Relatório PPFCentral

Equipe Raspagem de Dados

O que são Biblotecas ou Módulo?

Uma biblioteca é uma coleção de módulos de script acessíveis a um programa *Python*, isto é, um pacote de códigos que está pronto no Python. Dessa forma, você pode instalar uma biblioteca que foi produzida por outra pessoa e utilizar as ferramentas dessa biblioteca para resolver os problemas que você está enfrentando.

Bibliotecas Básicas

Numpy

Definição: É uma biblioteca da linguagem Python, chamada de *Numerical Python*, é uma coleção de funções e operações que ajudam a executar cálculos numéricos com facilidade. O *NumPy* oferece uma biblioteca para cálculos fáceis e rápidos.

Para deixar a ideia mais clara de como funciona a biblioteca, podemos utilizar um exemplo mais prático. Suponha que você queira resolver uma equação do segundo grau. Para isto você precisa, por exemplo, ter um conhecimento das operações de soma e subtração. Tendo isso em mente, você irá atrás de um livro de matemática básica para obter tal conhecimento e conseguir executar os cálculos. Nesse exemplo, se levarmos para linguagem de programação, podemos dizer que a biblioteca *Numpy* seria o livro de matemática básica, onde se encontra o conhecimento de soma e subtração. Logo, ao utilizarmos uma função dessa biblioteca, estamos dizendo para máquina ir nesta biblioteca e pegar um certo "conhecimento" para executar algum cálculo.

Pandas

Definição: É uma biblioteca da linguagem Python, utilizada para manipulação e análise de dados. A biblioteca permite ler, manipular, agregar e plotar os dados de forma simples.

Para exemplificarmos esta biblioteca, podemos usar o exemplo da criação de uma matriz. Digamos que você queira criar uma matriz. Para isso, você precisa manipular certos dados para encaixá-los corretamente na matriz. O *Pandas* funcionaria como o livro que contém o conhecimento necessário para manipulação. Logo, ao executar uma função da biblioteca, estamos pedindo para a máquina acessar este "livro" e executar esses conhecimentos para construir a matriz corretamente.

WebScrapping

Definição: É o ato de coletar dados estruturados na Web de maneira automatizada. Dessa forma, podemos chamar o *WebScrapping* de Raspagem de Dados ou Extração de Dados da Web - ambas definições refletem bem o que é feito pelo *WebScrapping*. Neste sentido, a Raspagem de Dados desempenha um papel fundamental ao ceder os dados que serão utilizados pelas bibliotecas *Numpy* ou *Pandas* e para outros fins.

A prática de *Webscrapping*, é uma maneira de automatizar o processo da coleta de dados, em uma certa página para uma análise posterior. Para exemplificar este caso, suponha que queremos saber quantas ofertas do produto *Macbook air* existem na página principal do Mercado Livre. Podemos fazer isto manualmente, acessando a página e contando um a um. Porém, se quisermos economizar mais tempo e obter uma resposta com uma menor margem de erro, podemos utilizar algumas sequências de códigos que basicamente diz para máquina acessar o site e realizar esta contagem. Fazendo isso, utilizamos a capacidade de processamento da máquina para economizar tempo.

Bibliotecas de WebScrapping

urlib

Definição: É uma biblioteca para acessar, ler e fazer o *parse* (que é basicamente transformar um dado de um formato para outro) de uma URL. De certa forma, é uma biblioteca que realiza o *request* de uma URL, que possibilita a extração de dados feitas pelo BeautifulSoup.

requests

Definição: É uma biblioteca que requisita o acesso a uma URL. De certa forma, a biblioteca *requests* é considerada como *easy-to-use* ao ser comparada com a *urlib*. Apesar disso, a *urllib* é uma biblioteca que apresenta algumas funções a mais que a *requests* não apresenta.

BeautifulSoup (bs4)

Definição: É uma biblioteca para extrair dados de HTML e arquivos XML. Neste sentido, o *bs4* é uma biblioteca que necessita de bibliotecas como a *urlib* e a *requests* para poder funcionar. No geral, tem ótimos resultados e funciona de forma eficiente.

Definição do diretório do projeto

In [1]:

```
wd = r'C:/Users/vitor/Desktop/Education/LAMFO/Projetos/#mcti_datascienc
e/Github/Testes de Código'
```

Import dos módulos auxiliares

As bibliotecas usadas no *ppfcentral* estão listadas acima. A maioria é embutida no *python* básico, porém é necessário instalar também as bibliotecas: ("pandas","numpy","bs4","requests","currencyconverter","googletrans","lxml") que são utilizadas tanto no *ppf* quanto nos imports dos *scrappers*.

In [2]:

```
import filecmp
import os
import shutil
from datetime import datetime
from filecmp import dircmp
from inspect import getmembers, isfunction
from itertools import compress
from os import walk

import numpy as np
import pandas as pd
```

Import automático dos módulos de Scrapping

In [3]:

```
def get_func(module):

    exec('import scrappers.' + module, globals())
    exec('funcs = getmembers(scrappers.' + module + ', isfunction)', glo
bals())

    return [f[0] for f in funcs if module.lower() in f[0].lower()]


os.chdir(wd)
name_scrappers = [
    i[:-3] for i in os.listdir(wd + '\scrappers') if i[-3:] == '.py'
]
funs = []
for i in name_scrappers:
    exec('from scrappers.' + i + ' import *')
    for f in get_func(i):
        funs.append(f)
```

A variável *name_scrappers* é uma lista com o nome de todos os arquivos que estão contidos na pasta *scrappers* do diretório do projeto. A função *get_func()* recebe o nome de um desses arquivos e retorna as funções de scrapping que aquele arquivo possui. Um exemplo de chamada desta função com o módulo *speciesconservation* retorna o seguinte resultado: *get_func('speciesconservation')* = {{get_func('speciesconservation')}}. A legenda utilizada para cada tipo de função é a seguinte:

1. Oportunidades
2. Notícias
3. Políticas
4. Projetos

O loop *'for'* utilizado no código faz o import de todos os módulos de *scrapping* usando a função imbutida no python *exec()*. Depois disso ele chama a função *get_func* e guarda todas as funções de scrapping que precisam ser chamadas na lista *'funs'*. Uma vantagem de fazer os imports dos módulos desta maneira é que economizamos centenas de linhas de código que seriam necessárias para fazer os *imports* por extenso.

Remover scrappers com problema

Algumas funções de *scrapping* apresentaram problemas que serão corrigidos futuramente com o auxílio do MCTI, por isso iremos inserir um bloco de código capaz de remover essas funções.

In [4]:

```
problematicos = [
    'fonplata1',
    'iadb2',
    'itto1',
    'erefdn4',
    'forskningsradet1'
]

for p in problematicos:
    funs.remove(p)
```

Definindo o diretório para salvar os arquivos

Neste bloco de código nós definimos a pasta que receberá as bases de dados que resultarem das funções de *scrapping*.

In [5]:

```
os.chdir(wd + r'/output')
```

Criação da pasta diária

A coleta de dados no *ppfcentral* acontece de forma contínua, todos os dias as funções são chamadas e caso encontrem atualizações na base de dados elas salvam essas alterações na pasta diária. A pasta *baseprincipal* funciona como um *"hard drive"* destas informações e guarda as informações que foram coletadas nos períodos anteriores. O bloco de código à seguir é responsável pela criação do diretório das informações coletadas no dia que o código for rodado.

In [6]:

```
dia = datetime.today().strftime('%y%m%d')  # yy/mm/dd
if os.path.exists(dia):
    print('Diretório já existente')  # nada acontece
else:
    os.makedirs(dia)  # cria o diretório
    print('Diretório criado')
```

```
Diretório já existente
```

Definição de termos chave para automarizar a classificação

Esta é a primeira tentativa de automatizar a classificação das oportunidades de acordo com a possibilidade de participação de cientistas brasileiros. Várias oportunidades não explicitam se o Brasil está ou não incluído na lista de países elegíveis, então definimos algumas palavras-chave que podem sugerir a inclusão do Brasil. Caso o computador encontre essas palavras nos textos referentes àquela oportunidade ele irá classificar aquele ponto de dado como 'Y', ou elegível. Futuramente modelos de inteligência Artificial (IA) serão desenvolvidos para melhorar a acurácia dessa classificação.#### Definição de termos chave para automarização da classificação:

In [7]:

```
keywords = [
    'brazil', 'latin america', 'underdeveloped', 'south america', 'bric
s',
    'mercosul', 'portuguese', 'middle income'
]

keywords = '(' + '|'.join(keywords) + ')'
```

Código que roda as funções de scrapping

A função *call_fun()* verifica se a função que está sendo chamada é correspondente à coleta de **Oportunidades** ou não. Caso seja, ela vai passar o parâmetro *keywords*, caso não seja ela somente chama a função usando a função imbutida *eval()*. O loop *'for'* itera por todos os itens da lista *funs* e vai salvando os dados coletados da internet na pasta diária definida.

In [ ]:

```
def call_fun(fun, path):

    global keywords
    # print(f'Início de {fun}') # esta linha de código foi comentada par
a deixar o relatório mais sucinto, é recomendável
    # deixar ela ativa para saber exatamente qual função que deu problem
a.
    if f[-1] != '1':
        text = fun + f"('{path}')"
        text = text.replace("\\", "\\\\")
    else:
        text = fun + f"('{path}', '{keywords}')"
        text = text.replace("\\", "\\\\")
    return eval(text)


try:
    for f in funs:
        call_fun(f, '.\\' + dia)

except Exception as e:
    print('Erro na extração, verificar arquivo fonte')
```

Função que atualiza a base

Essa função pega o arquivo do *"hard drive"*, que é o diretório *baseprincipal*, e adiciona ao *csv* existentes as novas informações que foram coletadas naquele dia.

In [8]:

```
def atualizador(baseprincipal, diamaisrecente):
    diario = pd.read_csv(diamaisrecente)
    main = pd.read_csv(baseprincipal)
    # checando o que do 'diario' está no 'main'
    a = diario['link'].isin(main['link'])  # usar o tag/id criado inves
 do link
    b = [not bool
         for bool in a]  # inverter para o TRUE ser a linha que não tem
 no main
    novaslinhas = diario[b]
    main = main.append(novaslinhas, ignore_index=True)  # novo main
    main.to_csv(baseprincipal, index=False, sep=",")
    #print('A base foi atualizada')
```

Pegando todos os diretórios da pasta output

Esse bloco de código acessa o diretório que contém os arquivos da base e os organiza em uma lista para ser usada nas próximas células do script.

In [9]:

```
_, dirnames, _ = next(walk(wd + r'/output'))
# Os obj 0 vai ser a base principal. o 1 vai ser o dia mais recente e o
 2 o dia anterior.
dirnames.sort(reverse=True)
# Arquivos extraídos no dia.
filenamesDia = next(walk(wd + r'/output'+'/'+dia))[-1]
# Arquivos extraídos na base.
filenamesBase = next(walk(wd + r'/output'+'/baseprincipal'))[-1]
```

Se o arquivo não estiver na base

Essa parte do código copia os arquivos que estão na pasta diária e não estão na base principal.

In [10]:

```
# Verificando se os nomes que estão no Dia estão na Base.
a = [i in filenamesBase for i in filenamesDia]
a = [not bool for bool in a]  # inversão pro True ser o arquivo faltante
# arquivo que está faltando na baseprincipal
arquivos = list(compress(filenamesDia, a))
for f in arquivos:
    shutil.copy('.\\'+dia+'\\'+f, '.\\baseprincipal')
```

Atualização

Essa parte aplica as funções definidas nos blocos de código anteriores e efetivamente concatena as novas informações que estão localizadas nos diretórios diários aos *dataframes* que estão localizados na *baseprincipal*.

In [11]:

```
def paths(pasta, arquivo):
    path = '''.\\''' + pasta + '''\\''' + arquivo
    return (path)


filenames = filenamesDia

for i in range(0, len(filenames)):
    try:
        base = str(paths(dirnames[0], filenames[i]))  # base
        dia1 = paths(dirnames[1], filenames[i])  # dia mais recente
        dia2 = paths(dirnames[2], filenames[i])  # dia anterior
        comp = filecmp.cmp(dia1, dia2, shallow=False)
        if comp == False:
            #print('arquivos diários são diferentes, base atualizada')
            atualizador(base, dia1)
        else:
            None
            #print('arquivos diários são iguais, base não atualizada')
    except:
        None
        #print("Arquivo do dia anterior não encontrado")

    #print('Concluido arquivo '+str(i+1))

# como a coluna codigo contem o dia, os arquivos sempre vão ser diferent
es.

# As linhas de código que jogam na tela a situação da atualização da bas
e foi comentadada para deixar o relatório mais sucinto
```

Criação das Bases Aumentadas

Cada função de *scrapping* criada salva um arquivo csv na base principal, então é necessário criar um bloco de código que junta essas centenas de arquivos em um só para termos nossa base de dados completa. Cada tipo de informação (oportunidades, notícias, políticas e projetos) possui um padrão de variáveis que é explicitado no arquivo README deste projeto. Por esse motivos é possível utilizar a função *pd.concat()* para juntar essas variáveis e concluir a base de dados final.

In [12]:

```
import warnings
warnings.filterwarnings('ignore')

# Criar dataframes
opo_baseprincipal = pd.concat([pd.read_csv(wd + r'/output'+'/baseprincip
al/'+i)
                               for i in [i for i in filenamesBase if i[-6
:] == '01.csv']])
not_baseprincipal = pd.concat([pd.read_csv(wd + r'/output'+'/baseprincip
al/'+i)
                               for i in [i for i in filenamesBase if i[-6
:] == '02.csv']])
pol_baseprincipal = pd.concat([pd.read_csv(wd + r'/output'+'/baseprincip
al/'+i)
                               for i in [i for i in filenamesBase if i[-6
:] == '03.csv']])
prj_baseprincipal = pd.concat([pd.read_csv(wd + r'/output'+'/baseprincip
al/'+i)
                               for i in [i for i in filenamesBase if i[-6
:] == '04.csv']])

# Salvamento das bases em folder específico
opo_baseprincipal.to_csv(
    wd + r'/output'+'/baseprincipal/basescompletas/oportunidades.csv', i
ndex=False)
not_baseprincipal.to_csv(
    wd + r'/output'+'/baseprincipal/basescompletas/noticias.csv', index=
False)
pol_baseprincipal.to_csv(
    wd + r'/output'+'/baseprincipal/basescompletas/politicas.csv', index
=False)
prj_baseprincipal.to_csv(
    wd + r'/output'+'/baseprincipal/basescompletas/projetos.csv', index=
False)
```

Resultado Esperado

Após rodar o código acima quatro arquivos no formato .csv devem ter sido criados na pasta *'basescompletas'* dentro da pasta *'baseprincipal'*. Exemplos de como cada um destes arquivos estão formatados serão incluídos abaixo.

Oportunidades:

In [13]:

```
opo_baseprincipal.head(10)
```

Out[13]:

	atualizacao	codigo	link	opo
0	220412	acmedsci_220412_1_000	https://acmedsci.ac.uk/grants-and-schemes/gran...	
0	220329	cargill_220329_1_000	https://www.cargillglobalscholars.com/scholars...	
0	220329	cepf_220329_01_000	https://www.cepf.net/grants/open-calls-for-pro...	
1	220329	cepf_220329_01_001	https://www.cepf.net/grants/open-calls-for-pro...	
2	220329	cepf_220329_01_002	https://www.cepf.net/grants/open-calls-for-pro...	
3	220329	cepf_220329_01_003	https://www.cepf.net/grants/open-calls-for-pro...	
0	220329	conservationleadership_220329_1_000	https://conservationleadership.si.edu/opportun...	
0	220329	rufford_220329_1_000	http://conservegrassland.org/our-programs/	
1	220329	rufford_220329_1_001	NaN	
2	220329	rufford_220329_1_002	NaN	

Notícias:

In [14]:

```
not_baseprincipal.head(10)
```

Out[14]:

	atualizacao	codigo	link	not_texto
0	220329	acmedsci_220329_2_000	https://acmedsci.ac.uk/more/news/press-office	Press Office Welcome to the Press and Communic...
1	220329	acmedsci_220329_2_001	https://acmedsci.ac.uk/more/news/academy-expla...	Academy explainer: What do this week's R&D ann...
2	220329	acmedsci_220329_2_002	https://acmedsci.ac.uk/more/news/health-and-so...	Health and social care leaders unite to improv...
3	220329	acmedsci_220329_2_003	https://acmedsci.ac.uk/more/news/ukraine-invas...	Ukraine invasion: Academy of Medical Sciences ...
4	220329	acmedsci_220329_2_004	https://acmedsci.ac.uk/more/news/cross-funder-...	Cross-funder statement on COVID-19 in future g...
5	220329	acmedsci_220329_2_005	https://acmedsci.ac.uk/more/news/health-inequa...	Health inequalities beyond the pandemic: new r...
6	220329	acmedsci_220329_2_006	https://acmedsci.ac.uk/more/news/use-our-succe...	Use our successful approach to supporting wome...
7	220329	acmedsci_220329_2_007	https://acmedsci.ac.uk/more/news/presidents-re...	President's response to the Government's Livin...
8	220329	acmedsci_220329_2_008	https://acmedsci.ac.uk/more/news/qa-with-our-p...	Q&A with our President: COVID-19 two years on ...
9	220329	acmedsci_220329_2_009	https://acmedsci.ac.uk/more/news/our-flagship-...	Our flagship future leaders programme is now o...

Políticas:

In [15]:

```
pol_baseprincipal.head(10)
```

Out[15]:

	atualizacao	codigo	lin
0	220329	acmedsci_220329_3_000	https://acmedsci.ac.uk/policy/overview/how-polic
1	220329	acmedsci_220329_3_001	https://acmedsci.ac.uk/about/support-us/how-we..
0	220329	adaptationfundorg_220329_3_000	https://www.adaptation-fund.org/about
0	220329	arcadia_220329_3_000	https://www.arcadiafund.org.uk/how-we-operat
0	220329	aucklandzoo_220329_3_000	https://www.aucklandzoo.co.nz/about-the-zo
0	220329	biocarbonfund_220329_3_000	https://www.biocarbonfund-isfl.org/who-we-ar
0	220329	biocodexmicrobiotafundation_220329_3_000	https://www.biocodexmicrobiotafoundation.com/f..
1	220329	biocodexmicrobiotafundation_220329_3_001	https://www.biocodexmicrobiotafoundation.com/i..
0	220329	cargill_220329_3_000	https://www.cargillglobalscholars.com/about-ca..

0	220329	climateandlandusealliance_220329_3_000	https://www.climateandlandusealliance.org/init..

Projetos:

```
In [16]: prj_baseprincipal.head(10)
```

Out[16]:

	atualizacao	codigo	link	prj_brazil	prj_instituicao
0	220329	adaptationfundorg4220329_4_000	https://www.adaptation-fund.org/project/enhanc...	N	adaptationfundorg4
1	220329	adaptationfundorg4220329_4_001	https://www.adaptation-fund.org/project/enhanc...	N	adaptationfundorg4
2	220329	adaptationfundorg4220329_4_002	https://www.adaptation-fund.org/project/streng...	N	adaptationfundorg4
3	220329	adaptationfundorg4220329_4_003	https://www.adaptation-fund.org/project/artik-...	N	adaptationfundorg4
4	220329	adaptationfundorg4220329_4_004	https://www.adaptation-fund.org/project/buildi...	N	adaptationfundorg4
5	220329	adaptationfundorg4220329_4_005	https://www.adaptation-fund.org/project/practi...	N	adaptationfundorg4
6	220329	adaptationfundorg4220329_4_006	https://www.adaptation-fund.org/project/increa...	N	adaptationfundorg4
7	220329	adaptationfundorg4220329_4_007	https://www.adaptation-fund.org/project/ecosys...	N	adaptationfundorg4
8	220329	adaptationfundorg4220329_4_008	https://www.adaptation-fund.org/project/enhanc...	N	adaptationfundorg4
9	220329	adaptationfundorg4220329_4_009	https://www.adaptation-fund.org/project/buildi...	N	adaptationfundorg4