais é do que a função calcular volume pirâmide sendo chamada já com altura e afins E com isso nós temos um resultado em mãos Então veja presencialmente que na utilização do Cláudio primeiro temos o calcular volume área base novamente nós definimos primeiro a base e virou dentro da pirâmide na última linha nós aí se nós definimos essa altura que no caso seria 2 e no final a nós conseguimos ser realmente é o resultado dessa desse cálculo Além disso nós temos o de corretor de corretor ele é um tipo específico de Cláudio Então essas cores é como se diz de corretores eles tivesse um apelido né tem 1 arroba E aí tem uma definição essa palavra chave do decorador ela é utilizada para garantir que o nosso código aí pode ser machucado sem grandes alterações na estrutura quando fala alteração estrutural é realmente pegar aquele código que tem que ficar mudando muita coisa reestruturando ele ela tando ele então isso nos ajuda bastante como eu falei fazemos pequenas alterações ao longo do código sem ter que ficar testando muito acaba introduzindo m b então de corrida ele acaba sendo muito útil nisso veja que temos aqui dois exemplos né então primeiro nós definimos o de correr pelo sinal de Arroba e dois casos que já existiram principais primeiro é o estético message e o segundo é o clássico E se ele é o método que é estático uma das definições que existem dentro de programação orientada a objetos são os métodos que eles são estáticos Ou seja que acabam não tendo o estado é alterado para a assim que acaba afetando a todo o estado de um objeto específico então nós conseguimos também definir certas funções ou chamados e quais poderiam estar definidos como M estágio dessa forma veja lá no exemplo do bolo e também na vimos anteriormente Se eu quisesse criar um novo método chamada calcular porções por roubo até porque estamos falando de bolo aqui né a partir de um número de ovos eu quero calcular quantas porções tem aquele bolo pode vender isso aqui em específico não é que vai ser específico a um objeto tão calcular porções por novo Isso é se aplica a qualquer bolo né até poderia ser não pode dizer o que vai auxiliar a receita do bolo mas não será o específico do bolo então por isso que tá aqui como método estatístico é queremos calcular porções provo mas não que você vai modificar o peso veja que nenhum momento temos o selfie nesse último método nenhum momento nós estamos alterando o peso eu não acho que temos aqui algumas diferenças que são bens essenciais em relação ao atribuir peso por exemplo mais uma vez desde que eu te vi o peso começa com selfie e tem peso atribuir calcular proporções por outro não tem isso porque não precisamos do céu fica a gente não vai alterar o peso em um momento é o mestre estágio então também por isso que nós colocamos esse arroba né No começo para definir esse esse método como sendo de fato estático e é por isso que temos esse arroba Tech do DF então por isso que ele está dessa maneira aqui também importante eu saber que quando criamos os decoradores também pode Joel consigo customizar isso consegui anotar o número de caso de uso e também é possível de adaptarmos a este melhorarmos o nosso código aí frisando esses covers e depois tudo bem

**CONVERSA INICIAL**

Olá! Nas nossas conversas anteriores, comentamos sobre o uso das bibliotecas e, ainda, sobre o trabalho com algoritmos de *machine learning*(ML). Por outro lado, o Python oferece também outras possibilidades de desenvolvimento: a sua versatilidade é uma das razões pelas quais estamos trabalhando com esta linguagem dentro destes estudos.

Dito isso, vamos conversar aqui sobre programação orientada a objetos (POO). Até este momento, é provável que você não tenha um contato profundo com este tema em Python, por outro lado, o uso de POO pode ser bem útil em sistemas corporativos de larga escala, servidores, sistemas de simulação, entre outros. Vamos lá?

**TEMA 1 – UM PROBLEMA PARA SER RESOLVIDO PELA PROGRAMAÇÃO ORIENTADA A OBJETOS**

Quando ingressamos em um curso de Tecnologia da Informação (TI) e aprendemos a desenvolver algoritmos utilizando linguagens de programação, quase sempre aprendemos da mesma forma. Independentemente da universidade, do país ou do professor, (quase) sempre aprendemos na seguinte linha de pensamento.

* **Algoritmos são como receitas de bolo**: possuem começo, meio e fim bem definidos.
* Também como receitas de bolo, foram feitos para **sempre funcionarem de uma mesma forma***:* não importa se você está fazendo a receita de noite ou de dia; se quem faz a receita é um adolescente ou um idoso; se a receita está sendo feita em uma fazenda ou em uma cozinha em um arranha-céu. **Sempre esperamos o mesmo resultado como consequência das mesmas entradas.**
  + Da mesma forma, um algoritmo pode ser executado em um *notebook* ou um *desktop*; um computador antigo ou um que acabou de ser comprado: sempre esperamos que ele produza o mesmo comportamento. Peguemos um exemplo de um aplicativo de *internet banking* de um dado banco. É esperado que uma transferência sempre funcione da mesma forma, independentemente do modelo de celular, da cor do celular ou da versão do sistema em uso.
* E, como receitas de bolo, é relativamente fácil mudarmos alguma coisa no **final** do processo. Por outro lado, se precisarmos mudar algo no **meio do caminho**,teremos problemas. Ou você nunca se deparou com alguma atividade em certa disciplina em que você arrumou algo no meio do seu código só para se deparar com um conjunto de novos erros em outras partes do código?

Este paradigma chama-se ***programação estruturada***. Essa forma de criar algoritmos possui suas vantagens: ao usar esse paradigma, por exemplo, é relativamente fácil começarmos a criar algoritmos simples e de entendermos o que ele faz, na prática. Isso inclui casos como:

* leitura e escrita de dados com base no que um usuário pode digitar;
* estruturas de seleção (como o *se/se-então/senão*);
* estruturas de repetição (como o *para/enquanto*);
* estruturas de dados (como vetores, listas e matrizes);
* estruturas de divisão de código como as funções; e
* também permite que você encontre os erros de forma mais rápida porque ela possui menos *overhead –*isto é, necessita que você produza menos código que serviriam somente para **servir como fundação** para a sua lógica.

Durante esta aula, voltaremos a usar várias analogias (igual a receita de bolo). Logo, se estamos falando em receitas culinárias, a programação estruturada serve como uma maneira mais simples de ensinar a cozinhar para quem nunca cozinhou na vida. Observe o quadro a seguir.

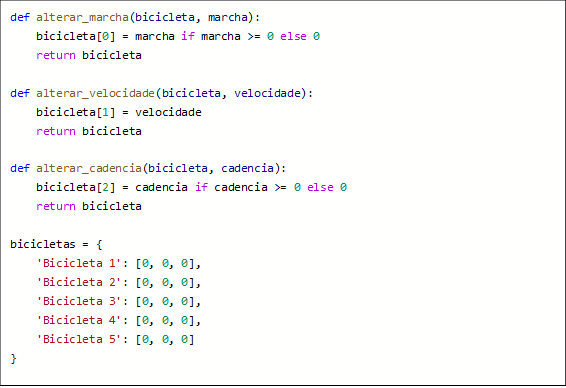
Quadro 1 – Programação estruturada

|  |  |
| --- | --- |
| **Cozinha** | **Programação** |
| Fazer salada | Mostrar um texto na tela |
| Ferver água | Guardar em uma variável o que um usuário digitou |
| Cozinhar um ovo | Usar estruturas de seleção (se/se-então/senão) |
| Fazer omelete | Usar estruturas de seleção com variáveis que o usuário digitou |
| Cozinhar miojo | Usar estruturas de repetição (para/enquanto) |
| Cozinhar arroz | Usar listas, vetores e/ou matrizes |
| Cozinhar carne moída | Usar funções |
| Preparar um almoço com arroz, carne moída, ovo e salada | Criar um algoritmo com várias funções, estruturas de seleção, repetição, variáveis e listas |

Entenda o quadro anterior como apenas formas de **comparar**o esforço entre ambos e não compreenda isto de forma literal. Dito isso, provavelmente, preparar um almoço com arroz, carne moída, ovo e salada não faria de você um *chef* de cozinha, muito menos o prepararia para todos os tipos de pratos possíveis. Por outro lado, poder-se-ia dizer que é a base necessária para preparar pratos mais complexos: cozinhar arroz envolve coisas como gerenciar a temperatura do fogão, saber lidar com os utensílios da cozinha, saber controlar o tempo e os ingredientes necessários, entre outros. Logo, quando fôssemos aprender sobre algo mais complexo, todos saberiam que você já dominaria o básico (e que alguns poderiam até mesmo dizer que seria o “óbvio”).

É exatamente a mesma coisa conosco: primeiro, você aprendeu os fundamentos da programação para então chegar ao momento de trabalhar com paradigmas mais complexos.

Dito isso, vamos pensar no seguinte desafio: você foi encarregado de desenvolver o sistema de software de uma empresa de aluguel de bicicletas. Utilizando sensores e tecnologia recente, a missão é construir um software em Python que registre dados das bicicletas, por exemplo, a troca de marchas, a velocidade e a cadência (a quantidade de pedaladas por minuto). Utilizando todas as técnicas que listamos e com o que já vimos anteriormente, poderíamos fazer algo assim:



Veja que temos um dicionário chamado *bicicletas* e que possui cinco bicicletas. As chaves são referentes ao nome das bicicletas (como *Bicicleta 1* e *Bicicleta 4*), e os valores são uma lista contendo os dados referentes às bibliotecas. Por exemplo:

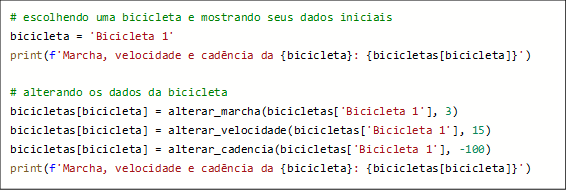
'Bicicleta 1': [0, 0, 0]



Armazena três valores para a *Bicicleta 1*. Sem olhar o código das funções, fica bem difícil de sabermos a que se refere o primeiro valor (ou o segundo, ou o terceiro). Ao analisarmos novamente as funções apresentadas, sabemos que o primeiro valor equivale à marcha, o segundo valor equivale à velocidade e o terceiro valor equivale à cadência.

Aqui vai a primeira dor de cabeça: e se, em vez de três características, estivéssemos trabalhando com 300 características? Em grandes sistemas corporativos, isto pode ser bem comum: imagine que não estamos mais trabalhando com bicicletas, mas sim com notas fiscais ou com dados cadastrais de empresas – já imaginou a quantidade de características que existem? Agora, pense se em algum dia um determinado algoritmo está dando um erro, e você precisa analisar o código. Pense só: se estivéssemos trabalhando com uma lista com 300 itens, como poderíamos encontrar rapidamente **qual é o elemento**que está com problemas, principalmente se nunca vimos aquele código antes em nossa vida?

Para falar da segunda dor de cabeça, vamos pensar em um código no qual alteraríamos marcha, velocidade e cadência da Bicicleta 1:



Ao rodar esse código, temos os seguintes resultados:



Veja que a marcha original (primeiro valor da lista) era 0 e foi alterada para 3; a velocidade (segundo valor da lista) era 0 e foi alterada para 15; a cadência (terceiro valor da lista) era 0 **e permaneceu em 0**. Por outro lado, observe novamente o código para a alteração da cadência: tentamos mudar a cadência de 0 para -100, mas sem sucesso:

bicicletas[bicicleta] = alterar_cadencia(bicicletas['Bicicleta 1'], -100)

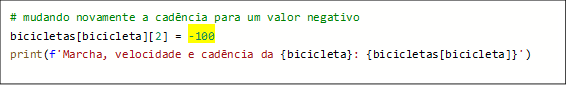


Ao analisarmos novamente a função *“alterar\_cadencia*”, vemos que isto, na realidade, é esperado: dentro da função existe um **if** que altera quaisquer cadências negativas para zero. Poderíamos dizer que é uma forma de evitarmos que tenhamos cadências abaixo de zero na nossa lista:

def alterar_cadencia(bicicleta, cadencia):
    bicicleta[2] = cadencia if cadencia >= 0 else 0
    return bicicleta



Mas é justamente aqui que a dor de cabeça se apresenta: se realmente quisermos, há a possibilidade de burlarmos essa regra – basta alterar diretamente o valor da lista:



E, assim, a cadência estará negativa:



É claro que o exemplo aqui é meramente ilustrativo, mas pensemos nos problemas de **segurança** (ou **confiança**) que isso poderia causar. E se, em vez da cadência de uma bicicleta, estivéssemos falando das notas do sistema da universidade? Ou do valor da nota fiscal em um sistema contábil? Ou do saldo na conta em um *internet banking*?

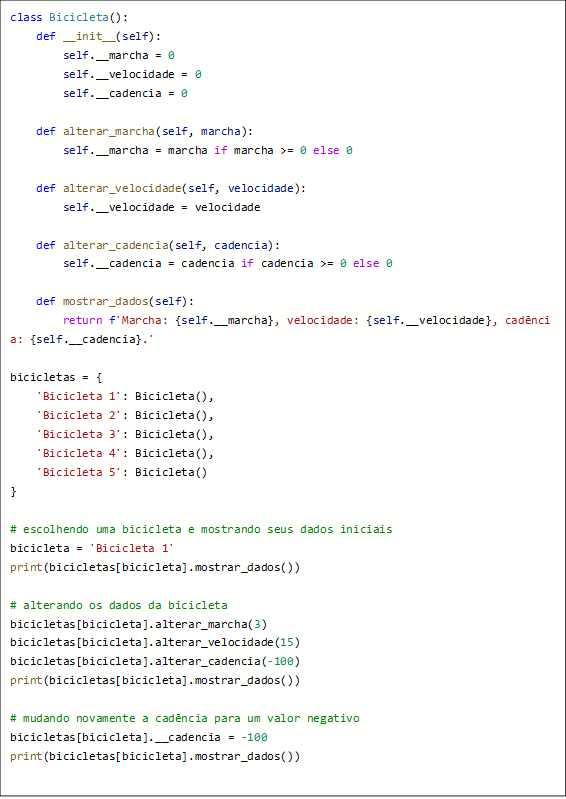
Como garantir que as **mesmas regras de negócio**sempre serão aplicadas para todos os casos sendo analisados, sem espaço para possíveis alterações indesejadas e tendo uma boa confiabilidade?

É aí que entra o paradigma da programação orientada a objetos.

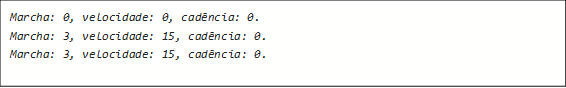
**TEMA 2 – NOÇÕES DE PROGRAMAÇÃO ORIENTADA A OBJETOS**

A programação orientada a objetos (POO) e os bancos de dados relacionais que usam o SQL possuem uma origem muito similar: ambos basicamente procuram representar o mundo real em um código. Da mesma forma que dois objetos não ocupam jamais o mesmo lugar, duas bicicletas também são coisas completamente diferentes. Esta lógica também se aplica a duas notas fiscais ou duas pessoas. Em SQL, temos tabelas e, dentro das tabelas, temos registros e colunas. Em POO, algo similar também é possível.

Veja como seria o mesmo código apresentado, mas implementado usando técnicas de POO:



Vamos explicar cada um dos conceitos. Antes disso, observe como ficaram os resultados – veja que, apesar de tentarmos “forçar” a cadência para um número negativo (-100), o algoritmo não permite:



**2.1 OBJETOS**

O principal conceito da POO são, como o próprio nome já diz, os objetos. Para falarmos dos objetos e de POO, deixemos um pouco de lado todos os conceitos aplicáveis ao mundo do desenvolvimento de algoritmos e pensemos no mundo real. Olhe ao seu redor. O que você vê agora? Provavelmente deve estar vendo uma tela, um teclado, um celular, uma janela, uma porta, quem sabe até mesmo um copo de água (a propósito, você já bebeu água recentemente?).

Agora, perceba que “uma tela” ou “uma porta” ou “um celular” são únicos em si: alguém não teria “um celular do modelo Apple iPhone 12” em mãos, mas sim “um celular do modelo Apple iPhone 12 com o número de série A1234567890”. A porta é única em sua estrutura: ainda que várias outras portas sejam bem parecidas, a porta mais próxima de você é única, e só ela existe exatamente do jeito que ela é (talvez com alguns riscos ou com a pintura um pouco descascada).

Observe que todas as coisas do mundo real são únicas em si: até mesmo se nós estivermos segurando uma folha de papel em branco cada, teremos em mãos *folhas diferentes*. Elas podem até ser bem parecidas, mas perceba que a sua folha de papel **não é**a minha folha de papel.

Ainda que tenhamos uma bicicleta azul cada um, fabricadas pela mesma fábrica e pertencentes ao mesmo modelo, teremos também bicicletas totalmente diferentes: a nossa bicicleta pertence a nós e, a sua, a você. Pode ser que você cuide melhor da sua bicicleta do que nós ou, independentemente do caso, ainda que estejam juntas uma da outra, elas **nunca ocuparão o mesmo lugar ao mesmo tempo** – lembra quando mencionam na física que dois objetos jamais ocupam o mesmo lugar? Pois bem.

Os objetos da vida real possuem **estados**e **comportamento**. Estados são aquelas coisas que **definem**um objeto. Para um carro, pode ser placa, número de chassi, cor, ano e modelo. Para um cachorro, pode ser cor, raça, idade e nome. Logo, entendemos estados como sendo as suas **características**. Já os comportamentos são **ações** que um objeto pode fazer. Um carro pode virar o volante, pode estar em movimento ou parado, pode estar ligado ou desligado. Um cachorro pode correr, abanar o seu rabo ou latir, por exemplo.

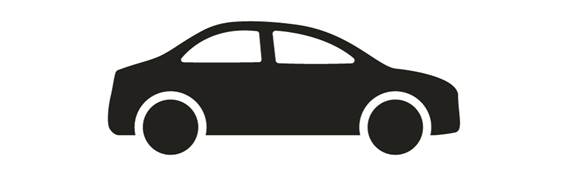
Em POO, mudamos um pouco os nomes:

* **estados**(características) – são armazenados em variáveis e constantes (que agora serão chamadas de **atributos**); e
* **comportamentos**(ações) – são geridos por funções (que agora serão chamadas de **métodos**).

**2.2 CLASSES**

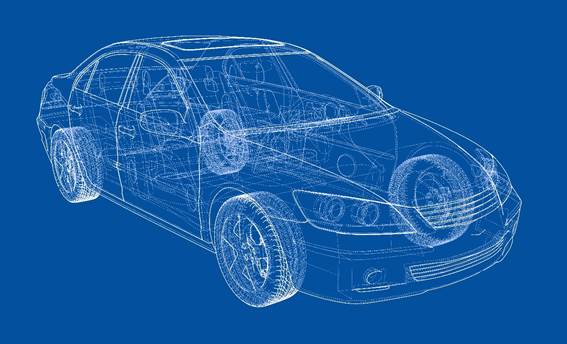
Como você deve imaginar, na realidade os objetos não possuem os mesmos estados e comportamentos ao mesmo tempo. Neste exato momento, todos os cachorros do mundo **não**estão latindo, nem todos os carros estão em movimento. Logo, cada carro e cada cachorro terá os seus estados e comportamentos de forma independente uns dos outros **ainda que possam compartilhar**os tipos de estados que são possíveis: todos os carros do mundo possuem alguma cor e foram fabricados em algum ano. Todos os cachorros do mundo também possuem alguma cor e alguma raça (ainda que a raça seja “sem raça definida”). É como se todos os objetos de um mesmo “tipo” seguissem sempre um molde mínimo contendo todos os estados e comportamentos esperados.

Figura 1 – Interface



Crédito: premium design/Shutterstock.

Figura 2 – Classe



Crédito: cherezoff/Shutterstock.

Figura 3 – Objetos da classe



Crédito: Paisan579/Shutterstock.

O nome que damos a este molde é ***classe***: uma classe chamada *Carro* terá um conjunto predefinido de atributos e métodos, e isso também vale para uma classe chamada *Cachorro*. Imagine uma classe como um manual de como gerar um novo objeto ou um cardápio – a classe *Carro* não é um carro em si, mas é tão somente uma lista de definições (como um manual de instruções, mesmo) contendo tudo o que um novo objeto (um novo “Carro”) deveria ter.

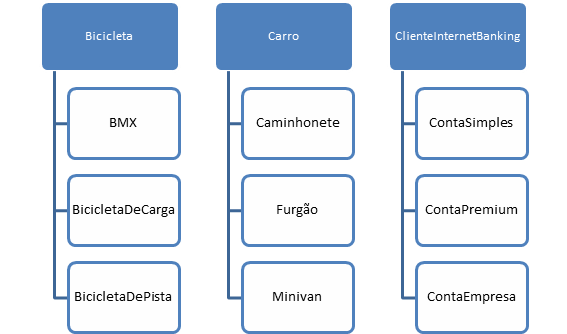
Ao criarmos um objeto de uma classe, estamos **instanciando-a**,ou seja, estamos criando um objeto seguindo o “manual” preestabelecido. Se a classe chamada *Carro* possui cinco atributos (placa, número de chassi, ano, cor e modelo) e três métodos (virar o volante, movimentar/parar, ligar/desligar), sabemos que todos os carros que instanciarmos deverão sempre seguir estes comportamentos: todos os carros virarão um volante seguindo o mesmo comportamento (para a direita e para a esquerda, e não como para cima e para baixo); vão se movimentar do mesmo jeito (ou seja, para frente ou para trás, mas não começarão a flutuar do nada); e terão um conjunto de atributos predeterminado. Isto nos dá muita segurança e previsibilidade ao trabalharmos com aplicações mais complexas: já pensou o que aconteceria se um método chamado *mostrar saldo* funcionasse de formas completamente diferentes a cada dia para um mesmo conjunto de pessoas? Ou já pensou o que aconteceria se cada “Carro” não tivesse nenhum **padrão**de direção?

**2.3 HERANÇAS**

Heranças são outro conceito bem importante em POO. Ao ler as descrições anteriores, você deve ter pensado algo como: “beleza, mas nem todos os carros são iguais – existem carros que são à combustão que funcionam de forma diferente de carros elétricos. Já vi carros que nem possuem volante para virar! E aí?”. Ou, ainda: “existem cachorros que fazem mais coisas do que abanar o rabo e latir. Existem cachorros que gostam de correr, e outros que gostam de caçar”. Ou, voltando ao nosso exemplo das bicicletas: “Existem bicicletas que são especializadas para cada caso. Existem bicicletas para correr em velódromos, e outras, para transportar cargas. Também existem bicicletas para manobras, como a BMX. Nem todas são iguais!”. Ou, no caso do *internet banking*: “nem todos os clientes são iguais! Alguns são pessoa física, e outros, jurídica. Existem também clientes com mais privilégios do que outros”. E todos esses argumentos são justificáveis.

É para isso que serve a **herança**: podemos criar classes com base em outras classes, mas que respeitam as especificidades do seu caso de uso. No caso do carro.

Figura 4 – Exemplo de classes



Veja que uma classe herdada possui esse nome porque ela **herda**os métodos e atributos da classe principal: BMX herdou tudo de bicicleta, mas podemos modificar algumas coisas (as BMX possuem marchas diferentes) e podemos incluir novos métodos (como a execução de manobras).

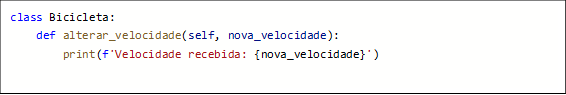
**2.4 INTERFACES**

Além disso, também podemos implementar **interfaces**: existem casos nos quais queremos somente definir quais métodos uma classe deve ter, mas não queremos implementar a **lógica**dentro dos métodos. No caso das bicicletas ou dos clientes de banco, pode ser que cada implementação de um método seja completamente diferente da de outro, mas, mesmo assim, precisamos garantir que todas as classes utilizem um conjunto predefinido de métodos: o nome disso é *interface*.

**TEMA 3 – CLASSES E OBJETOS EM PYTHON**

Da mesma forma que a biblioteca-padrão do Python oferece listas, dicionários de dados e funções, ela também possibilita a criação de algoritmos utilizando POO[[1]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a3&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnCIzWeKVrnvUa21OCM6CNZw=&ne=False#_ftn1).

Vamos começar com o mesmo caso das bicicletas que vimos anteriormente, mas agora seguindo um passo a passo:

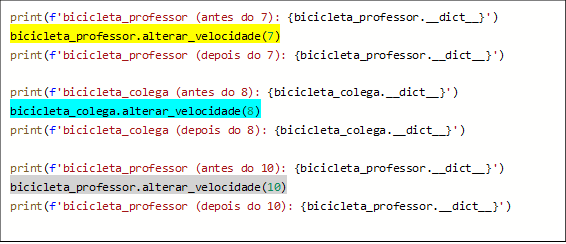


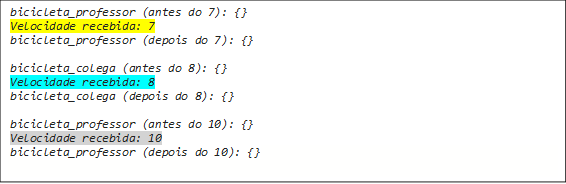
Veja que a classe foi **definida**ao usarmos a palavra-chave *class* seguida pelo nome (no caso, *Bicicleta*). Isto é bem parecido com a forma como criamos funções em Python, em que usamos o *def*seguido pelo nome da função. Após o nome da classe, é **opcional** usarmos parênteses, ao contrário das funções.

**Dentro**da classe, podemos criar uma função. Neste primeiro momento, só queremos testar para saber se estamos fazendo tudo certo. Para tal, só mostramos em tela a velocidade que recebemos no método *alterar\_velocidade*. Este método também possui o parâmetro *self*. Isto significa que o método pode ser utilizado pelos objetos, e os seus resultados se aplicam somente ao escopo do objeto em si. Lembre-se: agora só temos a *classe*, mas nenhum *objeto –*ou seja, não temos nenhuma bicicleta instanciada no nosso código. Para instanciar, basta fazer isto:

bicicleta_professor = Bicicleta()
bicicleta_colega = Bicicleta()

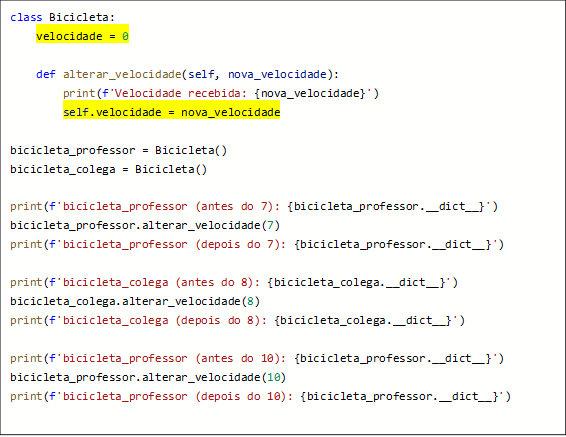

Agora, sim, temos duas instâncias de bicicletas: uma chamada *bicicleta\_professor*, e outra, *bicicleta\_colega.*Espera-se que a lógica de ambos seja igual, mas um é independente do outro em sua essência. Veja como chamamos os métodos para cada um desses objetos e os seus resultados:

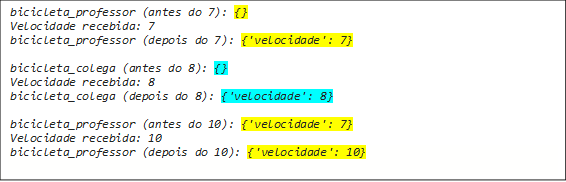




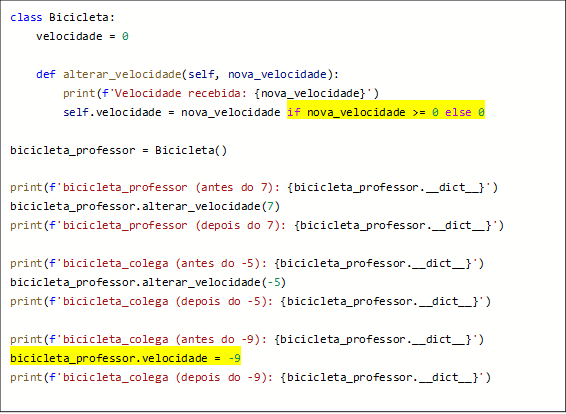
Os valores que informamos (7, 8 e 10) são as velocidades. Logo, aquele *self*não é considerado como o primeiro parâmetro, mas, sim, o parâmetro “*nova\_velocidade*”. Veja que usamos o ponto seguido pelo nome do método para chamar um método para um objeto em específico.

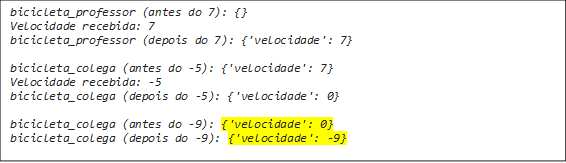
Isto posto, a forma como está agora não é das melhores: como saber se a velocidade realmente mudou para um objeto? Ao mostrarmos o antes e o depois de chamarmos cada um dos métodos, o resultado é aparentemente a mesma coisa: um dicionário vazio de atributos (o *\_\_dict\_\_*é uma funcionalidade-padrão do Python que mostra o que existe dentro de uma classe). Logo, a pergunta então é outra: como **armazenar**esses valores dentro de um objeto? Vamos inserir um novo parâmetro chamado *velocidade*fora do método, mas dentro da classe.



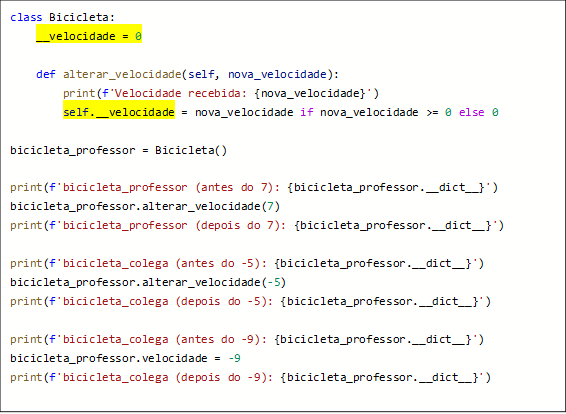


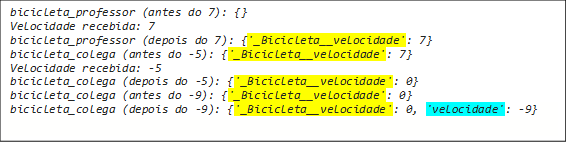
Vejamos isso! Agora, a variável *velocidade*está sendo armazenada dentro de cada objeto, e de forma individual. A velocidade da *bicicleta\_professor* não interfere na velocidade da *bicicleta\_colega*. Agora, vamos aumentar um pequeno grau de complexidade: vamos modificar o método *alterar\_velocidade*para não permitirmos velocidades negativas e tentar passar uma velocidade negativa de duas formas para a *bicicleta\_professor*: a primeira, via método; a segunda, forçando diretamente a variável.





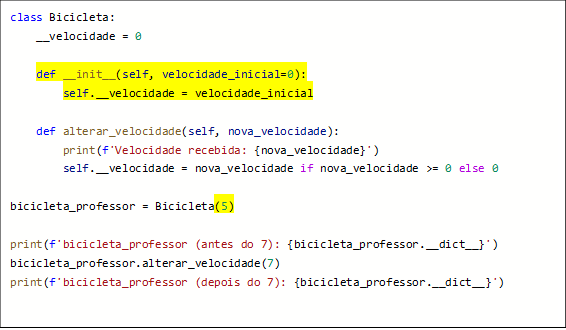
Veja que a classe permitiu que forçássemos o –9 sem maiores problemas. E agora, como endereçaríamos isso? É para isso que existem os **atributos privados**: atributos que somente deveriam ser acessados e modificados de dentro da classe e por seus respectivos métodos. Em Python, os atributos privados possuem dois *underscores* (“\_”) em seu início. Observe:

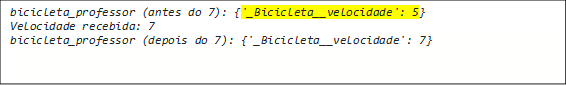




Note que ainda conseguimos visualizar a variável (agora chamada de *\_Bicicleta\_\_velocidade*), mas o risco de mudarmos manualmente é menor (quando tentamos mudar para –9, acabamos não afetando a *\_Bicicleta\_\_velocidade*, mas, sim, só criando uma variável chamada *velocidade*que o objeto acaba não utilizando). A própria documentação do Python[[2]](https://conteudosdigitais.uninter.com/libraries/newrota/?c=/gradNova/2022/bachareladoEngSoftware/linguagemProgramacao/a3&hash=3u0Jf1zuk93zpxVYIcnLAmHwQtRvF/KOUxjnpHAo9IsItQ+AbiwLM2ZnNOiiviwQYrpRE2cPLDfGZP8GY7ShDrifjTFdO8ZTK7JR0U1R+J73kuG9KINsDsgQKyh/jjuGhOpbG/8YAMKWd8feh6ylnCIzWeKVrnvUa21OCM6CNZw=&ne=False#_ftn2) diz que é possível modificarmos os valores de variáveis privadas, mas que, ao seguirmos essa convenção, diminuímos o risco de modificá-las acidentalmente.

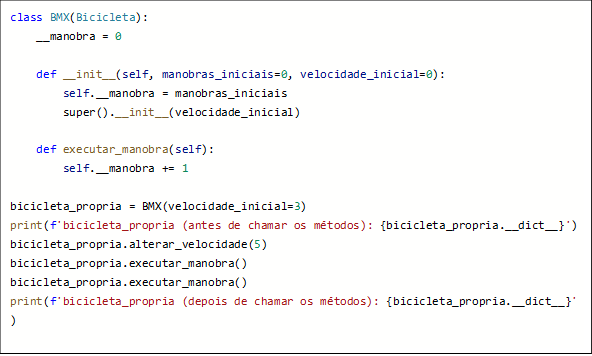
E se quiséssemos instanciar o *bicicleta\_professor*com um valor inicial para a velocidade? Há uma função-padrão em Python chamada *\_\_init\_\_* na qual podemos informar valores que devem ser usados logo ao inicializar a classe. Lembra do *random\_state* que usamos no *scikit-learn* anteriormente? Este é um exemplo de parâmetro que é usado na inicialização da classe daquela biblioteca. No nosso caso, vamos utilizar o *\_\_init\_\_* para definir uma velocidade inicial da bicicleta. Se nenhuma velocidade for informada, usaremos o zero como padrão.

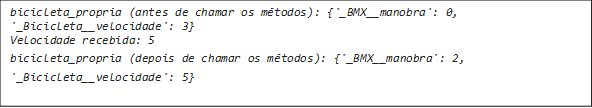




Seguindo na mesma linha, o que precisaríamos fazer se quiséssemos usar a herança para criar uma classe chamada BMX, a qual usaria a classe Biblioteca como base? Ao analisarmos o código a seguir, perceba algumas coisas.

* Na definição da classe BMX, estamos colocando entre parênteses a classe Bicicleta. Isso significa que BMX herda de Bicicleta.
* O \_\_init\_\_ é próprio da classe BMX, mas ela também logo em seguida chama o método \_\_init\_\_ próprio da classe Bicicleta por meio do *super()*, um termo-padrão do Python para chamar a classe herdada.
* Veja que em momento algum estamos implementando o método *alterar\_velocidade.*Logo, estamos reaproveitando exatamente a mesma lógica da *alterar\_velocidade*. Se quiséssemos, também poderíamos reimplementar este mesmo método com uma lógica específica à classe BMX.
* Também estamos criando um método específico para a classe BMX chamado *executar\_manobra* – este método é próprio da BMX, mas não da Bicicleta.





**TEMA 4 – POLIMORFISMO**

Outro conceito-chave em POO é o **polimorfismo**: a possibilidade de diferentes classes implementarem de formas diferentes um mesmo método e/ou de uma mesma função operar de formas diferentes com tipos de dados distintos.

Um exemplo básico de polimorfismo é a função a seguir: ela aceita dois ou três parâmetros e consegue gerar um resultado para ambas as combinações.

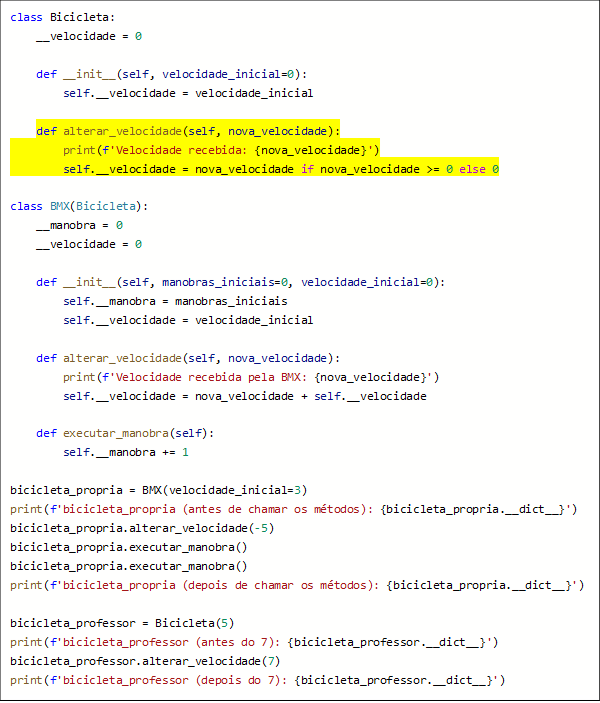
def mostrar_nome(primeiro_nome, sobrenome, nome_meio=None):
    if nome_meio is None:
        print(f'{primeiro_nome} {sobrenome}')
    else:
        print(f'{primeiro_nome} {nome_meio} {sobrenome}')

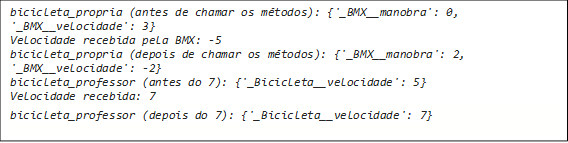
mostrar_nome('Jéssica', 'Silva')
mostrar_nome('Rodrigo', 'dos Santos', 'André')



Jéssica Silva
Rodrigo André dos Santos


Para classes, algo similar também pode ocorrer – comentamos anteriormente que a classe BMX poderia reimplementar a função *alterar\_velocidade*: este é um exemplo de polimorfismo também. Ambas as classes podem ter o mesmo nome da função, mas as lógicas podem ser completamente diferentes uma da outra.





Os resultados (e a lógica) de ambas as classes são diferentes. A BMX permite velocidades negativas – a Bicicleta, não.

**TEMA 5 – CLOSURES E DECORATORS**

Em Python também é possível termos **funções encapsuladas**, uma função que fica dentro de outra função. Essas funções que ficam dentro de outras funções e que usam variáveis atribuídas a elas são chamadas de ***closures***. O uso delas possui algumas vantagens, por exemplo, a redução de casos em que precisamos usar variáveis globais e, além disso, a oportunidade de encapsular a lógica (e protegê-la) do resto do algoritmo. O código a seguir implementa uma *closure* para transformar a escala de um valor – no caso, estamos convertendo números que estejam entre 0.0 e 1.0 para 0% a 100% e arredondando em 3 casas decimais quando necessário:

def converte_porcentagem(proporcao):
    # inicio closure
    def multiplicador(valor):
        return round(proporcao * valor, 3)
    # fim closure
    
    return multiplicador

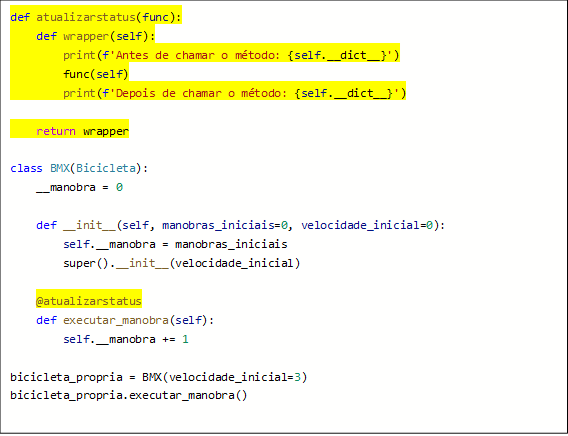
conversor = converte_porcentagem(100)
conversor(0.45)



45.0

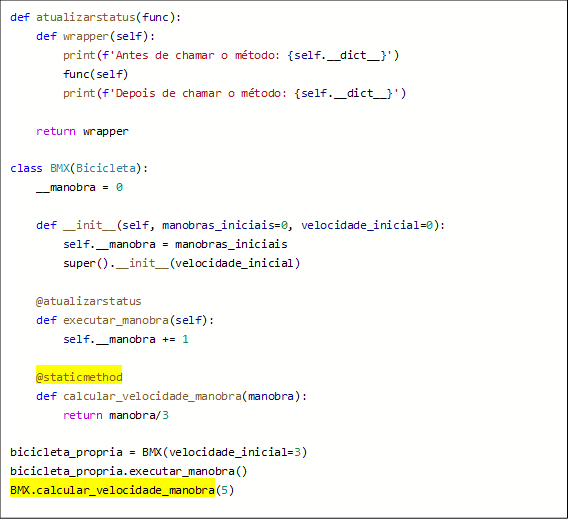
O ***decorator***é outra funcionalidade bem interessante em Python. Você deve ter percebido que em vários dos exemplos anteriores usamos muito a função *print*para mostrar como estavam as coisas antes e depois de aplicar determinada função. Felizmente, há uma forma de simplificarmos esse trabalho. Os *decorators* são *closures* que podem ser incluídas no código sem grandes alterações deste com o uso do “*@*”.

Vamos criar um *decorator* chamado “*atualizarstatus*”, o qual deve conter as mensagens que desejamos mostrar.





Veja que o *decorator* é uma função a qual encapsula outra função qualquer dentro dela – logo, da mesma forma como é utilizada para a *executar\_manobra*,ela também pode ser reaproveitada para a *atualizar\_velocidade* e quaisquer outras funções. O Python também possui alguns *decorators* já prontos: um deles é o @staticmethod. O @staticmethod serve para métodos estáticos (isto é, quando a lógica do método **não depende**do objeto, mas sim é algo específico da classe). Veja, por exemplo, o caso a seguir, em que calculamos a velocidade com base no número de manobras. O resultado desse método não possui relação com os objetos, mas é algo que possui relação com a **classe**.





**FINALIZANDO**

Esta aula teve como foco apresentar a você o tópico de programação orientada a objetos. Esse paradigma, como você deve ter percebido, possui vantagens e desvantagens. Como vantagens, poderíamos citar a sua facilidade em transcrever requisitos de negócio e conceitos da vida real em algoritmos funcionais (afinal de contas, é fácil entender o que é um “objeto” na realidade). Também é mais fácil de dar manutenção, pois é mais modular – logo, também é possível modificarmos pequenas partes do código com o risco menor de comprometer o todo. Além disso, pode ter vários trechos de código reaproveitados e tem ainda como vantagens a abstração da sua lógica e a segurança.

Por outro lado, também traz uma camada de complexidade bem maior, que se traduz em requisitos computacionais mais pesados, uma curva de aprendizagem bem maior e mais linhas de código. É por esta razão que aqui procuramos tomar um cuidado em traduzir vários conceitos para você de uma forma palatável, uma vez que existe um risco bem grande em tão somente repassar conceitos e não necessariamente **compreendê-los**.

Questão 1/10 - Linguagem de Programação

Independentemente da base sendo trabalhada é importante relembrar de um conceito chamado garbage in, garbage out: algoritmos de ML aprendem em cima de uma base emque os humanos fornecem. Ainda que a ideia possa soar interessante, por enquanto, os algoritmosainda não conseguem adivinhar o que nós pensamos e, ainda, procurar novos dados da internet semsupervisão alguma. Dessa maneira, pode-se realizar uma análise de dados exploratória.

Diante deste contexto, assinale a única afirmativa CORRETA sobre dados exploratória.

I. A matriz de confusão é uma ferramenta poderosa na preparação das bases de dados com a finalidade de evitar o "garbage in, garbage out".

II. A métrica que é mais utilizada por cientistas de dados para avaliar se um algoritmo está bom ou ruim é o R-quadrado (R2).

III. Serve para ver como está o relacionamento entre os dados  e a Iris é considerada como uma das bases de dados mais simples.

IV. Testes estatísticos também podem ser usados para medir a performance de algoritmos de aprendizagem não-supervisionada.

É correto apenas o que se afirma em

Nota: 10.0

|  |  |  |
| --- | --- | --- |
|  | A | II. |
|  | B | III e IV  **Você acertou!**  (Aula 02 – Tema 2.5 e 3) |
|  | C | I e III. |
|  | D | II e IV. |
|  | E | I. |

Questão 2/10 - Linguagem de Programação

Você foi recém-contratado para atuar em uma equipe de ciência de dados de forma remota. Após receber o seu computador em casa e configurar todos os seus acessos e softwares, você já recebeu a sua primeira demanda: desenvolver um algoritmo preditivo utilizando Python. Dito isso, uma das bibliotecas que você utilizará para o desenvolvimento deste algoritmo será o NumPy.

Diante deste contexto, avalie as afirmações a seguir.

I. O NumPy possui uma estrutura básica chamada de "ndarray", referida por vezes no código como "np.array". Esta é uma estrutura parecida com listas, mas com melhor performance e mais versátil.

II. O NumPy possui funções que permitem a criação de algoritmos preditivos.

III. O NumPy vem dentro da biblioteca padrão do Python.

IV. O NumPy é empregado principalmente para comparar as métricas de erro de algoritmos preditivos.

É correto apenas o que se afirmar em:

Nota: 0.0

|  |  |  |
| --- | --- | --- |
|  | A | II |
|  | B | I e III |
|  | C | II e IV |
|  | D | I  (Aula 01 – Tema 03) |
|  | E | I e II |

Questão 3/10 - Linguagem de Programação

Nos últimos 10 meses você este atuando como estagiário em Ciência de Dados de uma empresa da cidade que começou a atuar com isto recentemente. Você demonstrou um grande conhecimento e uma grande capacidade de aprendizagem e síntese e, por isso, você foi efetivado. Por outro lado, as demandas estão aumentando e precisam de mais pessoas para ingressar na equipe. Por isso, abriram um novo processo de vagas de estágio e pediram a sua ajuda para criar algumas questões com o intuito de avaliar o real conhecimento dos candidatos ao estágio. Dessa forma, uma das questões que você pensa em criar envolve a geração de gráficos utilizando Python.

Diante deste contexto, avalie as afirmações a seguir.  
I. Pode-se afirmar que gráficos não possuem ligação alguma com algoritmos em Python. Afinal, códigos em Python são códigos escritos em texto, e não possuem relação alguma com gráficos.  
II. NumPy e Pandas são bibliotecas conhecidas e especializadas na geração de gráficos.  
III. Um dos tipos mais simples de gráfico é o de dispersão (também conhecido como scatter plot).  
IV. Dependendo da biblioteca de geração de gráficos é possível termos acesso a configurações diferentes. Logo, algumas bibliotecas podem ser mais difíceis de usar, mas como contrapartida oferecem mais possibilidades para a customização de gráficos.  
  
É correto apenas o que se afirmar em:

Nota: 10.0

|  |  |  |
| --- | --- | --- |
|  | A | II |
|  | B | I e II |
|  | C | III e IV.  **Você acertou!**  (Aula 01 – Tema 04) |
|  | D | IV |
|  | E | I |

Questão 4/10 - Linguagem de Programação

Você está participando do processo de seleção de um programa de estágio em Tecnologia. Mais precisamente, este programa está procurando estudantes que estão no meio ou no final da Graduação para ingressar no departamento de Tecnologia de uma grande empresa. Sendo assim, neste processo de seleção existem dinâmicas de grupo para a resolução de situações que podem acontecer no dia a dia de um trabalho de desenvolvedor de software. A ideia é mensurar o seu nível de conhecimento e autonomia para o desenvolvimento de algoritmos. Uma dessas situações envolve o uso de bibliotecas em códigos escritos em Python.

"Considerando as informações apresentadas no texto, avalie as asserções a seguir e a relação proposta entre elas.

I. É possível conseguir rapidamente adicionarmos funcionalidades no seu código que já foram testadas e desenvolvidas por várias outras pessoas.

PORQUE

II. As bibliotecas possibilitam que testemos diferentes implementações que possam servir para a mesma finalidade e escolher, dentre elas, a melhor. Por exemplo: podemos ter dezenas de bibliotecas que geram gráficos, e podemos escolher uma que melhor se adapta aos nossos cenários."

A respeito dessas asserções, assinale a opção correta.

Nota: 10.0

|  |  |  |
| --- | --- | --- |
|  | A | As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa correta da I. |
|  | B | As asserções I e II são proposições verdadeiras, e a II é uma justificativa correta da I.  **Você acertou!**  (Aula 01 – Tema 01) |
|  | C | As asserções I e II são proposições falsas. |
|  | D | A asserção I é uma proposição verdadeira, e a II é uma proposição falsa. |
|  | E | A asserção I é uma proposição falsa, e a II é uma proposição verdadeira. |

Questão 5/10 - Linguagem de Programação

Você está atuando em uma equipe para o desenvolvimento do principal software de uma startup. Esta equipe trabalha em formato de squad, seguindo as premissas de uma metodologia ágil. Logo, existem reuniões todos os dias para o monitoramento das atividades, reuniões no início de cada nova rodada de desenvolvimento (sprint) para a definição das atividades que ocorrerão nas próximas duas semanas e, finalmente, reuniões de encerramento de cada sprint. Após definidas as atividades de uma nova sprint, você entende que uma das suas primeiras tarefas é desenvolver uma nova funcionalidade utilizando Python.

Considerando as informações apresentadas no texto, avalie as asserções a seguir e a relação proposta entre elas.

I. Uma das possibilidades do Python é a criação de algoritmos preditivos via aprendizagem de máquina (machine learning - ML)

PORQUE

II. Bibliotecas como scikit-learn é potencialmente a biblioteca mais versátil utilizada em ML e LightGBM trabalha com *gradient boosting*composta por várias árvores de decisão.

A respeito dessas asserções, assinale a opção correta.

Nota: 10.0

|  |  |  |
| --- | --- | --- |
|  | A | A asserção I é uma proposição verdadeira, e a II é uma proposição falsa. |
|  | B | As asserções I e II são proposições verdadeiras, e a II é uma justificativa correta da I.  **Você acertou!**  (Aula 02 – Tema 04) |
|  | C | As asserções I e II são proposições falsas. |
|  | D | A asserção I é uma proposição falsa, e a II é uma proposição verdadeira. |
|  | E | As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa correta da I. |

Questão 6/10 - Linguagem de Programação

Você foi recém-contratado para atuar em uma equipe de ciência de dados de forma remota. Após receber o seu computador em casa e configurar todos os seus acessos e softwares, você já recebeu a sua primeira demanda: desenvolver um algoritmo preditivo utilizando Python. No caso, você está vendo de que forma poderia começar a criar o seu algoritmo.

Considerando as informações apresentadas no texto, avalie as asserções a seguir e a relação proposta entre elas.

I. Programação estruturada é uma forma que pode ser utilizada para criar algoritmos simples

PORQUE

II. Programação estruturada pode-se dizer que é relativamente fácil de se criar algoritmos simples e de entender o que ele faz, na prática.

A respeito dessas asserções, assinale a opção correta.

Nota: 10.0

|  |  |  |
| --- | --- | --- |
|  | A | A asserção I é uma proposição falsa, e a II é uma proposição verdadeira. |
|  | B | A asserção I é uma proposição verdadeira, e a II é uma proposição falsa. |
|  | C | As asserções I e II são proposições verdadeiras, e a II é uma justificativa correta da I.  **Você acertou!**  (Aula 03 – Tema 01) |
|  | D | As asserções I e II são proposições falsas. |
|  | E | As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa correta da I. |

Questão 7/10 - Linguagem de Programação

Você está desenvolvendo um trabalho acadêmico o qual envolve o desenvolvimento de um algoritmo em Python. O prazo de entrega termina em breve, e você está com último problema no código que precisa ser resolvido. Logo, para que você possa ser aprovado na disciplina, bastaria somente resolver este problema. No caso, você está tendo dificuldades em entender como usar um DataFrame do Pandas.

A partir das informações apresentadas, é correto afirmar que:

Nota: 10.0

|  |  |  |
| --- | --- | --- |
|  | A | É possível criar DataFrames a partir de duas listas no Python. Por outro lado, para criar um DataFrame a partir de um arquivo precisaremos, primeiro, converter o conteúdo do arquivo para estas duas listas. |
|  | B | DataFrames possuem a finalidade principal de garantir que nós, desenvolvedores de algoritmos, visualizemos melhor os dados na tela. |
|  | C | DataFrames tratam automaticamente todos os valores como sendo string. Isto permite uma compatibilidade com datas, textos e números. |
|  | D | DataFrames são estruturas de dados que não fazem parte do Pandas, mas sim do NumPy. |
|  | E | É possível ler DataFrames a partir de planilhas do Excel, tabelas em SQL, arquivos CSV, JSON e outros.  **Você acertou!**  (Aula 01 – Tema 04) |

Questão 8/10 - Linguagem de Programação

A Universidade está promovendo um evento na sua região com o intuito de reunir estudantes e demais desenvolvedores de software para comentar sobre as últimas tendências de mercado. Além disso, este evento possui algumas oficinas sobre boas práticas de código. Você está participando de uma dessas oficinas, cujo código é todo desenvolvido em Python e utiliza o scikit-learn para a análise de modelos preditivos.

Considerando as informações apresentadas no texto, avalie as asserções a seguir e a relação proposta entre elas.

I. O scikit-learn apresenta implementação de técnicas de aprendizagem supervisionada e não supervisionada

PORQUE

II. Pois apresentam técnicas de pré-processamento de dados e métricas que não visam reduzir o erro de resposta.

A respeito dessas asserções, assinale a opção correta.

Nota: 10.0

|  |  |  |
| --- | --- | --- |
|  | A | A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.  **Você acertou!**  (Aula 02 – Tema 05) |
|  | B | As asserções I e II são proposições verdadeiras, e a II é uma justificativa correta da I. |
|  | C | As asserções I e II são proposições falsas. |
|  | D | A asserção I é uma proposição falsa, e a II é uma proposição verdadeira. |
|  | E | As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa correta da I. |

Questão 9/10 - Linguagem de Programação

Ao ter acesso às novas disciplinas do seu curso nos últimos meses você começou a ter uma ideia: abrir a sua própria empresa que implemente serviços de Tecnologia para outras empresas. Logo, você pensa em sugerir a arquitetura de novas soluções em Python que possam se conectar aos sistemas já existentes destas empresas maximizando a performance e seguindo boas práticas de desenvolvimento de algoritmos.

Considerando as informações apresentadas no texto, avalie as asserções a seguir e a relação proposta entre elas.

I. O "random\_state" é um parâmetro que permite a reprodutibilidade dos testes e confiabilidade dos algoritmos

PORQUE

II. O "random\_state" é um parâmetro associado ao número aleatório utilizado para gerar a lógica do algoritmo.

A respeito dessas asserções, assinale a opção correta.

Nota: 10.0

|  |  |  |
| --- | --- | --- |
|  | A | As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa correta da I. |
|  | B | A asserção I é uma proposição falsa, e a II é uma proposição verdadeira. |
|  | C | As asserções I e II são proposições verdadeiras, e a II é uma justificativa correta da I.  **Você acertou!**  (Aula 02 – Tema 04) |
|  | D | A asserção I é uma proposição verdadeira, e a II é uma proposição falsa. |
|  | E | As asserções I e II são proposições falsas. |

Questão 10/10 - Linguagem de Programação

Você está participando do processo de seleção de um programa de estágio em Tecnologia. Mais precisamente, este programa está procurando estudantes que estão no meio ou no final da Graduação para ingressar no departamento de Tecnologia de uma grande empresa. Sendo assim, neste processo de seleção existem dinâmicas de grupo para a resolução de situações que podem acontecer no dia-a-dia de um trabalho de desenvolvedor de software. A ideia é mensurar o seu nível de conhecimento e autonomia para o desenvolvimento de algoritmos.

Diante deste contexto, avalie as afirmações a seguir.

I. Dois objetos com nomes diferentes, mas que pertençam à mesma classe são exemplos de polimorfismo.

II. Duas funções com nomes diferentes, mas que pertençam à mesma classe são exemplos de polimorfismo.

III. Duas variáveis com o mesmo nome dentro de funções diferentes são exemplos de polimorfismo.

IV. Duas funções com o mesmo nome são exemplos de polimorfismo.

É correto apenas o que se afirma em

Nota: 0.0

|  |  |  |
| --- | --- | --- |
|  | A | III e IV. |
|  | B | II e IV. |
|  | C | III. |
|  | D | IV.  (Aula 03 – Tema 02) |
|  | E | I e II. |