

Daniel Gomes Benjamim Rodrigues

27 anos, brasileiro



Belo Horizonte



+55 (33) 988047654



danielgbenjamim@gmail.com



LinkedIn

In [8]:

1	Plotando_Minhas_Habilidades_em_Linguagens()
executed in 1.58s, finished 18:48:53 2021-01-18	

Python



R



C#



HTML



SQL



CSS



PySpark



In [9]:

1	Plotando_Conhecimento_em_Bibliotecas()
executed in 148ms, finished 18:48:53 2021-01-18	



In [29]:

1	Minha_Trajectoria_Projetada_no_Pandas(_)
executed in 714ms, finished 19:11:12 2021-01-18	

Sobre Mim

Quem sou eu:

Sou uma pessoa que se incomoda ao ver ou realizar manualmente tarefas que meu computador seria capaz de realizar cem vezes mais rápido. Resolvi aprender linguagens de programação para tornar as máquinas grandes parceiras nas minhas empreitadas. Sinto-me motivado sempre que estou a ensiná-las o que quero que façam ao automatizar processos. Minha característica mais forte é o pensamento divergente que, em consequência, manifesta em mim uma criatividade que esperneia quando fica muito tempo inativa. Essas características me permitem encontrar soluções diferentes, inusitadas e/ou não convencionais para os problemas e obstáculos que me aparecem. Possuo uma personalidade extrovertida que exige que eu vá além desses mundos virtual e artificial e clama por conversar e conhecer novas pessoas, desenvolver o networking e cativar amizades. Aprendi mais ainda sobre o quanto prezo o enriquecimento interpessoal ao adentrar no curso de oratória.

Engenheiro Químico pela Universidade Federal Fluminense
04/2013 - 07/2018

Curso online de Nanodegree de Machine Learning pela Udacity
Aprendizagem Supervisionada e Não Supervisionada / Redução de dimensionalidade / Clusterização / Classificação / Regressão / Redes Neurais
03/2019 - 10/2019

Formação:

Curso de Oratória pela Vox2You
03/2019 - 03/2020

Curso de Ciência de Dados no Instituto Metodista Izabela Hendrix
08/2019 - 12/2021

Curso Online End-to-End Machine Learning with TensorFlow on GCP no Coursera
01/2021 - atualmente

Curso Online IBM AI Enterprise Workflow Specialization no Coursera
01/2021 - atualmente

10/2017 - 04/2018

Atuava com a logística de entrada, estoque e saída de matérias primas e produtos. Para isso, me utilizava do software de gestão SAP e atualizava rotineiramente planilhas com as informações necessárias para gerir a movimentação de produtos e intermediários ao longo da fábrica. Analisava se o registro das informações no software estava de acordo com o ocorrido e buscava entender e corrigir possíveis erros que poderiam vir a se apresentar. Com o intuito de otimizar minhas tarefas diárias, aprofundi bastante meus conhecimentos sobre a ferramenta Visual Basics. Desenvolvi algoritmos no VBA que se utilizavam de relatórios exportados do SAP para preencher planilhas de controle da logística do setor de PCP e Almoarifado e concluía minhas tarefas diárias em minutos.

Excel Avançado / Visual Basics / SAP ERP

Estagiário de Ciência de Dados na Localiza

03/2020 - atualmente

Experiência:

Primeiros 6 meses

Atuei como parte do time de portfolio na empresa. Desenvolvi e acompanhei dashboards e planilhas - Power BI e Excel - que elucidam dados de métricas voltadas para a performance dos vários times das várias tribos do setor de tecnologia da empresa. Auxiliei na transição da rotina de trabalho dos colaboradores neste período de pandemia, quando houve a instauração do trabalho 100% home office. Atuei fornecendo suporte na utilização do Microsoft Teams como ferramenta de comunicação e do Azure Devops como ferramenta de organização das atividades. Me familiarizei fortemente com a metodologia Agile.

Power BI / Excel / Teams / Azure Devops

A partir de setembro

Trabalhei diretamente na construção e evolução de um modelo preditivo. Praticando pair programming diariamente, atuei com desenvolvimento do canvas, análise dos dados e das features, limpeza dos dados, feature engineering, criação de features temporais, aplicação de bibliotecas de machine learning como Random Forest, Logistic Regression, catboost e xgboost para treino e validação dos dados. Operei também com consultas aos bancos de dados por meio da linguagem SQL. Atuei ainda na apresentação das trajetórias e dos resultados dos modelos e estudos para o time de negócio.

SQL Server / Jupyter notebook / Mlflow / Spark / GCP

Línguas:

Português: Nativo

Inglês: Fluente

Criei um jogo 2D em 2015 no GameSalad de endless running. A ideia do jogo é se utilizar de botões com setas aos cantos da tela do jogo que ativam campos elétrico e magnético com o intuito de evitar choques de uma bolinha de metal com os obstáculos que aparecem à frente. A ativação dos campos modifica a aceleração da bolinha na direção evidenciada, aumentando a velocidade e permitindo ao jogador continuar vivo no jogo. A bolinha perde sua carga ao se chocar com qualquer superfície, portanto, a capacidade de sofrer alterações em suas velocidades por meio do uso dos campos. O ambiente é também composto pelo teto e pelo chão, que são duas placas paralelas eletricamente carregadas com a mesma carga, o que acaba repelindo a bolinha para o meio da tela.

Lógica de Programação

Resolvi todos exercícios de python do site codingbat com apenas uma linha de código em cada, abusando da utilização de list comprehension. Montei um jupyter notebook que realiza o login no site na minha conta, busca todos os exercícios de cada seção, cria um novo notebook para cada seção, nos quais cada célula representa um exercício, tendo acima dela como markdown o enunciado do exercício, a linha de código que utilizei para resolver o exercício na própria célula, e abaixo dela um print do resultado do exercício após rodá-lo no site do codingbat.

Jupyter Notebook / Numpy / Pandas / Selenium / Pyautogui / json

Projetos:

Desenvolvi um algoritmo de recomendação de filmes. Qualquer um que tiver acesso ao notebook consegue escolher filmes que já tenha visto e atribuir uma nota a eles. Após o limite de 30 filmes avaliados, a opção de finalizar a avaliação de filmes lhe é apresentada. Então, cria-se o perfil de avaliação de filmes deste novo usuário num dataset já existente e correlaciona-se o perfil deste novo usuário com o resto dos usuários existentes no dataset. Dessa forma, ao final do notebook, é gerado duas listas de filmes, uma que o novo usuário provavelmente gostará de assistir e outra com filmes que o usuário possui uma enorme chance de gostar.

Matplotlib / Scipy / Pyautogui / Sklearn / Javascript / ipywidgets

Realizei um projeto com o intuito de estudar as mudanças e evoluções de variáveis que demonstrassem uma variação significativa com a instauração do lockdown ao redor do mundo. Quatro variáveis foram escolhidas para o projeto: nível de confinamento (de 0 a 3), redução percentual de emissão de CO2, percentual médio de redução de mobilidade e novos casos de covid por milhão de habitantes. O resultado é um gif com um mapa mundi para cada variável que mostra o comportamento e a evolução de cada uma dessas variáveis diariamente durante o primeiro semestre de 2020.

Matplotlib / Bokeh / Geopandas / imgkit / glob / moviepy

Hobbies:

Gosto de descobrir novas e interessantes maneiras de usar as ferramentas de programação e machine learning que aprendo constantemente para automatizar algum processo ou prever o resultado de algum evento. Estou constantemente buscando alternativas para gastar minha criatividade.

Código utilizado para a plotagem das barras de habilidades em linguagens:

```
In [1]: 1 ▾ def Plotando_Minhas_Habilidades_em_Linguagens():
2      %matplotlib inline
3      import matplotlib.pyplot as plt
4      import matplotlib.patches as patches
5      import numpy as np
6      import pandas as pd
7
8      plt.rcParams()
9
10     people = ('Python', 'R', 'C#', 'HTML', 'SQL', 'CSS', 'PySpark')
11     performance = [1243.0, 382.50, 423.31, 539.42, 571.33, 408.09, 516.11]
12
13     fig, ax = plt.subplots(1, 7, figsize = (16.32,2.0), sharey='row')
14     ▾ for i in range(0, len(people)):
15         circle = patches.Circle((0, 0), color = '#274E13', radius=performance[i]*0.008045)
16         ax[i].add_patch(circle)
17         ax[i].set_title(people[i], fontsize = 12, pad = 8)
18         ax[i].axes.set(xlim=(-12, 12), ylim=(-12, 12))
19         ax[i].axes.get_xaxis().set_visible(False)
20         ax[i].axes.get_yaxis().set_visible(False)
21         #ax[i].axes.yaxis.set_ticklabels([])
22     ▾ for pos in ['right', 'top', 'bottom', 'left']:
23         ax[i].spines[pos].set_visible(False)
24     plt.show()
25
```

executed in 19ms, finished 17:12:25 2021-01-14

Código utilizado para a plotagem das barras de conhecimento em bibliotecas:

```
In [2]: 1 ▾ def Plotando_Conhecimento_em_Bibliotecas():
2      %matplotlib inline
3      import matplotlib.pyplot as plt
4      import numpy as np
5      import pandas as pd
6
7      plt.rcParams()
8      fig, ax = plt.subplots(figsize = (8.32,2.34))
9
10     libs = ('Pandas', 'Scikit-learn', 'Matplotlib', 'Keras', 'PyTorch', 'Scipy', 'Selenium', 'pyautogui', 'geopandas')
11     y_pos = np.arange(len(libs))
12     performance = [1243.00, 1195.98, 1052.83, 841.94, 761.74, 999.0, 1100.0, 1100.0, 890.0]
13     perf_libs = dict(zip(libs,performance))
14     perf_libs = {k: v for k, v in sorted(perf_libs.items(), key=lambda item: item[1])}
15
16     ax.barh(y_pos, perf_libs.values(), color = '#63A541', align='center')
17     ax.set_yticks(y_pos)
18     ax.set_yticklabels(perf_libs.keys())
19     ax.invert_yaxis()
20     ax.axes.get_xaxis().set_visible(False)
21     ax.tick_params(axis=u'both', which=u'both',length=0)
22     ▾ for pos in ['right', 'top', 'bottom', 'left']:
23         fig.gca().spines[pos].set_visible(False)
```

executed in 15ms, finished 17:12:26 2021-01-14

Código para evitar o jupyter de plotar os outputs com scrolling:

```
In [22]: 1 ▾ %%javascript
2     ▾ IPython.OutputArea.prototype._should_scroll = function(lines) {
3         return false;
4     }
```

executed in 10ms, finished 17:22:40 2021-01-18

Código utilizado para a visualização, no pandas, da minha trajetória:

```

In [105]: 1 ▾ def carregando_textos(i):
2         import requests
3
4         url = 'https://curriculofiles.s3-sa-east-1.amazonaws.com/' + str(i) + '.txt'
5
6         response = requests.get(url)
7         response.encoding = 'utf-8'
8         item = response.text
9
10        return item
11
12 ▾ def Minha_Trajectoria_Projetada_no_Pandas(b, a=True):
13
14        from IPython.display import display
15        import pandas as pd
16        import textwrap
17
18        bold = "<b>"
19        reset = "</b>"
20
21
22        quem_sou_eu = carregando_textos('quem sou eu')
23        projetos = carregando_textos('projetos')
24        formacao = carregando_textos('formacao')
25        experiencia = carregando_textos('experiencia')
26        linguas = carregando_textos('linguas')
27        hobbies = carregando_textos('hobbies')
28
29        linhas = ['Quem sou eu: ', 'Formação: ', 'Experiência: ', 'Línguas: ', 'Projetos: ', 'Hobbies: ']
30        conteudo = [quem_sou_eu, formacao, experiencia, linguas, projetos, hobbies]
31
32        conteudo_corrigido = []
33
34 ▾        for n in conteudo:
35
36 ▾            if n in [linguas, formacao]:
37                conteudo_corrigido.append(n)
38                continue
39 ▾            else:
40                n = n.replace('\r\n', '').replace('<b>', ' <b>').replace('<i>', ' <i>').replace(' <', '\r\n <').replace('/b>', '/b>\r\n')
41
42                n = n.split('\n')
43
44                l = []
45
46                [[l.append(j) for j in textwrap.wrap(i,118)] if len(i)>118 else l.append(i) for i in n]
47
48                n = '\n'.join(['\xA0' + i for i in l])
49
50                conteudo_corrigido.append(n.replace('\t', ''))
51
52        exp = pd.DataFrame(data = conteudo_corrigido, index = linhas, columns = ['Sobre Mim'])
53
54        pd.options.display.max_colwidth = 2000
55
56 ▾        styles = [
57 ▾            dict(selector="tr:hover",
58                  props=[("background-color", "%s" % hover_color),
59 ▾                  dict(selector="th", props=[("font-size", "160%"),
60                  ("text-align", "left")]),
61                  dict(selector="caption", props=[("caption-side", "bottom")]),
62                  dict(selector = 'tr:nth-child(odd)',
63 ▾                  props= [('background', '#eee')]),
64                  dict(selector = 'tr:nth-child(even)',
65 ▾                  props= [('background', '#FEF8BD')]),
66                  dict(selector = 'th',
67 ▾                  props = [('background-color', 'white')]),
68                  dict(selector = 'td',
69 ▾                  props = [('font-family', 'verdana')]),
70                ]
71
72 ▾        if a and b == 1:
73 ▾            table = display(exp[:2].style.set_table_styles(styles).set_properties(**{
74                'font-size': '100%',
75                'text-align': 'left',
76                'white-space': 'pre-wrap',
77                'width': '1000px',
78            }))
79
80 ▾        if a and b == 2:
81            exp.columns = [''] * len(exp.columns)
82
83 ▾            table = display(exp[2:].style.set_table_styles(styles).set_properties(**{
84                'font-size': '100%',
85                'text-align': 'left',
86                'white-space': 'pre-wrap',
87                'width': '1000px',
88            }))
89
90        return table

```