questaoquatro

October 17, 2018

1 Questão 4 - Prova 2 de Inteligência Artificial

1.1 Lucas Nóbrega e Nathália de Vasconcelos

Enunciado: utilizando a base de clusterização encontrada nesse link (https://drive.google.com/file/d/1_702eOQbimT1HhTuozwMEKV6HHOykLaJ/view), execute os algoritmos de clusterização citados a seguir e compare os resultados.

- Execute os algoritmos de agrupamento K-means e Hierárquico com os seguintes valores de K: 2, 5, 10 e 100. Compare os agrupamento resultantes dos 2 algoritmos.
- Escolha um número fixo de K e altere o parâmetro do K-Means referente ao número máximo de iterações: 1, 10 e 100 e o parâmetro de Linkage do Hierárquico, quais diferenças puderam ser observadas?
- Faça uma comparação entre os 2 algoritmos, qual você acha que teve o melhor desempenho e por quê?

2 Pré-processamento de dados

2.0.1 Importando o dataset da questão

```
In [1]: import scipy.io.arff as io
    import pandas as pd
    import matplotlib.pyplot as plt

//matplotlib inline

import numpy as np

# Aumentar o tamanho do plot na proporção 8/13
x_size = 17
y_size = x_size * (8/13)
plt.figure(figsize=(x_size, y_size))
# Aumentar o tamanho do plot na proporção 8/13

# Read arff data
with open("Genes_Atividade_IA.arff") as f:
    values = io.loadarff(f)
```

f.close()

```
dataset = pd.DataFrame(data=values[0])
#X = dataset.iloc[:, :-1].values
#y = dataset.iloc[:, 20].values
```

<Figure size 1224x753.231 with 0 Axes>

2.0.2 Melhorando a visualização da tabela do dataset

In [3]: dataset.describe()

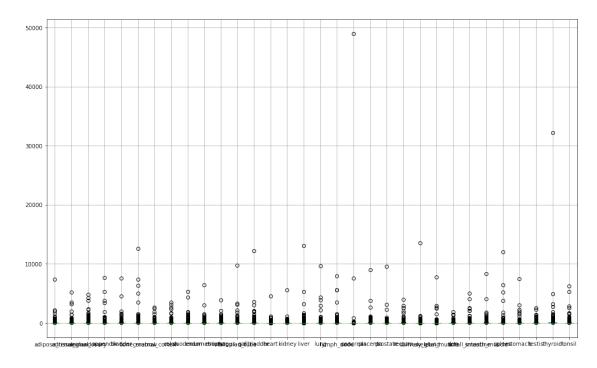
Out[3]:		adipose_tissue	e adrenal_gland	animal_ovary	appendix	bladder	\
	count	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	
	mean	39.040157	41.486336	51.338392	2 52.555181	48.932735	
	std	195.264717	7 181.466441	210.726346	252.295704	221.732356	
	min	0.000003	0.000003	0.000004	0.000004	0.00004	
	25%	3.173068	3.105311	3.140023	7.117802	7.087438	
	50%	12.692271	15.526556	19.625143	3 21.353406	21.262314	
	75%	31.730678	34.158423	43.175314	46.265713	42.524627	
	max	7377.382624	5216.922745	4827.785071	7644.519332	7548.121360	
		bone_marrow	cerebral_cortex	colon	duodenum	endometrium	\
	count	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	
	mean	53.852077	45.797667	40.663197	39.632611	52.066032	
	std	393.758129	123.183344	157.028175	185.991398	186.169270	
	min	0.000003	0.000004	0.000003	0.000003	0.000004	
	25%	1.257345	7.631545	3.071134	2.765053	7.428423	
	50%	9.430086	19.078863	15.355670	13.825265	21.224067	
	75%	34.576981	41.973499	33.782474	30.415583	50.937760	
	max	12623.741610	2663.409328	3528.732999	5275.721159	6460.605882	
		esophagus i	fallopian_tube	gallbladder	heart	kidney	\
	count	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	
	mean	41.493874	53.691974	57.023709	21.915492	27.728612	
	std	139.247221	261.725555	314.203534	116.726607	136.411996	
	min	0.000004	0.000004	0.00004	0.000002	0.000002	
	25%	3.506638	7.896982	8.147757	1.569051	2.492157	
	50%	14.026553	23.690945	20.369393	6.724503	9.968628	

75%	35.066382	47.38189	0 48.886	544 15.690	507	24.921571	
max	3857.301972	9733.02999	2 12201.266	550 4512.141	526 55	87.416115	
	liver	lung	lymph_node	pancreas	_	lacenta \	
count	2000.000000	2000.000000	2000.000000	2000.000000		.000000	
mean	35.790131	54.864549	54.294117	35.087462		. 105545	
std	336.523288	277.654367	277.993904	1108.11440		. 242244	
min	0.000003	0.000004	0.000004	0.00000		.000004	
25%	1.170095	7.812597	3.751913	0.602562		.128284	
50%	5.850474	23.437791	18.759564	2.008539		.641418	
75%	17.551423	46.875582	48.774865	6.025616		.411120	
max	13058.258680	9707.151804	7931.543465	48970.179710	9032	. 684565	
	prostate	rectum	salivary_gla	nd skeletal_r	nuscla	skin	\
count	2000.000000	2000.000000	2000.00000		000000	2000.000000	`
mean	46.389605	42.376928	27.6287)53380	35.257338	
std	241.613858	165.470468	308.54356		551352	95.223057	
min	0.000004	0.000003	0.0000		000002	0.000003	
25%	7.150531	3.094211	2.4140		127236	3.304699	
50%	17.876327	15.471057	8.04674		136179	16.523495	
75%	42.903184	34.036326	18.7757		544717	36.351689	
max	9585.286431	3945.119614	13529.2582		968610	1939.858329	
	00001200101		1001011001	.,,			
	small_intesti	ine smooth_mu	scle	spleen st	comach	testis	\
count	2000.0000	2000.00	0000 2000.0	000000 2000.0	00000	2000.000000	
mean	43.3968	344 52.18	5982 56.	164531 37.	587312	44.472439	
std	193.6381	153 233.84	8790 347.	561819 207.0	009834	114.878046	
min	0.0000	0.00	0.004	0.00004	00003	0.000003	
25%	5.9839	976 4.24	8427 3.6	678518 5.8	374027	6.997996	
50%	14.9599	941 21.24	2137 22.0	071107 14.6	885068	20.993988	
75%	32.9118	370 42.48	4273 47.8	320732 29.3	370135	45.486973	
max	5032.5241	122 8322.66	9088 12036.	110500 7486.4	147452	2554.268510	
	thyroid	tonsil					
count	2000.000000	2000.000000					
mean	72.736881	46.143284					
std	742.846516	222.768691					
min	0.000005	0.000003					
25%	4.580134	3.174720					
50%	22.900670	15.873599					
75%	54.961609	38.096639					
max	32253.304180	6231.975147					

2.0.3 Visualizando o dataset através de um box plot, a fim de encontrar possíveis outliers

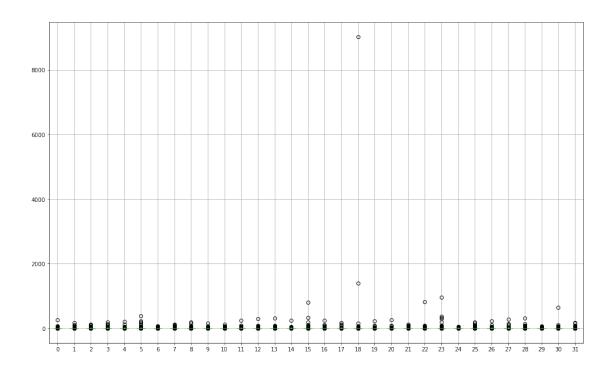
```
plt.figure(figsize=(x_size, y_size))
# Aumentar o tamanho do plot na proporção 8/13
dataset.boxplot()
```

Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x7efced3770b8>



2.0.4 Detectamos outliers, logo faz-se necessário um ajuste na escala

2.0.5 Visualizando o boxplot depois de ter sua escala reajustada, percebe-se a presença de um *outlier* na coluna 18...



2.0.6 Tentando localizar a instância com um valor acima de 9000

```
In [9]: a = df.loc[df[18] > 9000]
In [10]: a
Out [10]:
                          1
                                     2
                                          3
                                                4
                                                          5
         1940 -0.377778 -0.5 0.490196 -0.5 -0.59 -0.283019 -0.555555 -0.49 0.2
                               10
                                      11
                                           12
                                                    13
                                                              14
                                                                         15
                                                                               16
         1940 -0.478049 -0.444444 -0.58 -0.5 0.15873 -0.444444 -0.357143 -0.58
                     17
                                   18
                                        19
                                              20
                                                   21
                                                             22
                                                                        23
                                                                              24 \
         1940 -0.408333 9029.629629 0.4 -0.49 -0.5 0.655738 -0.236842 -0.45
                     25
                               26
                                                                      30
                                          27
                                                 28
                                                           29
                                                                                31
         1940 -0.111111 -0.544444 -0.416667 -0.375 -0.472727 -0.454545 -0.454545
In [11]: aux = np.array(df[18])
```

2.0.7 Substituindo o valor desse outlier pela mediana da coluna 18 do dataset

In [12]: df[18] = df[18].replace(9029.629629, np.nanmedian(aux))

3 Identificando o número de classes

```
In [13]: class_array = dataset.iloc[:, 32]
```

3.0.1 Significa que existem 2000 classes, ou seja, uma para cada instância.

In [17]: X = dataset.iloc[:, :-1].values

Out[19]: 1

4 1. Execução dos algoritmos de aprendizagem de máquina K-means e Hierárquico para resolver um problema de agrupamento

Execute os algoritmos de agrupamento K-means e Hierárquico com os seguintes valores de K: 2, 5, 10 e 100. Compare os agrupamentos resultantes dos 2 algoritmos.

4.0.1 Separando as entradas do problema na variável X e a saída esperada do problema na variável y

4.1.1 Perceba que a soma da quantidade de rótulos foi igual a 1 devido ao *outlier* na coluna "pancreas" que causou um desbalanceamento nas classes

```
Out [20]:
            Classe Nž Instancias
         0
                 0
                             1999
         1
                 1
                                 1
In [21]: kmeans.cluster_centers_[0]
Out [21]: array([39.05873463, 41.50708976, 51.34443912, 52.58058144, 48.95703592,
                53.87901696, 45.82057733, 40.68338489, 39.64275479, 52.09186528,
                41.51463154, 53.7184383 , 57.05223534, 21.9219698 , 27.74248304,
                35.80803475, 54.89160398, 54.32108966, 10.60767637, 52.11302411,
                46.41263207, 42.39812701, 27.63319137, 22.06430544, 35.27414936,
                43.41256626, 52.21187541, 56.19078705, 37.60317633, 44.49328609,
                72.77326733, 46.16636715])
               confirmado pois
                                  é possível
                                                ver
                                                     que
                                                           ele
                                                                foi
                                                                      capturado
                                                                                 no
     kmeans.cluster_centers_[1] (valor 4.89e+4)
In [22]: kmeans.cluster_centers_[1]
Out [22]: array([1.90384068e+00, 3.10531000e-06, 3.92502851e+01, 1.77945050e+00,
                3.54371895e-01, 3.14336000e-06, 3.81577000e-06, 3.07113403e-01,
                1.93553711e+01, 4.24481333e-01, 3.50664000e-06, 7.89698174e-01,
                4.07388000e-06, 8.96600403e+00, 2.49216000e-06, 2.92524000e-06,
                7.81259703e-01, 3.75191271e-01, 4.89701797e+04, 3.71545526e+01,
                3.57526536e-01, 3.09421000e-06, 1.87757351e+01, 2.13617917e-01,
                1.65234951e+00, 1.19679527e+01, 4.24842730e-01, 3.67851788e+00,
                5.87402703e+00, 2.79919837e+00, 4.58013000e-06, 3.17472000e-06])
4.1.3 Adicionando a coluna calculada kmeans.label_ ao conjunto de dados
In [23]: dataset["K-classes"] = kmeans.labels_
4.1.4 Provado que a única instância da classe 1 que contém o valor com outlier, ou seja, o
     outilier interfere diretamente na análise de dados.
In [24]: dataset.loc[ dataset["pancreas"] > 40000]
Out [24]:
                               adrenal_gland animal_ovary
               adipose_tissue
                                                             appendix
                                                                         bladder \
                                                  39.250285
         1940
                     1.903841
                                     0.000003
                                                              1.77945
                                                                       0.354372
               bone_marrow cerebral_cortex
                                                 colon
                                                         duodenum
                                                                   endometrium
         1940
                  0.000003
                                    0.000004 0.307113
                                                        19.355371
                                                                       0.424481
               esophagus
                         fallopian_tube gallbladder
                                                           heart
                                                                    kidney
                                                                                liver
         1940
                0.000004
                                 0.789698
                                              0.000004 8.966004 0.000002 0.000003
                  lung lymph_node
                                        pancreas
                                                   placenta prostate
                                                                         rectum \
```

1940

0.78126

0.375191 48970.17971 37.154553 0.357527

0.000003

```
salivary_gland skeletal_muscle
                                         skin small_intestine \
1940
          18.775735
                            0.213618 1.65235
                                                     11.967953
     smooth_muscle
                      spleen
                                          testis
                                                   thyroid
                                                             tonsil \
                               stomach
          0.424843 3.678518 5.874027
                                        2.799198 0.000005 0.000003
1940
                  class K-classes
1940 b'ENSG00000091704'
```

4.2 K-means, K = 5

4.2.1 Analisando os rótulos de saída do algoritmo, percebe-se que temos 3 classes, de forma que a classe 0 é afetada pelo desbalanceamento das classes devido ao *outlier* presente na coluna "pancreas"

```
In [27]: unique, counts = np.unique(a, return_counts=True)
         dicionario = dict(zip(unique, counts))
         print(dicionario)
         lista = list(dicionario.items())
         class_instances = pd.DataFrame(lista, columns=["Classe", "Nž Instancias"])
         class_instances
{0: 1995, 1: 1, 2: 2, 3: 1, 4: 1}
Out [27]:
            Classe Nž Instancias
         0
                 0
                              1995
         1
                 1
                                 1
         2
                 2
                                 2
         3
                 3
                                 1
                                 1
```

- 4.2.2 A quantidade de instâncias de cada classe afetou diretamente na classificação das classes, devido ao número de *outliers* nas amostras.
- 4.2.3 Adicionando a coluna calculada kmeans.label_ ao conjunto de dados

```
In [28]: dataset["K-classes"] = kmeans.labels_
In [29]: dataset.loc[dataset["K-classes"] > 0]
Out [29]:
              adipose_tissue adrenal_gland animal_ovary
                                                                             bladder \
                                                               appendix
         591
                     0.317307
                                    1.863187
                                                 0.000004
                                                               0.711780
                                                                            0.708744
         1315
                 2027.590321
                                 1400.495332
                                               2296.141680 7644.519332 4514.697940
         1718
                 7377.382624
                                3490.369741
                                              4827.785071 5295.644677 7548.121360
```

1874 1940	28.5576 1.9038			54.950399 39.250289		836.317672 0.354372	
1940	1.9030	41 0.000	003	39.230200	1.779450	0.354372	
	bone_marrow	cerebral_cor	tex	colon	duodenum	endometrium \	\
591	0.628672			0.307113	0.000003	0.00004	
1315	3463.984873	2430.647	195 23	346.346398	1800.049515	6460.605882	
1718	6349.591147	1938.412	520 30	095.703101	4330.073027	2996.838208	
1874	12623.741610	19.078	863 5	546.661857	5275.721159	114.609960	
1940	0.000003	0.000	004	0.307113	19.355371	0.424481	
	esophagus	fallopian_tub	e gal	llbladder	heart	kidney \setminus	
591	0.350664	0.39484	9	0.814776	6.724503	1.246079	
1315	3857.301972	3293.04138	5 363	37.973632		799.982414	
1718	2114.502808	3158.79269		01.266550	1187.995534 5	587.416115	
1874	199.878375	110.55774	4 305	55.408986	26.898012	32.398042	
1940	0.000004	0.78969	8	0.000004	8.966004	0.000002	
	liver	lung	lymph_		-	lacenta \	
591	0.000003	0.390630	26.26			.412828	
1315	1292.954824		7931.54			.020932	
1718	5344.408289		5612.86			.684565	
1874	125.785198		1485.75			. 285898	
1940	0.000003	0.781260	0.37	75191 4897	70.179710 37	. 154553	
			. .	, ,			,
504	prostate				skeletal_muscle		\
591	0.715053	0.309421		1.341124	0.000002		
1315	3128.357190	2558.912879		2.431712	183.711409		
1718	2288.169831	3945.119614		6.777824	324.699234		
1874	60.779511	519.827526		9.258260	2.136179		
1940	0.357527	0.000003	18	3.775735	0.213618	1.652350	
	small_intest	ine smooth_mu	aala	splee	en stomach	testis	\
591	0.000	-		0.36785		2.449299	\
1315	2600.037			5201.42428			
1718	5032.524			3201.42420 12036.11050			
1874	2058.487			1364.73013			
1940	11.967			3.6785			
1940	11.907	900 0.42	4043	3.0765.	10 5.014021	2.799190	
	thyroid	tonsil		cla	ass K-classes		
591	32253.304180	0.952416	b'ENS(30000004283			
1315	2729.759911	6231.975147		G0000004260 G0000007562			
1718	4919.064000	2809.627104		G0000007302			
1874	87.022547			G0000000100			
1940	0.000005	0.000003		G0000009030 G0000000917(
	0.000000	0.00000	~ -110	~~~~~~~~~~~	·		

- 4.2.4 Acima podemos ver as instâncias que pertencem às demais classes. Em cada uma delas existe um *outlier* que altera o comportamento do algoritmo.
- 4.3 K-means, K = 10

4.4 Analisando os rótulos de saída do algoritmo, percebem-se 3 classes, de forma que a classe 1 continua sendo afetada pelo desbalanceamento de classes causado pelo *outlier* presente na coluna "pancreas"

```
Out[32]:
              Classe Nž Instancias
          0
                    0
                                  1925
          1
                    1
                                      1
          2
                    2
                                      1
          3
                    3
                                      1
          4
                    4
                                      1
          5
                    5
                                      1
          6
                                     66
                    6
          7
                    7
                                      1
                                      2
          8
                    8
          9
                    9
                                      1
```

- 4.4.1 Podemos perceber uma melhoria, com a classe 6 possuindo 66 instâncias, mas pode ser que todas elas sejam *outliers*.
- 4.4.2 Adicionando a coluna calculada kmeans.label_ ao conjunto de dados

```
In [33]: dataset["K-classes"] = kmeans.labels_
In [34]: dataset.loc[(dataset["K-classes"] > 0) & (dataset["K-classes"] != 6)]
Out [34]:
              adipose_tissue adrenal_gland animal_ovary
                                                             appendix
                                                                           bladder \
        393
                  621.921288
                                 586.903809
                                              726.130275 3459.251765 1998.657487
        510
                 2138.647694
                                 794.959656
                                             3768.027372 3800.906260 1587.586089
        591
                    0.317307
                                   1.863187
                                                0.000004
                                                             0.711780
                                                                          0.708744
                                             2296.141680 7644.519332 4514.697940
        1315
                 2027.590321
                             1400.495332
```

1657	1.90384	41 1.552	2656	3.92	25029	2.847	121	3.5437	19	
1718	7377.38262	24 3490.369	9741	4827.78	35071	5295.644	1677	7548.1213	60	
1874	28.5576	10 46.579	9667	54.95	50399	1854.187	417	836.3176	72	
1933	0.31730	0.000	0003	0.00	00004	0.000	0004	0.0000	04	
1940	1.90384	41 0.000	0003	39.25	50285	1.779	9450	0.3543	72	
	bone_marrow	cerebral_co	rtex	CC	olon	duoder	nım e	ndometriu	m \	
393	644.389201	450.26		786.210		1476.5383		827.73859		
510	1518.243824	2663.409		3528.732		1139.2018		073.93777		
591	0.628672	0.38		0.307		0.0000		0.00000		
1315	3463.984873	2430.647		2346.346		1800.0495		460.60588		
1657	3.143362	3.81		2.764		2.4885		16.97925		
1718	6349.591147	1938.412		3095.703		4330.0730		996.83820		
1874	12623.741610	19.078		546.661		5275.7211		114.60996		
1933	0.000003	0.000		1.228		2.7650		0.00000		
1940	0.000003	0.000	0004	0.307	7113	19.3553	371	0.42448	1	
	esophagus	fallopian_tub	ре	gallbladd	der	hear	rt	kidney	\	
393	511.969171	1772.87240	00	2049.1609	960	179.32008	30 4	53.572584		
510	999.391875	1113.47442	25	1861.7625	542	342.94965	54 4	95.939254		
591	0.350664	0.39484	19	0.8147	776	6.72450)3	1.246079		
1315	3857.301972	3293.04138	35	3637.9736	332	428.12669	92 7	99.982414		
1657	2.805311	9733.02999	92	4.0738	379	0.67245	50	2.492157		
1718	2114.502808	3158.79269		12201.2665	550	1187.99553	34 55	87.416115		
1874	199.878375	110.55774		3055.4089		26.89801		32.398042		
1933	0.000004	0.78969		8.1477		0.00000		34.890199		
1940	0.000004	0.78969		0.0000		8.96600		0.000002		
1010	0.00001	0.7000		0.000	, , ,	0.0000	, -	0.000002		
	liver	lung	1,	mph_node		pancreas	n	lacenta	\	
393	532.393163	3921.923707	-	12.861412		70.298851	-	.775824	`	
510	476.813657	2925.817586		19.309422		48.631857		.203533		
591	0.000003	0.390630		26.263389	1	0.000002		.412828		
					1					
1315	1292.954824	4394.585827		31.543465		88.802629		.020932		
1657	5.850474			7.503825		0.200854		. 256567		
	5344.408289									
	125.785198					18.076847		. 285898		
1933	13058.258680			0.000004				.000004		
1940	0.000003	0.781260		0.375191	489	70.179710	37	. 154553		
	1	rectum		v – v		keletal_mı				\
393		1079.879800				66.22				
510	793.708910	2558.912879		482.80461	17	123.89	98392	343.688	699	
591	0.715053	0.309421		1.34112	24	0.00	00002	2.313	289	
1315	3128.357190	2558.912879		992.43171	12	183.71	1409	1391.278	291	
1657	7.150531	6.188423		2.14579	98	1.06	8090	9.914	097	
1718	2288.169831	3945.119614	1	1166.77782	24	324.69	9234	1311.965	514	
1874		519.827526		3529.25826			36179			
	0.000004	0.928263		0.00000			00002			

```
1940
                  0.357527
                                0.000003
                                               18.775735
                                                                  0.213618
                                                                               1.652350
               small_intestine
                                smooth_muscle
                                                      spleen
                                                                   stomach
                                                                                 testis
         393
                   2133.287574
                                   1291.521900
                                                 6463.155915
                                                                             199.442884
                                                              1527.247028
         510
                   2456.422297
                                   1797.084749
                                                 3833.015631
                                                              1715.215893
                                                                            1091.687363
         591
                      0.000003
                                      0.424843
                                                    0.367852
                                                                  0.000003
                                                                               2.449299
         1315
                   2600.037730
                                   8322.669088
                                                 5201.424283
                                                              2232.130272
                                                                            1137.174337
                                                                              17.494990
         1657
                      2.094392
                                      4.248427
                                                    3.678518
                                                                  2.643312
                   5032.524122
                                                12036.110500 3072.116138
         1718
                                   4112.477630
                                                                           1445.086157
         1874
                   2058.487869
                                    195.427656
                                                 1364.730134 7486.447452
                                                                              24.492986
         1933
                      1.196795
                                      0.000004
                                                    0.000004
                                                                  5.874027
                                                                               0.349900
         1940
                     11.967953
                                      0.424843
                                                    3.678518
                                                                  5.874027
                                                                               2.799198
                    thyroid
                                   tonsil
                                                        class
                                                               K-classes
                 609.157832
         393
                             5352.577737
                                           b'ENSG0000019582'
         510
                 751.141989
                             2533.426473
                                           b'ENSG0000034510'
                                                                        8
         591
               32253.304180
                                 0.952416
                                           b'ENSG00000042832'
                                                                        4
         1315
                2729.759911
                             6231.975147
                                           b'ENSG00000075624'
                                                                        5
         1657
                                 3.174720 b'ENSG00000085465'
                                                                        7
                   1.832054
                                                                        2
         1718
                4919.064000 2809.627104 b'ENSG00000087086'
         1874
                  87.022547
                             1561.962187
                                           b'ENSG00000090382'
                                                                        3
                                                                        9
         1933
                   0.000005
                                 0.000003
                                           b'ENSG00000091583'
         1940
                   0.000005
                                 0.000003 b'ENSG00000091704'
                                                                        1
4.5 K-means, K = 100
In [35]: from sklearn.cluster import KMeans
         kmeans = KMeans(n_clusters=100, random_state=0).fit(X)
In [36]: a = np.array(kmeans.labels_)
In [37]: unique, counts = np.unique(a, return_counts=True)
         dicionario = dict(zip(unique, counts))
         print(dicionario)
         lista = list(dicionario.items())
         class_instances = pd.DataFrame(lista, columns=["Classe", "Nž Instancias"])
         class_instances
{0: 38, 1: 1, 2: 1, 3: 1, 4: 1, 5: 1, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1, 11: 1, 12: 1, 13: 1, 14:
Out[37]:
             Classe
                     Nž Instancias
                  0
                                 38
         0
```

1

1

1

1

1

1

1

2

3

4

5

6

1

2

3

4

5

6

7 7 1 8 8 1 9 9 1 10 10 1 11 11 1 11 11 1 12 12 1 13 13 1 14 14 1 1 15 15 1 1 16 16 86 1 17 17 1 1 18 18 1 1 19 19 1 2 20 20 1 1 21 21 1 1 22 22 13 3 23 23 1 1 22 22 13 3 23 23 1 2 24 24 2 2 25 25 1 2 26 26 1 1 27 27 1 1 72			
8 8 1 9 9 1 10 10 1 11 11 11 11 11 1 12 12 1 13 13 1 14 14 1 1 15 15 1 1 16 16 86 1 17 17 1 1 18 18 1 1 19 19 1 2 20 20 1 1 21 21 21 1 22 22 23 1 24 24 2 2 25 25 1 1 26 26 1 1 27 27 1 1 28 28 1 1 70 70 2 2 71 71 1 1 72 72 47 1 <	7	7	1
9 9 1 10 10 1 11 11 1 12 12 1 13 13 1 14 14 1 15 15 1 16 16 86 17 17 1 18 18 1 19 19 1 20 20 1 21 21 1 22 22 13 23 23 1 24 24 2 25 25 1 26 26 1 27 27 1 28 28 1 29 29 4 70 70 2 71 71 1 72 72 47 73 73 12 74 74 4 4 74 74			
10 10 1 11 11 1 12 12 1 13 13 1 14 14 1 15 15 1 16 16 86 17 17 1 18 18 1 19 19 1 20 20 1 21 21 1 22 22 13 23 23 1 24 24 2 25 25 1 26 26 1 27 27 1 28 28 1 29 29 4 70 70 2 71 71 1 72 72 47 73 73 12 74 74 4 75 75 1 76 76 8			
11 11 1 12 12 1 13 13 1 14 14 1 15 15 1 16 16 86 17 17 1 18 18 1 19 19 1 20 20 1 21 21 1 22 22 13 23 23 1 24 24 2 25 25 1 26 26 1 27 27 1 28 28 1 29 29 4 70 70 2 71 71 1 72 72 47 73 73 12 74 74 4 75 75 1 76 76 8 77 77 2			
12 12 1 13 13 1 14 14 1 15 15 1 16 16 86 17 17 1 18 18 1 19 19 1 20 20 1 21 21 1 22 22 13 23 23 1 24 24 2 25 25 1 26 26 1 27 27 1 28 28 1 29 29 4 70 70 2 71 71 1 72 72 47 73 73 12 74 74 4 75 75 1 76 76 8 77 77 2 78 78 1	10	10	1
12 12 1 13 13 1 14 14 1 15 15 1 16 16 86 17 17 1 18 18 1 19 19 1 20 20 1 21 21 1 22 22 13 23 23 1 24 24 2 25 25 1 26 26 1 27 27 1 28 28 1 29 29 4 70 70 2 71 71 1 72 72 47 73 73 12 74 74 4 75 75 1 76 76 8 77 77 2 78 78 1	11	11	1
13 13 1 14 14 1 15 15 1 16 16 86 17 17 1 18 18 1 19 19 1 20 20 1 21 21 1 22 22 13 23 23 1 24 24 2 25 25 1 26 26 1 27 27 1 28 28 1 29 29 4 70 70 2 71 71 1 72 72 47 73 73 12 74 74 4 75 75 1 76 76 8 77 77 2 78 78 1 80 80 1			
14 14 1 15 15 1 16 16 86 17 17 1 18 18 1 19 19 1 20 20 1 21 21 1 22 22 13 23 23 1 24 24 2 25 25 1 26 26 1 27 27 1 28 28 1 29 29 4 70 70 2 71 71 1 72 72 47 73 73 12 74 74 4 75 75 1 76 76 8 77 77 2 78 78 1 80 80 1 81 81 15			
15 15 1 16 16 86 17 17 1 18 18 1 19 19 1 20 20 1 21 21 1 22 22 13 23 23 1 24 24 2 25 25 1 26 26 1 27 27 1 28 28 1 29 29 4 70 70 2 71 71 1 72 72 47 73 73 12 74 74 4 75 75 1 76 76 8 77 77 2 78 78 1 80 80 1 81 81 15 82 82 8			
16 16 86 17 17 1 18 18 1 19 19 1 20 20 1 21 21 1 22 22 13 23 23 1 24 24 2 25 25 1 26 26 1 27 27 1 28 28 1 29 29 4 70 70 2 71 71 1 72 72 47 73 73 12 74 74 4 75 75 1 76 76 8 77 77 2 78 78 1 80 80 1 81 81 15 82 82 8 83 83 1	14	14	1
16 16 86 17 17 1 18 18 1 19 19 1 20 20 1 21 21 1 22 22 13 23 23 1 24 24 2 25 25 1 26 26 1 27 27 1 28 28 1 29 29 4 70 70 2 71 71 1 72 72 47 73 73 12 74 74 4 75 75 1 76 76 8 77 77 2 78 78 1 80 80 1 81 81 15 82 82 8 83 83 1	15	15	1
17 17 1 18 18 1 19 19 1 20 20 1 21 21 1 22 22 13 23 23 1 24 24 2 25 25 1 26 26 1 27 27 1 28 28 1 29 29 4 70 70 2 71 71 1 72 72 47 73 73 12 74 74 4 75 75 1 76 76 8 77 77 2 78 78 1 79 79 160 80 80 1 81 81 15 82 82 8 83 83 1	16		86
18 18 1 19 19 1 20 20 1 21 21 1 22 22 13 23 23 1 24 24 2 25 25 1 26 26 1 27 27 1 28 28 1 29 29 4 70 70 2 71 71 1 72 72 47 73 73 12 74 74 4 75 75 1 76 76 8 77 77 2 78 78 1 79 79 160 80 80 1 81 81 15 82 82 8 83 83 1 84 84 1			
19 19 1 20 20 1 21 21 1 22 22 13 23 23 1 24 24 2 25 25 1 26 26 1 27 27 1 28 28 1 29 29 4 70 70 2 71 71 1 72 72 47 73 73 12 74 74 4 75 75 1 76 76 8 77 77 2 78 78 1 80 80 1 81 81 15 82 82 8 83 83 1 84 84 1 85 85 1 86 86 1			
20 20 1 21 21 1 22 22 13 23 23 1 24 24 2 25 25 1 26 26 1 27 27 1 28 28 1 29 29 4 70 70 2 71 71 1 72 72 47 73 73 12 74 74 4 75 75 1 76 76 8 77 77 2 78 78 1 79 79 160 80 80 1 81 81 15 82 82 8 83 83 1 84 84 1 85 85 1 86 86 1			
21 21 1 22 22 13 23 23 1 24 24 2 25 25 1 26 26 1 27 27 1 28 28 1 29 29 4 70 70 2 71 71 1 72 72 47 73 73 12 74 74 4 75 75 1 76 76 8 77 77 2 78 78 1 79 79 160 80 80 1 81 81 15 82 82 8 83 83 1 84 84 1 85 85 1 86 86 1 87 87 1			
22 22 13 23 23 1 24 24 2 25 25 1 26 26 1 27 27 1 28 28 1 29 29 4 70 70 2 71 71 1 72 72 47 73 73 12 74 74 4 75 75 1 76 76 8 77 77 2 78 78 1 79 79 160 80 80 1 81 81 15 82 82 8 83 83 1 84 84 1 85 85 1 86 86 1 87 87 1 88 88 5	20	20	1
22 22 13 23 23 1 24 24 2 25 25 1 26 26 1 27 27 1 28 28 1 29 29 4 70 70 2 71 71 1 72 72 47 73 73 12 74 74 4 75 75 1 76 76 8 77 77 2 78 78 1 79 79 160 80 80 1 81 81 15 82 82 8 83 83 1 84 84 1 85 85 1 86 86 1 87 87 1 88 88 5	21	21	1
23 23 1 24 24 2 25 25 1 26 26 1 27 27 1 28 28 1 29 29 4 70 70 2 71 71 1 72 72 47 73 73 12 74 74 4 75 75 1 76 76 8 77 77 2 78 78 1 79 79 160 80 80 1 81 81 15 82 82 8 83 83 1 84 84 1 85 85 1 86 86 1 87 1 30 88 88 5 90 90 7			13
24 24 2 25 25 1 26 26 1 27 27 1 28 28 1 29 29 4 70 70 2 71 71 1 72 72 47 73 73 12 74 74 4 75 75 1 76 76 8 77 77 2 78 78 1 79 79 160 80 80 1 81 81 15 82 82 8 83 83 1 84 84 1 85 85 1 86 86 1 87 87 1 88 88 5 89 89 25 90 90 7			
26 26 1 27 27 1 28 28 1 29 29 4 70 70 2 71 71 1 72 72 47 73 73 12 74 74 4 75 75 1 76 76 8 77 77 2 78 78 1 79 79 160 80 80 1 81 81 15 82 82 8 83 83 1 84 84 1 85 85 1 86 86 1 87 87 1 88 88 5 89 89 25 90 90 7 91 91 300 92 92 13 <td></td> <td></td> <td></td>			
27 27 1 28 28 1 29 29 4 70 70 2 71 71 1 72 72 47 73 73 12 74 74 4 75 75 1 76 76 8 77 77 2 78 78 1 79 79 160 80 80 1 81 81 15 82 82 8 83 83 1 84 84 1 85 85 1 86 86 1 87 87 1 88 88 5 89 89 25 90 90 7 91 91 300 92 92 13	25	25	1
28 28 1 29 29 4 70 70 2 71 71 1 72 72 47 73 73 12 74 74 4 75 75 1 76 76 8 77 77 2 78 78 1 79 79 160 80 80 1 81 81 15 82 82 8 83 83 1 84 84 1 85 85 1 86 86 1 87 87 1 88 88 5 89 89 25 90 90 7 91 91 300 92 92 13	26		1
29 4 70 70 2 71 71 1 72 72 47 73 73 12 74 74 4 75 75 1 76 76 8 77 77 2 78 78 1 79 79 160 80 80 1 81 81 15 82 82 8 83 83 1 84 84 1 85 85 1 86 86 1 87 87 1 88 88 5 89 89 25 90 90 7 91 91 300 92 92 13	27		
70 70 2 71 71 1 72 72 47 73 73 12 74 74 4 75 75 1 76 76 8 77 77 2 78 78 1 79 79 160 80 80 1 81 81 15 82 82 8 83 83 1 84 84 1 85 85 1 86 86 1 87 87 1 88 88 5 90 90 7 91 91 91 92 92 13	28	28	1
70 70 2 71 71 1 72 72 47 73 73 12 74 74 4 75 75 1 76 76 8 77 77 2 78 78 1 79 79 160 80 80 1 81 81 15 82 82 8 83 83 1 84 84 1 85 85 1 86 86 1 87 87 1 88 88 5 89 89 25 90 90 7 91 91 300 92 92 13	29	29	4
71 71 1 72 72 47 73 73 12 74 74 4 75 75 1 76 76 8 77 77 2 78 78 1 79 79 160 80 80 1 81 81 15 82 82 8 83 83 1 84 84 1 85 85 1 86 86 1 87 87 1 88 88 5 89 89 25 90 90 7 91 91 300 92 92 13			
72 72 47 73 73 12 74 74 4 75 75 1 76 76 8 77 77 2 78 78 1 79 79 160 80 80 1 81 81 15 82 82 8 83 83 1 84 84 1 85 85 1 86 86 1 87 87 1 88 88 5 90 90 7 91 91 300 92 92 13	70	70	2
72 72 47 73 73 12 74 74 4 75 75 1 76 76 8 77 77 2 78 78 1 79 79 160 80 80 1 81 81 15 82 82 8 83 83 1 84 84 1 85 85 1 86 86 1 87 87 1 88 88 5 90 90 7 91 91 300 92 92 13	71	71	1
73 73 12 74 74 4 75 75 1 76 76 8 77 77 2 78 78 1 79 79 160 80 80 1 81 81 15 82 82 8 83 83 1 84 84 1 85 85 1 86 86 1 87 87 1 88 88 5 89 89 25 90 90 7 91 91 300 92 92 13		72	47
74 74 4 75 75 1 76 76 8 77 77 2 78 78 1 79 79 160 80 80 1 81 81 15 82 82 8 83 83 1 84 84 1 85 85 1 86 86 1 87 87 1 88 88 5 89 89 25 90 90 7 91 91 300 92 92 13			
75 75 1 76 76 8 77 77 2 78 78 1 79 79 160 80 80 1 81 81 15 82 82 8 83 83 1 84 84 1 85 85 1 86 86 1 87 87 1 88 88 5 89 89 25 90 90 7 91 91 300 92 92 13			
76 76 8 77 77 2 78 78 1 79 79 160 80 80 1 81 81 15 82 82 8 83 83 1 84 84 1 85 85 1 86 86 1 87 87 1 88 88 5 89 89 25 90 90 7 91 91 300 92 92 13			
77 77 2 78 78 1 79 79 160 80 80 1 81 81 15 82 82 8 83 83 1 84 84 1 85 85 1 86 86 1 87 87 1 88 88 5 89 89 25 90 90 7 91 91 300 92 92 13			
78 78 1 79 79 160 80 80 1 81 81 15 82 82 8 83 83 1 84 84 1 85 85 1 86 86 1 87 87 1 88 88 5 89 89 25 90 90 7 91 91 300 92 92 13	76	76	8
78 78 1 79 79 160 80 80 1 81 81 15 82 82 8 83 83 1 84 84 1 85 85 1 86 86 1 87 87 1 88 88 5 89 89 25 90 90 7 91 91 300 92 92 13	77	77	2
79 79 160 80 80 1 81 81 15 82 82 8 83 83 1 84 84 1 85 85 1 86 86 1 87 87 1 88 88 5 89 89 25 90 90 7 91 91 300 92 92 13	78		
80 80 1 81 81 15 82 82 8 83 83 1 84 84 1 85 85 1 86 86 1 87 87 1 88 88 5 89 89 25 90 90 7 91 91 300 92 92 13			
81 81 15 82 82 8 83 83 1 84 84 1 85 85 1 86 86 1 87 87 1 88 88 5 89 89 25 90 90 7 91 91 300 92 92 13			
82 82 8 83 83 1 84 84 1 85 85 1 86 86 1 87 87 1 88 88 5 89 89 25 90 90 7 91 91 300 92 92 13			15
83 83 1 84 84 1 85 85 1 86 86 1 87 87 1 88 88 5 89 89 25 90 90 7 91 91 300 92 92 13			
84 84 1 85 85 1 86 86 1 87 87 1 88 88 5 89 89 25 90 90 7 91 91 300 92 92 13			
85 85 1 86 86 1 87 87 1 88 88 5 89 89 25 90 90 7 91 91 300 92 92 13			
86 86 1 87 87 1 88 88 5 89 89 25 90 90 7 91 91 300 92 92 13			
87 1 88 88 5 89 89 25 90 90 7 91 91 300 92 92 13			
88 88 5 89 89 25 90 90 7 91 91 300 92 92 13			
89 89 25 90 90 7 91 91 300 92 92 13			
90 90 7 91 91 300 92 92 13			
91 91 300 92 92 13			
92 92 13			
	91	91	
93 93 1	92	92	13
	93	93	1

```
94
          94
                              1
95
          95
                             28
96
          96
                              1
97
          97
                              1
98
          98
                              3
99
          99
                              1
```

[100 rows x 2 columns]

4.5.1 Observando quais classes possuem mais de 10 instâncias...

```
In [38]: class_instances.loc[class_instances["Nž Instancias"] > 10]
```

Out[38]:		Classe	Nž Instancias
	0	0	38
	16	16	86
	22	22	13
	30	30	24
	38	38	448
	43	43	73
	55	55	519
	60	60	22
	72	72	47
	73	73	12
	79	79	160
	81	81	15
	89	89	25
	91	91	300
	92	92	13
	95	95	28

É possível perceber que o número de instâncias em algumas classes aumentou consideravelmente. Isso pode ser explicado pelo fato de que existem mais classes a serem alocadas, logo, os *outliers* podem ficar em classes individuais sem atrapalhar tanto o algoritmo como acontecia com as outras classes para um k menor.

4.5.2 Adicionando a coluna calculada kmeans.label_ ao conjunto de dados

```
In [39]: dataset["K-classes"] = kmeans.labels_
In [40]: dataset.loc[ dataset["pancreas"] > 40000]
Out [40]:
               adipose_tissue
                               adrenal_gland
                                              animal_ovary
                                                                        bladder \
                                                             appendix
                                    0.000003
                                                  39.250285
                     1.903841
                                                              1.77945
                                                                       0.354372
         1940
               bone_marrow cerebral_cortex
                                                 colon
                                                         duodenum
                                                                   endometrium \
                  0.000003
                                   0.000004 0.307113 19.355371
         1940
                                                                      0.424481
```

```
esophagus fallopian_tube gallbladder
                                                         heart
                                                                  kidney
              0.000004
                               0.789698
                                            0.000004 8.966004 0.000002 0.000003
        1940
                                                 placenta prostate
                 lung lymph_node
                                      pancreas
                                                                       rectum \
                         0.375191 48970.17971 37.154553 0.357527 0.000003
        1940 0.78126
              salivary_gland skeletal_muscle
                                                  skin small intestine \
        1940
                   18.775735
                                     0.213618 1.65235
                                                              11.967953
              smooth_muscle
                               spleen
                                        stomach
                                                   testis
                                                            thyroid
                                                                       tonsil \
                   0.424843 3.678518 5.874027 2.799198 0.000005 0.000003
        1940
                           class K-classes
        1940 b'ENSG00000091704'
4.6 Hierárquico, K = 2
In [41]: from sklearn.cluster import AgglomerativeClustering
         clustering = AgglomerativeClustering(n_clusters=2, affinity='euclidean', linkage="ware
In [42]: a = np.array(clustering.labels_)
In [43]: unique, counts = np.unique(a, return_counts=True)
        dicionario = dict(zip(unique, counts))
        print(dicionario)
        lista = list(dicionario.items())
        class_instances = pd.DataFrame(lista, columns=["Classe", "Nž Instancias"])
        class_instances
{0: 1999, 1: 1}
Out [43]:
           Classe Nž Instancias
                            1999
        1
                1
                               1
In [44]: dataset.drop("K-classes", axis=1, inplace=True)
In [45]: dataset["K-classes-hierar"] = clustering.labels_
In [46]: dataset.loc[ dataset["pancreas"] > 40000]
Out [46]:
              adipose_tissue adrenal_gland animal_ovary appendix
                                                                      bladder \
                                   0.000003
                                                39.250285
        1940
                    1.903841
                                                            1.77945
                                                                     0.354372
              bone_marrow cerebral_cortex
                                               colon
                                                       duodenum endometrium \
                 0.000003
                                  0.000004 0.307113 19.355371
                                                                    0.424481
        1940
              esophagus fallopian_tube gallbladder
                                                         heart
                                                                  kidney
                                                                             liver \
```

liver \

```
1940
                                     0.000004
                                                                          0.789698
                                                                                                        0.000004 8.966004 0.000002 0.000003
                                          lung lymph_node
                                                                                          pancreas
                                                                                                                    placenta prostate
                                                                                                                                                                       rectum \
                                  0.78126
                                                            0.375191 48970.17971 37.154553 0.357527 0.000003
                     1940
                                   salivary_gland skeletal_muscle
                                                                                                                      skin small_intestine
                     1940
                                               18.775735
                                                                                        0.213618 1.65235
                                                                                                                                                  11.967953
                                  smooth muscle
                                                                          spleen
                                                                                               stomach
                                                                                                                        testis
                                                                                                                                              thyroid
                                                                                                                                                                       tonsil \
                                              0.424843 3.678518 5.874027 2.799198 0.000005 0.000003
                     1940
                                                                 class K-classes-hierar
                     1940 b'ENSG00000091704'
4.7 Hierárquico, K = 5
In [47]: from sklearn.cluster import AgglomerativeClustering
                     clustering = AgglomerativeClustering(n_clusters=5, affinity='euclidean', linkage="wardenged by the clustering of the clu
In [48]: a = np.array(clustering.labels_)
4.7.1 Percebemos que a classe 2 possui muitas instâncias quando comparamos com as outras
             classes, o que caracteriza um desbalanceamento de classes. Com o desbalanceamento,
             isto prejudica significativamente a classificação do algoritmo e separação em grupos.
In [49]: unique, counts = np.unique(a, return_counts=True)
                     dicionario = dict(zip(unique, counts))
                    print(dicionario)
                    lista = list(dicionario.items())
                     class_instances = pd.DataFrame(lista, columns=["Classe", "Nž Instancias"])
                     class_instances
{0: 4, 1: 1, 2: 1993, 3: 1, 4: 1}
Out [49]:
                            Classe
                                              Nž Instancias
                    0
                                       0
                     1
                                       1
                                                                          1
                     2
                                       2
                                                                   1993
                     3
                                       3
                                                                          1
                                        4
                                                                          1
In [50]: dataset["K-classes-hierar"] = clustering.labels_
In [51]: dataset.loc[dataset["K-classes-hierar"] != 2]
Out [51]:
                                  adipose_tissue adrenal_gland animal_ovary
                                                                                                                                                  appendix
                                                                                                                                                                                   bladder \
                     393
                                            621.921288
                                                                               586.903809
                                                                                                                726.130275
                                                                                                                                           3459.251765
                                                                                                                                                                          1998.657487
                                          2138.647694
                                                                               794.959656
                                                                                                             3768.027372 3800.906260
                     510
                                                                                                                                                                         1587.586089
```

591	0.3173				711780	0.708744	
1315	2027.5903					4514.697940	
1718	7377.3826					7548.121360	
1874	28.5576				187417	836.317672	
1940	1.9038	41 0.00	0003 39.2	50285 1.7	779450	0.354372	
	bone_marrow	cerebral_co	rtex	olon duo	denum e	endometrium \	\
393	644.389201	450.26	1176 786.21	0311 1476.53	38312	827.738599	
510	1518.243824	2663.40	9328 3528.73	2999 1139.20	01844 1	.073.937771	
591	0.628672	0.38	1577 0.30	7113 0.00	00003	0.000004	
1315	3463.984873	2430.64	7195 2346.34	6398 1800.04	1 9515 6	460.605882	
1718	6349.591147	1938.41	2520 3095.70	3101 4330.07	73027 2	996.838208	
1874	12623.741610	19.07	8863 546.66	1857 5275.72	21159	114.609960	
1940	0.000003	0.00	0004 0.30	7113 19.3	55371	0.424481	
	esophagus	fallopian_tu	be gallblad	der he	eart	kidney \	
393	511.969171	1772.8724	_	960 179.320	0080 4	53.572584	
510	999.391875	1113.4744	25 1861.762	342.949	9654 4	95.939254	
591	0.350664	0.3948	49 0.814	776 6.72	1503	1.246079	
1315	3857.301972	3293.0413	85 3637.973	632 428.126	6692 7	99.982414	
1718	2114.502808	3158.7926	95 12201.266	550 1187.99	5534 55	87.416115	
1874	199.878375	110.5577	44 3055.408	986 26.898	3012	32.398042	
1940	0.000004	0.7896	98 0.000	004 8.966	6004	0.000002	
		_			_		
	liver	lung	lymph_node	pancreas	_	acenta \	
393	532.393163	3921.923707	5612.861412	70.29885		775824	
510	476.813657	2925.817586	3549.309422	148.63185		203533	
591	0.000003	0.390630	26.263389	0.00000		412828	
1315	1292.954824	4394.585827	7931.543465	188.802629		020932	
1718	5344.408289	9707.151804	5612.861412	843.586214		684565	
1874	125.785198	2179.714570	1485.757432	18.07684		285898	
1940	0.000003	0.781260	0.375191	48970.179710	37.	154553	
	prostate	rectum	salivary_gla	nd skeletal	_muscle	skin	\
393	507.687681	1079.879800	565.9543	66	.221554	591.541126	
510	793.708910	2558.912879	482.8046	123	.898392	343.688699	
591	0.715053	0.309421	1.3411	24 0	.000002	2.313289	
1315	3128.357190	2558.912879	992.4317	12 183	.711409	1391.278291	
1718	2288.169831	3945.119614	1166.7778	324	.699234	1311.965514	
1874	60.779511	519.827526	13529.2582	60 2	. 136179	99.140971	
1940	0.357527	0.000003	18.7757	35 0	.213618	1.652350	
	small_intest	ine smooth_m	uscle	spleen s	stomach	testis	\
393	2133.287			-	. 247028	199.442884	,
510	2456.422				.215893	1091.687363	
591	0.000				.000003	2.449299	
1315	2600.037				.130272	1137.174337	
1718	5032.524				.116138	1445.086157	

```
class K-classes-hierar
                    thyroid
                                   tonsil
                 609.157832 5352.577737 b'ENSG00000019582'
         393
                 751.141989 2533.426473 b'ENSG00000034510'
                                                                               0
         510
         591
               32253.304180
                                 0.952416 b'ENSG00000042832'
                                                                               3
         1315
                2729.759911 6231.975147 b'ENSG00000075624'
                                                                               0
                4919.064000 2809.627104 b'ENSG00000087086'
                                                                               0
         1718
         1874
                  87.022547 1561.962187 b'ENSG00000090382'
                                                                               4
                                 0.000003 b'ENSG00000091704'
         1940
                   0.000005
                                                                               1
4.8 Hierárquico, K = 10
In [52]: from sklearn.cluster import AgglomerativeClustering
         clustering = AgglomerativeClustering(n_clusters=10, affinity='euclidean', linkage="waters=10")
In [53]: a = np.array(clustering.labels_)
In [54]: unique, counts = np.unique(a, return_counts=True)
         dicionario = dict(zip(unique, counts))
         print(dicionario)
         lista = list(dicionario.items())
         class_instances = pd.DataFrame(lista, columns=["Classe", "Nž Instancias"])
         class_instances
{0: 60, 1: 1, 2: 1931, 3: 1, 4: 1, 5: 1, 6: 2, 7: 1, 8: 1, 9: 1}
                   Nž Instancias
Out [54]:
            Classe
                 0
         0
                                60
         1
                 1
                                 1
         2
                 2
                              1931
         3
                 3
                                 1
         4
                 4
                                 1
         5
                 5
                                 1
         6
                 6
                                 2
         7
                 7
                                 1
         8
                 8
                                 1
         9
                 9
                                 1
In [55]: dataset["K-classes-hierar"] = clustering.labels_
In [56]: dataset.loc[(dataset["K-classes-hierar"] != 2) & (dataset["K-classes-hierar"] != 0)]
Out [56]:
               adipose_tissue adrenal_gland animal_ovary
                                                                 appendix
                                                                               bladder
         393
                   621.921288
                                   586.903809
                                                 726.130275
                                                              3459.251765
                                                                           1998.657487
         510
                  2138.647694
                                   794.959656
                                                3768.027372
                                                              3800.906260
                                                                           1587.586089
         591
                     0.317307
                                     1.863187
                                                   0.000004
                                                                 0.711780
                                                                              0.708744
```

1874

1940

2058.487869

11.967953

195,427656

0.424843

1364.730134 7486.447452

5.874027

3.678518

24,492986

2.799198

1315	2027.5903			2296.14		7644.519		4514.697940	
1657	1.90384	41 1.552	2656	3.92	5029	2.847	121	3.543719	
1718	7377.3826	24 3490.369	741	4827.78	5071	5295.644	677	7548.121360	
1874	28.5576	10 46.579	9667	54.95	0399	1854.187	417	836.317672	
1933	0.31730	0.000	0003	0.00	0004	0.000	004	0.000004	
1940	1.90384	41 0.000	0003	39.25	0285	1.779	450	0.354372	
	bone_marrow	cerebral_cor	tex	со	lon	duoden	um e	ndometrium	\
393	644.389201	450.261		786.210	311	1476.5383	12	827.738599	
510	1518.243824	2663.409	328	3528.732	999	1139.2018	44 1	073.937771	
591	0.628672	0.381	.577	0.307	113	0.0000	03	0.000004	
1315	3463.984873	2430.647		2346.346		1800.0495		460.605882	
1657	3.143362	3.815		2.764		2.4885		16.979253	
1718	6349.591147	1938.412		3095.703		4330.0730		996.838208	
1874	12623.741610	19.078		546.661		5275.7211		114.609960	
1933	0.000003	0.000		1.228		2.7650		0.000004	
1940	0.000003	0.000		0.307		19.3553		0.424481	
1340	0.000003	0.000	7004	0.507	110	13.0000	' -	0.424401	
	esophagus	fallopian_tuk		allbladd	or	hear	+	kidney \	
393	511.969171	1772.87240	_	3049.1609		179.32008		53.572584	`
510	999.391875	1113.47442		.861.7625		342.94965		95.939254	
510	0.350664	0.39484		0.8147				1.246079	
						6.72450			
1315	3857.301972	3293.04138		4 0736		428.12669		99.982414	
1657	2.805311	9733.02999		4.0738		0.67245		2.492157	
1718	2114.502808	3158.79269		201.2665		1187.99553		87.416115	
1874	199.878375	110.55774		3055.4089		26.89801		32.398042	
1933	0.000004	0.78969		8.1477		0.00000		34.890199	
1940	0.000004	0.78969	98	0.0000	04	8.96600	4	0.000002	
	liver	lung	•	ph_node		pancreas	-	lacenta \	
393	532.393163	3921.923707		2.861412		70.298851		.775824	
510	476.813657	2925.817586		.309422		48.631857		.203533	
591	0.000003	0.390630		.263389		0.000002		.412828	
	1292.954824				18	88.802629		.020932	
1657	5.850474	3.906299		.503825		0.200854	8	. 256567	
1718	5344.408289	9707.151804	5612	2.861412	84	43.586214	9032	.684565	
1874	125.785198	2179.714570	1485	.757432		18.076847	202	. 285898	
1933	13058.258680	15.625194	C	.000004		4.017077	0	.000004	
1940	0.000003	0.781260	C	.375191	489	70.179710	37	.154553	
	prostate	rectum	saliv	ary_glan	d sl	keletal_mu	scle	skir	ı \
393	507.687681	1079.879800	5	65.95430	1	66.22	1554	591.541126	3
510	793.708910	2558.912879	4	82.80461	7	123.89	8392	343.688699)
591	0.715053	0.309421		1.34112	4	0.00	0002	2.313289)
1315	3128.357190	2558.912879	9	92.43171	2	183.71	1409	1391.278291	
1657	7.150531	6.188423		2.14579		1.06	8090	9.914097	
1718		3945.119614	11	.66.77782					
1874				29.25826			6179	99.140971	

```
1933
         0.00004
                       0.928263
                                       0.000003
                                                         0.000002
                                                                       0.00003
1940
                       0.000003
         0.357527
                                      18.775735
                                                         0.213618
                                                                       1.652350
      small_intestine smooth_muscle
                                              spleen
                                                                         testis
                                                          stomach
393
          2133.287574
                          1291.521900
                                        6463.155915
                                                      1527.247028
                                                                    199.442884
510
          2456.422297
                          1797.084749
                                        3833.015631
                                                      1715.215893
                                                                   1091.687363
591
             0.000003
                             0.424843
                                           0.367852
                                                         0.000003
                                                                       2.449299
                                                                   1137.174337
1315
          2600.037730
                          8322.669088
                                        5201.424283
                                                      2232.130272
1657
             2.094392
                             4.248427
                                           3.678518
                                                         2.643312
                                                                      17.494990
1718
          5032.524122
                          4112.477630
                                       12036.110500 3072.116138
                                                                   1445.086157
1874
          2058.487869
                           195.427656
                                        1364.730134
                                                      7486.447452
                                                                      24.492986
                                           0.000004
1933
             1.196795
                             0.000004
                                                         5.874027
                                                                       0.349900
1940
            11.967953
                             0.424843
                                           3.678518
                                                         5.874027
                                                                       2.799198
           thyroid
                          tonsil
                                                class
                                                       K-classes-hierar
393
        609.157832
                    5352.577737
                                  b'ENSG0000019582'
510
        751.141989
                    2533.426473
                                  b'ENSG00000034510'
                                                                       6
                                                                       7
591
      32253.304180
                                  b'ENSG0000042832'
                        0.952416
       2729.759911
                    6231.975147
                                  b'ENSG00000075624'
                                                                       8
1315
1657
          1.832054
                        3.174720 b'ENSG00000085465'
                                                                       9
1718
       4919.064000 2809.627104
                                  b'ENSG00000087086'
                                                                       5
                                                                       4
1874
         87.022547
                    1561.962187
                                  b'ENSG00000090382'
1933
          0.000005
                        0.000003
                                  b'ENSG00000091583'
                                                                       3
1940
          0.000005
                        0.000003 b'ENSG00000091704'
                                                                       1
```

4.9 Hierárquico, K = 100

```
Out [59]:
                        Nž Instancias
               Classe
          0
                     0
                                     564
          1
                     1
                                       2
          2
                     2
                                      10
          3
                     3
                                      28
          4
                     4
                                      10
          5
                     5
                                      12
```

6	6	8
7	7	6
		4
8	8	
9	9	69
10	10	9
11	11	34
12	12	5
13	13	2
14	14	2
15	15	2
16	16	15
17	17	3
18	18	13
19	19	17
20	20	333
21	21	8
		2
22	22	
23	23	320
24	24	2
25	25	1
26	26	1
27	27	2
28	28	2
29	29	60
70	70	1
71	71	1
72	72	1
73	73	1
74	74	3
75	75	1
76	76	_
77	1 0	1
		1
	77	1
78	77 78	1 1
78 79	77 78 79	1 1 1
78 79 80	77 78 79 80	1 1 1 1
78 79 80 81	77 78 79 80 81	1 1 1 1
78 79 80 81 82	77 78 79 80 81 82	1 1 1 1 1
78 79 80 81 82 83	77 78 79 80 81 82 83	1 1 1 1 1 1
78 79 80 81 82 83 84	77 78 79 80 81 82 83	1 1 1 1 1 1 1
78 79 80 81 82 83	77 78 79 80 81 82 83	1 1 1 1 1 1
78 79 80 81 82 83 84	77 78 79 80 81 82 83	1 1 1 1 1 1 1
78 79 80 81 82 83 84 85	77 78 79 80 81 82 83 84	1 1 1 1 1 1 1 1
78 79 80 81 82 83 84 85 86	77 78 79 80 81 82 83 84 85	1 1 1 1 1 1 1 1
78 79 80 81 82 83 84 85 86 87	77 78 79 80 81 82 83 84 85 86	1 1 1 1 1 1 1 1 1
78 79 80 81 82 83 84 85 86 87	77 78 79 80 81 82 83 84 85 86 87	1 1 1 1 1 1 1 1 1 1 8
78 79 80 81 82 83 84 85 86 87 88 89	77 78 79 80 81 82 83 84 85 86 87 88	1 1 1 1 1 1 1 1 1 8 1 4
78 79 80 81 82 83 84 85 86 87 88	77 78 79 80 81 82 83 84 85 86 87 88	1 1 1 1 1 1 1 1 1 8

93	93	1
94	94	1
95	95	1
96	96	1
97	97	1
98	98	1
99	99	1

[100 rows x 2 columns]

4.9.1 Com k = 100 percebemos que há uma diversidade maior de classes com mais de dez instâncias

```
In [60]: class_instances.loc[class_instances["Nž Instancias"] > 10]
```

```
Out[60]:
              Classe Nž Instancias
          0
                    0
                                  564
          3
                    3
                                   28
          5
                    5
                                   12
          9
                   9
                                   69
          11
                   11
                                   34
                   16
                                   15
          16
          18
                   18
                                   13
          19
                   19
                                   17
          20
                   20
                                  333
          23
                   23
                                  320
          29
                  29
                                   60
                   34
          34
                                   20
          40
                   40
                                   19
          44
                   44
                                  203
          48
                   48
                                   33
          49
                   49
                                   79
```

In [61]: dataset["K-classes-hierar"] = clustering.labels_

In [62]: dataset.describe()

Out $[62]$:	adipose_tissue	adrenal_gland	animal_ovary	appendix	bladder	\
count	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	
mean	39.040157	41.486336	51.338392	52.555181	48.932735	
std	195.264717	181.466441	210.726346	252.295704	221.732356	
min	0.000003	0.000003	0.000004	0.000004	0.000004	
25%	3.173068	3.105311	3.140023	7.117802	7.087438	
50%	12.692271	15.526556	19.625143	21.353406	21.262314	
75%	31.730678	34.158423	43.175314	46.265713	42.524627	
max	7377.382624	5216.922745	4827.785071	7644.519332	7548.121360	
	bone_marrow o	cerebral_cortex	colon	duodenum	endometrium	\
count	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	

mean	53.852077	45.7976	67 40.663197	39.632611	52.066032	
std	393.758129	123.1833	44 157.028175	185.991398	186.169270	
min	0.000003	0.0000	0.000003	0.000003	0.000004	
25%	1.257345	7.6315	45 3.071134	2.765053	7.428423	
50%	9.430086	19.0788	63 15.355670	13.825265	21.224067	
75%	34.576981	41.9734	99 33.782474	30.415583	50.937760	
max	12623.741610	2663.4093			6460.605882	
	esophagus	fallopian_tube	gallbladder	heart	kidney \	
count	2000.000000	2000.000000	2000.000000		2000.000000	
mean	41.493874	53.691974	57.023709	21.915492	27.728612	
std	139.247221	261.725555	314.203534		136.411996	
min	0.000004	0.000004	0.000004	0.000002	0.000002	
25%	3.506638	7.896982		1.569051	2.492157	
50%	14.026553	23.690945	20.369393	6.724503	9.968628	
75%	35.066382	47.381890	48.886544	15.690507	24.921571	
max	3857.301972	9733.029992	12201.266550	4512.141526 5	5587.416115	
	liver	lung	lymph_node	-	placenta \	
count	2000.000000	2000.000000	2000.000000 2	000.000000 200	0.00000	
mean	35.790131	54.864549	54.294117	35.087462 5	2.105545	
std	336.523288	277.654367	277.993904 1	108.114405 24	2.242244	
min	0.000003	0.000004	0.000004	0.000002	0.00004	
25%	1.170095	7.812597	3.751913	0.602562	4.128284	
50%	5.850474	23.437791	18.759564	2.008539 2	0.641418	
75%	17.551423	46.875582	48.774865	6.025616 4	5.411120	
max	13058.258680				2.684565	
	prostate	rectum s	alivary_gland	skeletal_muscle	skin	\
count	2000.000000	2000.000000	2000.000000	2000.000000		`
mean	46.389605	42.376928	27.628763	22.053380		
std	241.613858	165.470468	308.543568	208.651352		
	0.000004	0.000003	0.000003	0.000002		
min or"				0.427236		
25%	7.150531	3.094211	2.414023			
50%	17.876327	15.471057	8.046744	2.136179		
75%	42.903184	34.036326	18.775735	8.544717		
max	9585.286431	3945.119614	13529.258260	7732.968610	1939.858329	
	small_intest:	_	-			\
count	2000.000					
mean	43.3968	344 52.185	982 56.1645	31 37.587312	44.472439	
std	193.638	153 233.848	790 347.5618	19 207.009834	114.878046	
min	0.000	0.000	0.0000	0.000003	0.000003	
25%	5.983	976 4.248	427 3.6785	18 5.874027	6.997996	
50%	14.9599	941 21.242	137 22.0711	07 14.685068	20.993988	
75%	32.9118	370 42.484	273 47.8207	32 29.370135	45.486973	
max	5032.524					

	thyroid	tonsil	K-classes-hierar
count	2000.000000	2000.000000	2000.000000
mean	72.736881	46.143284	20.402500
std	742.846516	222.768691	19.007371
min	0.000005	0.000003	0.000000
25%	4.580134	3.174720	0.000000
50%	22.900670	15.873599	20.000000
75%	54.961609	38.096639	29.000000
max	32253.304180	6231.975147	99.000000

Para k = 100, percebeu-se que as classes com maior quantidade de instâncias foram as iniciais no Hierárquico. Já no K-Means, as classes intermediárias apresentaram a maior quantidade de instâncias.

5 2.1. Fixar K no K-means e alterar o parâmetro de quantidade máxima de iterações

Escolha um número fixo de K e altere o parâmetro do K-Means referente ao número máximo de iterações: 1, 10 e 100.

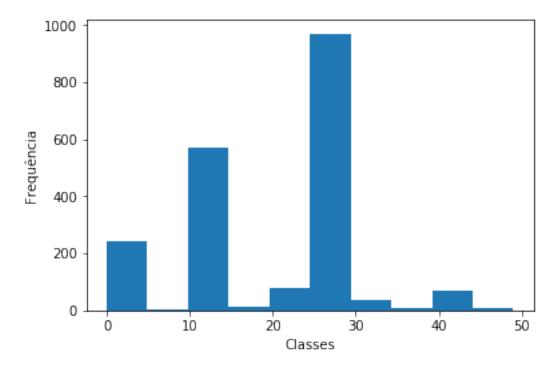
5.1 K-means com K = 50, max_iter = 1

```
In [63]: from sklearn.cluster import KMeans
         kmeans = KMeans(n_clusters=50, max_iter=1, random_state=0).fit(X)
In [64]: a = np.array(kmeans.labels_)
In [65]: unique, counts = np.unique(a, return_counts=True)
         dicionario = dict(zip(unique, counts))
         print(dicionario)
         lista = list(dicionario.items())
         class_instances = pd.DataFrame(lista, columns=["Classe", "Nž Instancias"])
         class_instances
{0: 238, 1: 1, 2: 1, 3: 1, 4: 1, 5: 1, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1, 11: 1, 12: 1, 13: 1, 14:
Out [65]:
             Classe Nž Instancias
         0
                  0
                               238
         1
                  1
                                 1
```

```
10
         10
                             1
11
         11
                             1
12
         12
                             1
13
         13
                             1
                          568
14
         14
15
         15
                             2
                             1
16
         16
17
                             6
         17
18
         18
                             1
19
         19
                             1
20
                             2
         20
21
         21
                            72
22
         22
                             1
23
                             1
         23
24
         24
                             1
25
         25
                            11
26
         26
                             4
                             2
27
         27
28
         28
                          952
29
         29
                             1
30
                             1
         30
31
         31
                            31
32
                             1
         32
33
                             2
         33
34
         34
                             3
35
         35
                             1
36
                             1
         36
37
         37
                             1
                             2
38
         38
                             3
39
         39
40
         40
                            35
41
         41
                             2
42
         42
                             2
43
         43
                            27
44
         44
                             1
                             2
         45
45
                             5
46
         46
47
         47
                             1
48
         48
                             1
49
         49
                             1
```

5.1.1 Adicionando a coluna calculada kmeans.label_ ao conjunto de dados

Out[67]: Text(0,0.5,'Frequência')



In [68]: class_instances.loc[class_instances["Nž Instancias"] > 10]

Out[68]:		Classe	Nž	Instancias
	0	0		238
	14	14		568
	21	21		72
	25	25		11
	28	28		952
	31	31		31
	40	40		35
	43	43		27

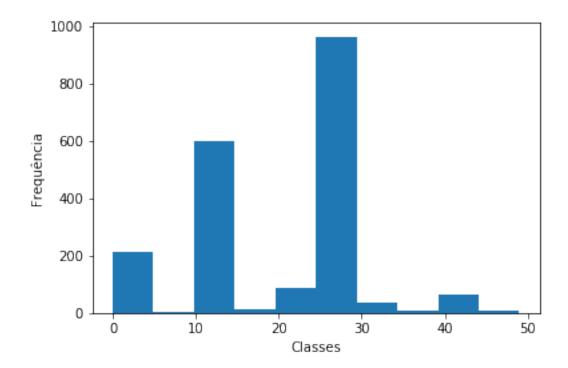
5.2 K-means com K = 50, max_iter = 10

{0: 207, 1: 1, 2: 1, 3: 1, 4: 1, 5: 1, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1, 11: 1, 12: 1, 13: 1, 14:

Out[71]:		Classe	Nž	Instancias
	0	0		207
	1	1		1
	2	2		1
	3	3		1
	4	4		1
	5	5		1
	6	6		1
	7	7		1
	8	8		1
	9	9		1
	10	10		1
	11	11		1
	12	12		1
	13	13		1
	14	14		596
	15	15		2
	16	16		1
	17	17		6
	18	18		1
	19	19		1
	20	20		2
	21	21		83
	22	22		1
	23	23		1
	24	24		1
	25	25		11
	26	26		4
	27	27		2
	28	28		945
	29	29		1
	30	30		1
	31	31		31
	32	32		1
	33	33		2
	34	34		3
	35	35		1
	36	36		1
	37	37		1
	38	38		2
	39	39		3
	40	40		35
	41	41		2
	42	42		2
	43	43		26

```
44
         44
                             1
45
         45
                             2
46
         46
                             5
47
         47
                             1
                             1
48
         48
49
         49
                             1
```

5.2.1 Adicionando a coluna calculada kmeans.label_ ao conjunto de dados



In [74]: class_instances.loc[class_instances["Nž Instancias"] > 10]

Out[74]:		Classe	Nž	Instancias
	0	0		207
	14	14		596
	21	21		83
	25	25		11
	28	28		945

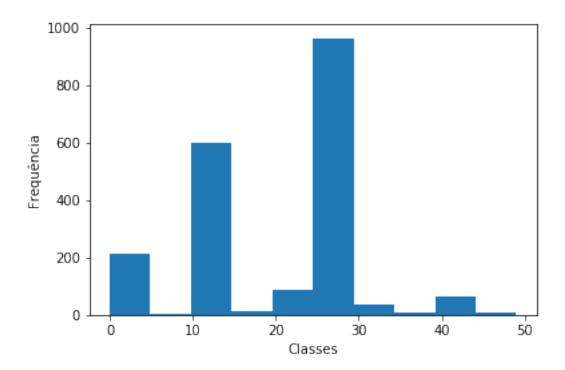
```
31
         31
                           31
40
         40
                           35
43
         43
                           26
```

K-means com K = 50, max_iter = 100

```
In [99]: from sklearn.cluster import KMeans
         kmeans = KMeans(n_clusters=50, max_iter=100, random_state=0).fit(X)
         y_kmeans = kmeans.fit_predict(X)
In [100]: a = np.array(kmeans.labels_)
In [101]: unique, counts = np.unique(a, return_counts=True)
          dicionario = dict(zip(unique, counts))
          print(dicionario)
          lista = list(dicionario.items())
          class_instances = pd.DataFrame(lista, columns=["Classe", "Nž Instancias"])
          class_instances
{0: 207, 1: 1, 2: 1, 3: 1, 4: 1, 5: 1, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1, 11: 1, 12: 1, 13: 1, 14:
Out[101]:
              Classe
                     Nž Instancias
          0
                   0
                                 207
          1
                   1
                                   1
          2
                   2
                                   1
          3
                   3
                                   1
          4
                   4
                                   1
          5
                   5
                                   1
          6
                   6
          7
```

```
26
         26
                            4
27
         27
                            2
28
         28
                          945
29
         29
                            1
30
         30
                            1
31
         31
                           31
32
         32
                            1
                            2
33
         33
34
         34
                            3
35
         35
                            1
36
         36
                            1
         37
37
                            1
38
         38
                            2
39
         39
                            3
40
         40
                           35
                            2
41
         41
42
         42
                            2
43
         43
                           26
44
         44
                            1
                            2
45
         45
                            5
46
         46
47
         47
                            1
48
         48
                            1
49
         49
                            1
```

5.3.1 Adicionando a coluna calculada kmeans.label_ ao conjunto de dados



In [104]: class_instances.loc[class_instances["Nž Instancias"] > 10]

Out[104]:		Classe	Nž	${\tt Instancias}$
	0	0		207
	14	14		596
	21	21		83
	25	25		11
	28	28		945
	31	31		31
	40	40		35
	43	43		26

6 2.2. Alterar o parâmetro linkage do Hierárquico

Linkage: {"ward", "complete", "average", "single"}, optional (default="ward") Which linkage criterion to use. The linkage criterion determines which distance to use between sets of observation. The algorithm will merge the pairs of cluster that minimize this criterion.

ward minimizes the variance of the clusters being merged. average uses the average of the distances of each observation of the two sets. complete or maximum linkage uses the maximum distances between all observations of the two sets. single uses the minimum of the distances between all observations of the two sets.

Referência: http://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.htm

6.1 Hierárquico, K = 100

```
In [88]: from sklearn.cluster import AgglomerativeClustering
         clustering = AgglomerativeClustering(n_clusters=100, affinity='euclidean', linkage="a'
In [89]: a = np.array(clustering.labels_)
In [90]: unique, counts = np.unique(a, return_counts=True)
         dicionario = dict(zip(unique, counts))
         print(dicionario)
         lista = list(dicionario.items())
         class_instances = pd.DataFrame(lista, columns=["Classe", "Nž Instancias"])
         class_instances
{0: 1824, 1: 21, 2: 2, 3: 6, 4: 16, 5: 3, 6: 4, 7: 1, 8: 3, 9: 2, 10: 2, 11: 3, 12: 2, 13: 1,
Out [90]:
             Classe
                     Nž Instancias
         0
                   0
                               1824
         1
                   1
                                 21
         2
                   2
                                  2
                   3
         3
                                  6
                   4
                                  16
         4
         5
                   5
                                  3
         6
                   6
                                  4
         7
                   7
                                  1
         8
                   8
                                  3
         9
                   9
                                  2
                                  2
         10
                  10
                                  3
         11
                  11
         12
                  12
                                  2
         13
                  13
                                  1
         14
                  14
                                  4
         15
                  15
                                  1
         16
                  16
                                  1
         17
                  17
                                  1
         18
                  18
                                  1
         19
                  19
                                  1
         20
                  20
                                  5
         21
                                  1
                 21
         22
                  22
                                  1
         23
                  23
                                  3
         24
                 24
                                  2
         25
                  25
                                  1
```

```
70
                   70
                                     1
          71
                   71
                                     1
          72
                   72
                                     1
          73
                   73
                                     1
          74
                   74
                                     1
          75
                   75
                                     1
          76
                   76
                                     1
          77
                   77
                                     1
          78
                   78
                                     1
          79
                   79
                                     1
          80
                   80
                                     1
          81
                   81
                                     1
          82
                   82
                                     1
          83
                   83
                                     1
          84
                   84
                                     1
          85
                   85
                                     1
          86
                   86
                                     1
          87
                   87
                                     1
          88
                   88
                                     1
          89
                   89
                                     1
          90
                   90
                                     1
          91
                   91
                                     1
          92
                   92
                                     1
          93
                   93
                                     1
          94
                   94
                                     1
          95
                   95
                                     1
          96
                   96
                                     1
          97
                   97
                                     1
                                     9
          98
                   98
          99
                   99
                                     1
          [100 rows x 2 columns]
In [91]: class_instances.loc[class_instances["Nž Instancias"] > 10]
Out [91]:
             Classe Nž Instancias
          0
                   0
                                1824
          1
                   1
                                   21
                   4
                                   16
In [92]: dataset['K-classes-hierar'] = clustering.labels_
In [93]: dataset.describe()
```

0.000003

adrenal_gland

2000.000000

41.486336

181.466441

animal_ovary

2000.000000

51.338392

210.726346

0.000004

appendix

2000.000000

52.555181

252.295704

0.000004

bladder

2000.000000

48.932735

221.732356

0.000004

Out [93]:

count

mean

std

min

adipose_tissue

2000.000000

39.040157

195.264717

0.000003

25% 50% 75%	3.17306 12.6922 31.7306	71 15.52	6556	3.140 19.625 43.175	5143	7.117 21.353 46.265	406	7.087438 21.262314 42.524627	
max	7377.3826	24 5216.92	2745	4827.785	5071	7644.519	332	7548.121360	
count mean std	bone_marrow 2000.000000 53.852077 393.758129	cerebral_co 2000.00 45.79 123.18	0000 7667 3344	2000.0000 40.6631 157.0281	197 175	duoden 2000.0000 39.6326 185.9913	00 2 11 98	ndometrium 000.000000 52.066032 186.169270	\
min 25%	0.000003 1.257345	0.00 7.63		0.0000 3.0711		0.0000 2.7650		0.000004 7.428423	
50%	9.430086	19.07		15.3556		13.8252		21.224067	
75%	34.576981	41.97		33.7824		30.4155		50.937760	
max	12623.741610	2663.40	9328	3528.7329	999	5275.7211	59 6	460.605882	
count	esophagus 2000.000000	fallopian_tu		gallbladd@ 2000.00000		hear		kidney '	\
mean	41.493874	53.6919	74	57.02370		21.91549		27.728612	
std	139.247221	261.7255		314.20353		116.72660		36.411996	
min	0.000004	0.0000		0.00000		0.00000		0.000002	
25%	3.506638	7.8969		8.14775		1.56905		2.492157	
50% 75%	14.026553 35.066382	23.6909 47.3818		20.36939 48.88654		6.72450 15.69050		9.968628 24.921571	
max	3857.301972	9733.0299		40.0005 2201.26655		15.09050		87.416115	
max	0001.001012	0100.0200	02 1.	2201.2000		.012.11102	0 00	01.110110	
	liver	lung		mph_node		pancreas	_	lacenta \	
count	2000.000000	2000.000000		0.000000		00.00000		.000000	
mean	35.790131	54.864549		4.294117		35.087462		.105545	
std	336.523288	277.654367		7.993904		8.114405		.242244	
min	0.000003	0.000004		0.000004		0.000002		.000004	
25%	1.170095	7.812597		3.751913		0.602562		.128284	
50%	5.850474	23.437791		8.759564		2.008539		.641418	
75%	17.551423			8.774865		6.025616		.411120	
max	13058.258680	9707.151804	193	1.543465	4897	70.179710	9032	. 684565	
	prostate	rectum	sali	vary_gland	d sk	celetal_mu	scle	skir	ı \
count	2000.000000	2000.000000	20	000.00000)	2000.00	0000	2000.000000)
mean	46.389605	42.376928		27.628763	3	22.05	3380	35.257338	3
std	241.613858	165.470468	;	308.543568	3	208.65	1352	95.223057	7
min	0.000004	0.000003		0.000003	3	0.00	0002	0.000003	3
25%	7.150531	3.094211		2.414023	3	0.42	7236	3.304699)
50%	17.876327	15.471057		8.046744	4	2.13	6179	16.52349	5
75%	42.903184	34.036326		18.775735		8.54		36.351689	
max	9585.286431	3945.119614	13	529.258260)	7732.96	8610	1939.858329)
	small_intest:	ine smooth_m	uscle	នា	oleer	n sto	mach	testis	s \
count	2000.0000	_		2000.00	•			2000.000000	
mean	43.3968		85982	56.16				44.472439	

std	193.6381	53 233.84	8790	347.561819	207.009834	114.878046
min	0.0000	0.00	0004	0.000004	0.000003	0.000003
25%	5.9839	76 4.24	8427	3.678518	5.874027	6.997996
50%	14.9599	41 21.24	2137	22.071107	14.685068	20.993988
75%	32.9118	70 42.48	34273	47.820732	29.370135	45.486973
max	5032.5241	22 8322.66	9088	12036.110500	7486.447452	2554.268510
	thyroid	tonsil	K-cl	asses-hierar	K-classes	
count	2000.000000	2000.000000		2000.00000	2000.000000	
mean	72.736881	46.143284		3.20350	21.039000	
std	742.846516	222.768691		14.20521	10.258708	
min	0.000005	0.000003		0.00000	0.00000	
25%	4.580134	3.174720		0.00000	14.000000	
50%	22.900670	15.873599		0.00000	28.000000	
75%	54.961609	38.096639		0.00000	28.000000	

99.00000

49.000000

6231.975147

32253.304180

Out[94]: Text(0,0.5,'Frequência')

max

