

Kmeans_Agrupamentos_hard_skillv2

August 15, 2021

1 PROJETO

2 Agrupamento de Skills para grupo de treinamento entre colaboradores

O Objetivo do processo é buscar afinidade entre os conhecimento atuais dos colaboradores de uma empresa e definir Turmas onde eles possam trocar conhecimentos e se aprimorar.

Foi realizado um levantamento com colaboradores sobre seus conhecimentos e mapeados os níveis de conhecimento como material de fonte para o processo.

Para este processo foi levantado os dados dos colaboradores com o conhecimentos atuais e listados no arquivo em Excel, com as seguintes colunas:

Colaborador_gen: Indicação de cada colaborador (por questões de segurança os nomes foram anonimizados no arquivo fonte) Celula: Setor ou equipe que o colaborador pertence hoje Categoria: Área de conhecimento relacionado com o conhecimento que o colaborador possui Especialidade: Assunto em que o colaborador relaciona que está apto a aplicar Nivel: Nivel em que o colaborador se avalia no assunto

Definimos que o próprio colaborador se avalie para que possamos afinar o auto-conhecimento da equipe após os processos de troca de conhecimento.

Carregando e Explorando o Dataset

```
[1]: # Carrega os pacotes
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import OneHotEncoder
from sklearn.decomposition import PCA
from scipy.spatial.distance import cdist, pdist
%matplotlib inline
```

```
[2]: # Carrega o dataset
hardskills = pd.read_csv('20210629_Colabs_HardSkills.csv', sep = ';')
hardskills.head()
```

```
[2]:
```

	Colaborador_gen	Celula	Categoria	Especialidade	Nivel
0	Colaborador 1	TECH	Cloud	Azure	Avançado
1	Colaborador 1	TECH	Desenvolvimento	Script powershell	Avançado
2	Colaborador 1	TECH	Devops	Terraform	Avançado
3	Colaborador 1	TECH	Firewall	ForefrontTMG	Avançado
4	Colaborador 1	TECH	Firewall	Windows	Avançado

```
[3]: hardskills.columns
```

```
[3]: Index(['Colaborador_gen', 'Celula', 'Categoria', 'Especialidade', 'Nivel'],
dtype='object')
```

```
[4]: # Avaliei que a coluna referente a célula de trabalho é indiferente para a
      ↳ formação das turmas, pois o importante
      # é o conhecimento a ser compartilhado. Assim, foi efetuada a remoção da coluna:

hardskills = hardskills.drop(['Celula'],axis=1)
```

```
[5]: hardskills.shape
```

```
[5]: (537, 4)
```

```
[6]: # Validando se há algum valor nulo no dataset, o que seria um erro de
      ↳ preenchimento:
      #(True= Existe valor Nulo / False = Não existe Valor Nulo)
hardskills.isnull().values.any()
```

```
[6]: False
```

```
[7]: hardskills.dtypes
```

```
[7]: Colaborador_gen    object
     Categoria         object
     Especialidade     object
     Nivel             object
     dtype: object
```

```
[8]: hardskills.head()
```

```
[8]:
```

	Colaborador_gen	Categoria	Especialidade	Nivel
0	Colaborador 1	Cloud	Azure	Avançado
1	Colaborador 1	Desenvolvimento	Script powershell	Avançado
2	Colaborador 1	Devops	Terraform	Avançado
3	Colaborador 1	Firewall	ForefrontTMG	Avançado
4	Colaborador 1	Firewall	Windows	Avançado

Pré-processamento para variáveis categóricas - Encoding

```

[9]: ore = OneHotEncoder(handle_unknown='ignore')

[10]: hs_ore = ore.fit(hardskills.values)

[11]: hs_ore.categories_

[11]: [array(['Colaborador 1', 'Colaborador 10', 'Colaborador 11',
        'Colaborador 12', 'Colaborador 13', 'Colaborador 14',
        'Colaborador 15', 'Colaborador 16', 'Colaborador 17',
        'Colaborador 18', 'Colaborador 19', 'Colaborador 2',
        'Colaborador 20', 'Colaborador 21', 'Colaborador 3',
        'Colaborador 4', 'Colaborador 5', 'Colaborador 6', 'Colaborador 7',
        'Colaborador 8', 'Colaborador 9'], dtype=object),
array(['Automação', 'Backup', 'Cloud', 'Data Science', 'Database',
        'Desenvolvimento', 'DevOps', 'Devops', 'E-mail/Collab',
        'E-mails/Collab', 'Firewall', 'Firewalls', 'Hardware',
        'Infraestrutura', 'Monitoramento', 'Proxy', 'Redes', 'SO',
        'Segurança', 'Servidores', 'Storage', 'Telefonia'], dtype=object),
array(['3CX', 'AD', 'ASA Cisco', 'AWS', 'AWS EC2', 'AWS S3', 'Ansible',
        'Antivirus', 'Antivirus Trend', 'Antivirus KasperSky',
        'Antivirus Macaffe', 'Antivirus Sophos', 'Antivirus Trend',
        'Anydesk', 'Apache Hadoop', 'Apache Hive', 'Apache Mahout',
        'Apache Spark', 'ArchServer', 'Assembler', 'Azure', 'Azure Devops',
        'Bigbrother Xymon', 'C', 'C#', 'C++', 'CI/CD', 'CSS', 'Cabeamento',
        'CentOS', 'Certificado Digital', 'Checkpoint', 'Cisco',
        'Cisco ASA', 'Citrix Xen Server', 'Cobian', 'Controle de acessos',
        'DHCP', 'DNS', 'Data Bricks', 'Data Domain', 'Data Lake',
        'Data Protection', 'Data Protector', 'Debian', 'Dell', 'Delphi',
        'Digital Ocean', 'Digitro/Asterisk', 'Docker',
        'Docker / Container', 'Dokers Forms', 'ETL SQL Integration',
        'Endian Firewall', 'Estrutura de Redes', 'Exchange',
        'Failover Cluster', 'File Server', 'Firewall', 'Firewall Fortinet',
        'Firewall Sophos', 'ForefrontTMG', 'Fortinet', 'Gerenciamento',
        'Git', 'Git Lab ', 'GitHub', 'Google', 'Grafana', 'HP',
        'HP Data Protector', 'HTML', 'Hyper-V', 'IBM', 'IPTables', 'ISO',
        'Intelbrás', 'Java', 'JavaScript', 'Jenkins', 'Jquery', 'Junniper',
        'Kaspersky', 'Kubernets', 'LGPD', 'Linguagem SQL', 'Linux ',
        'Linux Arch', 'Linux Cent OS/Red Hat', 'Linux Debian',
        'Linux Kali', 'Linux Red Hat', 'Linux Server', 'Linux Suse',
        'Linux Ubuntu', 'Lógica', 'Lógica e Algoritmos', 'MABS', 'MAC',
        'MACOS', 'MCB', 'Manutenção', 'Manutenção de Celular',
        'Manutenção de Computadores', 'Manutenção de Servidores',
        'MongoDB', 'MonitoraIT', 'MySQL', 'NAS Lenovo', 'Nagios',
        'Office 365', 'Oracle', 'Oracle Linux', 'PHP', 'Packer',
        'Pascal/Delphi', 'Pentaho', 'Perfsense', 'PostgreSQL', 'Power App',
        'Power Apps', 'Power Automate', 'Power Automate ', 'Power BI',
        'Power Platform', 'PowerShell DSC', 'Proteção de identidade',

```

```

'Python', 'R', 'RDS', 'RMM', 'Raspberry Pi', 'React', 'Redes',
'Roteador Cisco', 'Roteador Tplink', 'Roteadores', 'S4', 'SIEM',
'SIEM Inteliacts', 'SPFX', 'SQL Server ', 'Script Powershell',
'Script bash', 'Script powershell', 'ScriptBash', 'Service Now',
'Sharepoint', 'SolarWinds', 'SonicWall', 'SonicWall VPN', 'Sophos',
'Speed', 'Switch Cisco', 'Switches', 'Switches Cisco', 'Synogie',
'TSM', 'TakeControl', 'Team Viewer', 'Teams', 'Terraform',
'TypeScript', 'URA', 'Unify', 'Unix Server', 'UnltraVNC', 'VB6',
'VBScript', 'VEEM', 'VMWare', 'VPN', 'VPN Servidor', 'Varonis',
'Wi-Fi', 'Windows', 'Windows Backup', 'Windows Firewall',
'Windows Server', 'Windows Server 2016', 'Windows VPN', 'Zabbix',
'Zention*', 'Zimbra'], dtype=object),
array(['Avançado', 'Iniciante', 'Intermediario'], dtype=object)]

```

```
[12]: hs_ore = ore.transform(hardskills.values).toarray()
```

```
[13]: hs_ore.shape
```

```
[13]: (537, 230)
```

```
[14]: hs_ore
```

```
[14]: array([[1., 0., 0., ..., 1., 0., 0.],
             [1., 0., 0., ..., 1., 0., 0.],
             [1., 0., 0., ..., 1., 0., 0.],
             ...,
             [0., 0., 0., ..., 0., 0., 1.],
             [0., 0., 0., ..., 0., 0., 1.],
             [0., 0., 0., ..., 0., 0., 1.]])
```

Determinando melhor valor de K com PCA – Com o objetivo de usar o K-means

```
[15]: # Aplica redução de dimensionalidade
pca = PCA(n_components = 2).fit_transform(hs_ore)
```

```
[16]: # Determinando um range de K
k_range = range(1,12)
```

```
[17]: # Aplicando o modelo K-Means para cada valor de K (esta célula pode levar
↳ bastante tempo para ser executada)
k_means_var = [KMeans(n_clusters = k).fit(pca) for k in k_range]
```

```
[18]: # Ajustando o centróide do cluster para cada modelo
centroids = [X.cluster_centers_ for X in k_means_var]
```

```
[19]: # Calculando a distância euclidiana de cada ponto de dado para o centróide
k_euclid = [cdist(pca, cent, 'euclidean') for cent in centroids]
```

```
dist = [np.min(ke, axis = 1) for ke in k_euclid]
```

```
[20]: # Soma dos quadrados das distâncias dentro do cluster  
soma_quadrados_intra_cluster = [sum(d**2) for d in dist]
```

```
[21]: # Soma total dos quadrados  
soma_total = sum(pdist(pca)**2)/pca.shape[0]
```

```
[22]: # Soma dos quadrados entre clusters  
soma_quadrados_inter_cluster = soma_total - soma_quadrados_intra_cluster
```

```
[23]: # Curva de Elbow  
fig = plt.figure()  
ax = fig.add_subplot(111)  
ax.plot(k_range, soma_quadrados_inter_cluster/soma_total * 100, 'b*-')  
ax.set_ylim((0,100))  
plt.grid(True)  
plt.xlabel('Número de Clusters')  
plt.ylabel('Percentual de Variância Explicada')  
plt.title('Variância Explicada x Valor de K')
```

```
[23]: Text(0.5, 1.0, 'Variância Explicada x Valor de K')
```



Clusterização em Python - Construção e Treinamento do Modelo KMeans

```
[24]: # Criando a versão do modelo a melhor quantidade de centroíde indicada pelo PCA
      ↪--> 3 clusters

modelo_v1 = KMeans(n_clusters = 3)
modelo_v1.fit(pca)
```

```
[24]: KMeans(n_clusters=3)
```

Definição de Turmas de trabalho (Clusters)

```
[25]: # Lista com nomes das colunas
names = hardskills.columns
names
```

```
[25]: Index(['Colaborador_gen', 'Categoria', 'Especialidade', 'Nivel'],
      dtype='object')
```

```
[26]: # Cria o cluster map
resultado = pd.DataFrame(hardskills, columns = names)
resultado['Colaborador_gen'] = pd.Categorical(resultado['Colaborador_gen'])
resultado['Turma'] = modelo_v1.labels_
```

```
[27]: resultado
```

```
[27]:
```

	Colaborador_gen	Categoria	Especialidade	Nivel	\
0	Colaborador 1	Cloud	Azure	Avançado	
1	Colaborador 1	Desenvolvimento	Script powershell	Avançado	
2	Colaborador 1	Devops	Terraform	Avançado	
3	Colaborador 1	Firewall	ForefrontTMG	Avançado	
4	Colaborador 1	Firewall	Windows	Avançado	
..	
532	Colaborador 9	Firewalls	Windows Firewall	Intermediario	
533	Colaborador 9	Monitoramento	Service Now	Intermediario	
534	Colaborador 9	Segurança	Antivirus KasperSky	Intermediario	
535	Colaborador 9	Segurança	Antivirus Trend	Intermediario	
536	Colaborador 9	Segurança	VPN	Intermediario	

	Turma
0	0
1	0
2	0
3	0
4	0
..	...
532	2
533	2

```
534      2
535      2
536      2
```

```
[537 rows x 5 columns]
```

```
[28]: resultado.to_csv("resultado.csv")
```

1. Tratando o resultado para demonstrar a melhor turma por colaborador O dataset mostrará a turma que o colaborador terá a melhor afinidade.

```
[29]: resultado_t1= resultado.drop(['Categoria', 'Especialidade', 'Nivel'],axis=1)
resultado_t1
```

```
[29]:
```

	Colaborador_gen	Turma
0	Colaborador 1	0
1	Colaborador 1	0
2	Colaborador 1	0
3	Colaborador 1	0
4	Colaborador 1	0
..
532	Colaborador 9	2
533	Colaborador 9	2
534	Colaborador 9	2
535	Colaborador 9	2
536	Colaborador 9	2

```
[537 rows x 2 columns]
```

```
[30]: resultado_t2 = resultado_t1.value_counts()
resultado_t2 = resultado_t2.to_frame()
resultado_t2 = resultado_t2.set_axis(['valores'], axis=1)
resultado_t2.head()
```

```
[30]:
```

	Colaborador_gen	Turma	valores
	Colaborador 6	0	35
	Colaborador 4	1	27
	Colaborador 5	1	23
	Colaborador 1	1	22
		2	20

```
[31]: resultado_t2.reset_index(inplace=True)
```

```
[32]: resultado_t2.head()
```

```
[32]: Colaborador_gen Turma valores
0    Colaborador 6      0      35
1    Colaborador 4      1      27
2    Colaborador 5      1      23
3    Colaborador 1      1      22
4    Colaborador 1      2      20
```

```
[33]: resultado_t3 = resultado_t2.
      ↪pivot(index='Colaborador_gen',columns='Turma',values='valores')
resultado_t3
```

```
[33]: Turma          0      1      2
Colaborador_gen
Colaborador 1      19.0  22.0  20.0
Colaborador 10      2.0   1.0   7.0
Colaborador 11     16.0   5.0   9.0
Colaborador 12      NaN   3.0   1.0
Colaborador 13      NaN  12.0  12.0
Colaborador 14      NaN   2.0   NaN
Colaborador 15      3.0   9.0  13.0
Colaborador 16      4.0   8.0   5.0
Colaborador 17      NaN  15.0   1.0
Colaborador 18      NaN  14.0   NaN
Colaborador 19      5.0  13.0   6.0
Colaborador 2      14.0   9.0  15.0
Colaborador 20      1.0  14.0   1.0
Colaborador 21      1.0  10.0   3.0
Colaborador 3       NaN  12.0   9.0
Colaborador 4       1.0  27.0  20.0
Colaborador 5      10.0  23.0  15.0
Colaborador 6      35.0   6.0  15.0
Colaborador 7       7.0   2.0  11.0
Colaborador 8       2.0  16.0   3.0
Colaborador 9       7.0  12.0   9.0
```

```
[34]: resultado_tfinal = resultado_t3.replace(np.nan,0)
```

```
[35]: resultado_tfinal
```

```
[35]: Turma          0      1      2
Colaborador_gen
Colaborador 1      19.0  22.0  20.0
Colaborador 10      2.0   1.0   7.0
Colaborador 11     16.0   5.0   9.0
Colaborador 12      0.0   3.0   1.0
Colaborador 13      0.0  12.0  12.0
Colaborador 14      0.0   2.0   0.0
```


Colaborador 15	3.0	9.0	13.0
Colaborador 16	4.0	8.0	5.0
Colaborador 17	0.0	15.0	1.0
Colaborador 18	0.0	14.0	0.0
Colaborador 19	5.0	13.0	6.0
Colaborador 2	14.0	9.0	15.0
Colaborador 20	1.0	14.0	1.0
Colaborador 21	1.0	10.0	3.0
Colaborador 3	0.0	12.0	9.0
Colaborador 4	1.0	27.0	20.0
Colaborador 5	10.0	23.0	15.0
Colaborador 6	35.0	6.0	15.0
Colaborador 7	7.0	2.0	11.0
Colaborador 8	2.0	16.0	3.0
Colaborador 9	7.0	12.0	9.0

```
[36]: resultado_tfinal.to_csv('resultado_turma_final.csv')
```

2.Tratando o resultado para determinar o assunto melhor aboradado em cada turma
O dataset mostrará a turma que o assunto terá melhor afinidade terá a melhor afinidade.

```
[45]: resultado_c1= resultado.drop(['Colaborador_gen', 'Especialidade', 'Nivel'],axis=1)
resultado_c1
```

```
[45]:
```

	Categoria	Turma
0	Cloud	0
1	Desenvolvimento	0
2	Devops	0
3	Firewall	0
4	Firewall	0
..
532	Firewalls	2
533	Monitoramento	2
534	Segurança	2
535	Segurança	2
536	Segurança	2

[537 rows x 2 columns]

```
[56]: resultado_c2 = resultado_c1.value_counts()
resultado_c2 = resultado_c2.to_frame()
resultado_c2 = resultado_c2.set_axis(['valores'], axis=1)
resultado_c2.head()
```

```
[56]:
```

		valores
Categoria	Turma	
Segurança	0	32

Servidores	1	28
	0	25
Desenvolvimento	2	25
Servidores	2	22

```
[57]: resultado_c2.reset_index(inplace=True)
```

```
[58]: resultado_c2.head()
```

```
[58]:
```

	Categoria	Turma	valores
0	Segurança	0	32
1	Servidores	1	28
2	Servidores	0	25
3	Desenvolvimento	2	25
4	Servidores	2	22

```
[59]: resultado_c3 = resultado_c2.  
      ↪pivot(index='Categoria',columns='Turma',values='valores')  
      resultado_c3
```

```
[59]:
```

Turma	0	1	2
Categoria			
Automação	NaN	NaN	1.0
Backup	2.0	18.0	10.0
Cloud	1.0	13.0	3.0
Data Science	NaN	2.0	2.0
Database	2.0	20.0	5.0
Desenvolvimento	9.0	22.0	25.0
DevOps	NaN	4.0	NaN
Devops	1.0	19.0	10.0
E-mail/Collab	1.0	4.0	6.0
E-mails/Collab	2.0	12.0	13.0
Firewall	8.0	6.0	3.0
Firewalls	6.0	8.0	5.0
Hardware	2.0	NaN	NaN
Infraestrutura	1.0	2.0	2.0
Monitoramento	2.0	11.0	14.0
Proxy	2.0	1.0	NaN
Redes	16.0	19.0	17.0
SO	13.0	19.0	12.0
Segurança	32.0	17.0	14.0
Servidores	25.0	28.0	22.0
Storage	2.0	5.0	6.0
Telefonia	NaN	5.0	5.0

```
[60]: resultado_cfina1 = resultado_c3.replace(np.nan,0)
```

```
[61]: resultado_cfina
```

```
[61]: Turma          0      1      2
      Categoria
      Automação      0.0    0.0    1.0
      Backup         2.0   18.0   10.0
      Cloud           1.0   13.0    3.0
      Data Science    0.0    2.0    2.0
      Database        2.0   20.0    5.0
      Desenvolvimento 9.0   22.0   25.0
      DevOps          0.0    4.0    0.0
      Devops          1.0   19.0   10.0
      E-mail/Collab   1.0    4.0    6.0
      E-mails/Collab  2.0   12.0   13.0
      Firewall        8.0    6.0    3.0
      Firewalls       6.0    8.0    5.0
      Hardware        2.0    0.0    0.0
      Infraestrutura  1.0    2.0    2.0
      Monitoramento  2.0   11.0   14.0
      Proxy           2.0    1.0    0.0
      Redes           16.0   19.0   17.0
      SO              13.0   19.0   12.0
      Segurança       32.0   17.0   14.0
      Servidores      25.0   28.0   22.0
      Storage         2.0    5.0    6.0
      Telefonia       0.0    5.0    5.0
```

```
[62]: resultado_cfina.to_csv('resultado_categoria_final.csv')
```

3.Tratando o resultado para definir a Especialidade de cada turma O dataset mostrará a especialidade que terá a melhor afinidade com cada turma.

```
[63]: resultado_e1= resultado.drop(['Categoria', 'Colaborador_gen', 'Nivel'],axis=1)
      resultado_e1
```

```
[63]:      Especialidade  Turma
0           Azure        0
1  Script powershell        0
2         Terraform        0
3  ForefrontTMG          0
4         Windows        0
..          ...      ...
532  Windows Firewall        2
533      Service Now        2
534  Antivirus KasperSky        2
535      Antivirus Trend        2
536             VPN          2
```

[537 rows x 2 columns]

```
[64]: resultado_e2 = resultado_e1.value_counts()
resultado_e2 = resultado_e2.to_frame()
resultado_e2 = resultado_e2.set_axis(['valores'], axis=1)
resultado_e2.head()
```

```
[64]:
```

Especialidade	Turma	valores
Windows	0	11
VEEM	1	7
Windows Server	1	6
Oracle	1	6
VEEM	2	6

```
[65]: resultado_e2.reset_index(inplace=True)
```

```
[66]: resultado_e2.head()
```

```
[66]:
```

	Especialidade	Turma	valores
0	Windows	0	11
1	VEEM	1	7
2	Windows Server	1	6
3	Oracle	1	6
4	VEEM	2	6

```
[67]: resultado_e3 = resultado_e2.
      ↪pivot(index='Especialidade', columns='Turma', values='valores')
resultado_e3
```

```
[67]:
```

Turma	0	1	2
Especialidade			
3CX	NaN	5.0	NaN
AD	3.0	2.0	3.0
ASA Cisco	NaN	NaN	1.0
AWS	NaN	4.0	2.0
AWS EC2	NaN	1.0	NaN
...
Windows Server 2016	NaN	NaN	1.0
Windows VPN	1.0	NaN	NaN
Zabbix	NaN	5.0	4.0
Zention*	NaN	1.0	NaN
Zimbra	NaN	NaN	1.0

[184 rows x 3 columns]

```
[68]: resultado_efinal = resultado_e3.replace(np.nan,0)
```

```
[69]: resultado_efinal
```

```
[69]: Turma          0    1    2
Especialidade
3CX              0.0  5.0  0.0
AD               3.0  2.0  3.0
ASA Cisco        0.0  0.0  1.0
AWS              0.0  4.0  2.0
AWS EC2          0.0  1.0  0.0
...
Windows Server 2016 0.0  0.0  1.0
Windows VPN         1.0  0.0  0.0
Zabbix             0.0  5.0  4.0
Zention*           0.0  1.0  0.0
Zimbra             0.0  0.0  1.0
```

```
[184 rows x 3 columns]
```

```
[70]: resultado_efinal.to_csv('resultado_especialidade_final.csv')
```

```
[ ]:
```